

EFFICIENT HVAC SYSTEM
GROUP 16 - Fall 2011 - Spring 2012
FINAL DOCUMENTATION



Daniel Galarza, CpE

Chung Chien Hsia, EE

Kevin Ivy, EE

Marvin Fernandez, CpE

Table of Contents

1.0 Introduction	1
1.1 Executive Summary	1
2.0 Project Description	2
2.1 Project History	2
2.2 Motivation and Goals.....	5
2.2.1 Carbon Dioxide Safety	7
2.3 Sponsors and Subcontractors	8
2.4 Objectives	9
2.5 Design Specifications	10
2.5.1 Original Hardware Specifications	10
2.5.2 New Software Specifications	11
2.6 Smart Phone Application Description	12
2.6.1 Primary Goals.....	12
2.6.2 Requirements and Ideas for Development	12
2.7 User capabilities	13
2.8 Administrator Capabilities.....	15
2.9 User Administrator Comparison	17
2.10 Roles and Responsibilities	18
2.11 Coding Breakdown	20
2.12 Project Milestones.....	20
2.12.1 Fall Semester	21
2.12.2 Winter Break Preparation	21
2.12.3 Spring Semester.....	22
2.13 Financing and Budget	22
3.0 Research Related to Project Definition.....	25
3.1 Researching Strategies	26
3.2 Existing Competitor Products	26
3.2.1 Honeywell Prestige 2.0® Comfort System.....	26

3.2.2 BAYweb.....	29
3.3 Web Infrastructure.....	30
3.4 Choosing Service Providers	31
3.4.1 Thingspeak.....	31
3.4.2 ioBridge	32
3.4.3 Pachube	33
3.4.4 Go Daddy	33
3.4.5 Hostmonster	33
3.4.6 Host Gator	33
3.5 The Internet of Things	34
3.6 Web Browser Interface.....	35
3.7 Coding Languages and Protocols	36
3.7.1 HTML	36
3.7.2 XML.....	36
3.7.3 CSS.....	36
3.7.4 PHP.....	37
3.7.5 SQL and MySQL	38
3.7.6 Java.....	39
3.7.7 JavaScript.....	39
3.7.8 AJAX	39
3.7.9 REST.....	40
3.7.10 C#.....	40
3.7.11 Ruby on Rails.....	40
3.7.12 C.....	40
3.7.13 jQuery.....	41
3.7.14 TCP/IP.....	42
3.8 OSI.....	44
3.9 Smart Phone Application Technology Research	46
3.9.1 Overview	46
3.9.2 Specific Definitions.....	47
3.9.3 Technologies Considered.....	47

3.9.4 Mobile Web Site Development	50
3.9.5 Best Practices	51
3.9.6 Issues to Consider.....	54
3.10 Wireless Communication.....	55
3.10.1 Overview	55
3.10.2 Relationship of Web site and Database.....	55
3.10.3 Zigbee Protocol Model for Communication.....	56
3.11 Hardware Components	57
3.11.1 LCD Touch Screen.....	57
3.11.2 Environmental Conditions for NHD LCD Monitor.....	59
3.11.3 Built-in SSD1963 controller	59
3.11.4 LCD Touch Screen Comparisons.....	62
3.11.5 Panda Board	64
3.11.6 Explorer 16 Kit.....	66
3.11.7 Comparison.....	66
3.12 Security	67
4.0 Hardware and Software Design.....	70
4.1. Hardware Design.....	71
4.1.1 Hardware Redesign Purpose	71
4.1.2 I/O Expansion Board	72
4.1.3 Wi-Fi Extension	73
4.1.4 Ethernet Daughter Board	74
4.2 PIC32 Microprocessor.....	76
4.3 Remote Unit	78
4.4 Hardware Components	80
4.4.1 Temperature Sensor	81
4.4.2 Humidity Sensor	82
4.4.3 CO2 Sensor.....	83
4.4.4 Testing the Indoor and Outdoor Sensor	85
4.4.5 Testing Machine to Machine Communication.....	87
4.5 User Account Data Structure.....	87

4.6 MySQL Database	91
4.7 Smart Phone Application Design.....	93
4.8 Smart Phone Web.....	95
5.0 Design Summary.....	97
5.1 Design of HVAC system emulator	97
5.1.1 Layout of Project	97
5.2 Design of Web Database Structure and Website	99
5.2.1 Structure of the Database.....	99
5.2.2 Customer Information Structure	100
5.2.3 User Settings Structure	100
5.2.4 Current Settings Structure	101
5.2.5 Scheduler Structure.....	101
5.2.6 Administrator Structure.....	102
5.3 Design of the Web Site to host the system.....	103
5.3.1 Design on Web Pages.....	104
5.4 Design of the Mobile Phone Application.....	107
5.4.1 Common Features of All Pages.....	108
5.4.2 Main Page	109
5.4.3 Scheduler Page.....	111
5.4.4 User Settings Page	112
5.4.5 Administrator Settings Page	114
5.4.6 Final Layout of Mobile Site	116
5.5 Simulated HVAC Hardware.....	117
6.0 Testing Requirements Summary	117
6.1 Hardware Wireless Testing	118
6.2 MySQL Database Testing	118
6.3 Web Browser HVAC Control System Testing.....	119
6.4 Smart Phone Application Testing	119
6.5 Web Security Testing	120
6.6 Testing Environment	121
7.0 Future Additions and Considerations	121

7.1 Future Additions	122
7.1.1 Self-Updating Firmware.....	122
7.1.2 Weather Monitoring Service	123
7.1.3 Distributing the Phone App.....	123
7.2 Concluding Comments	124
Appendix	126
Bibliography	126
Figures	127
Project Operation	128
Copy Permissions	131

Figure Table of Contents

Figure 2.1: Generation 1 HVAC Control System (iTemp)	3
Figure 2.2: Generation 2	4
Figure 2.3: Residential Energy Consumption	6
Figure 2.4 User Capabilities Diagram.....	14
Figure 2.5: Screen display in generation two	15
Figure 2.6: Administrator Capabilities Diagram	16
Figure 2.7: User/Administrator Comparison	18
Figure 2.8: Roles and Responsibility	18
Figure 2.9: Milestone Schedule	21
Figure 2.10: Senior Design Team 16 Budget	24
Figure 3.1: Researching Strategies Summarized, divided by Team Members	25
Figure 3.2: The residential use main menu	27
Figure 3.3: Energy Save Settings for maximum cost savings	28
Figure 3.4: BAYweb.com web HVAC application	30
Figure 3.5: Sample Thingspeak embedded graph.....	32
Figure 3.6: HTML Form (left) and its hidden PHP programming (right)	38
Figure 3.7: SQL connection and query written in C.....	41
Figure 3.8: TCP/IP Hierarchy Summary	43
Figure 3.9: Microchip TCP/IP Stack for PIC18, PIC24, dsPIC & PIC32	44
Figure 3.10: OSI Model Layer Structure, and comparisons to TCP/IP	46
Figure 3.11: Touch Panel Characteristics for NHD-7.0-800480WF-CTXI#-T	58
Figure 3.12: Table with Specifications for 7 inch NHD LCD Panel	58
Figure 3.13: Table of Environmental conditions of NHD 7" LCD	59
Figure 3.14: Table of Features of SSD 1963 controller	60
Figure 3.15: Clock Control Diagram	60
Figure 3.16: SSD1963 controller's State diagram	61
Figure 3.17: Table Comparison for LCD touch screen for three generations	63
Figure 3.18: Block diagram of SSD1963	64
Figure 3.19: OMAP4430 PandaBoard.....	65
Figure 3.20: Display interface for PandaBoard.....	66

Figure 3.21: Table of Comparison between different board selections.....	67
Figure 3.22: Security Risk Assessment.....	70
Figure 4.1: Wi-Fi Extension Board.....	74
Figure 4.2: Ethernet PICtail Plus Daughter Board.....	75
Figure 4.3: PIC32 detail information.....	76
Figure 4.4: Table of Support resolution and size for the PIC32.....	77
Figure 4.5: Comparison between PIC24 and PIC32.....	78
Figure 4.6: PIC24 Pin Out Diagram.....	79
Figure 4.7: TXM315.....	80
Figure 4.8: RXM315.....	80
Figure 4.9: Basic Fahrenheit Temperature Sensor (+5° to +300°F).....	81
Figure 4.10: LM34 temperature sensor (Bottom view).....	81
Figure 4.11: CHS-MSS humidity sensor.....	83
Figure 4.12: Sensor Linearity (Ta=25°C, Edc=5V).....	83
Figure 4.13: CO2 Sensor schematics.....	84
Figure 4.14: Pin description for 2792 CO2 sensor.....	84
Figure 4.15: Indoor sensor schematics.....	86
Figure 4.16: Customer Information Data Structure.....	88
Figure 4.17: User Settings Data Structure.....	88
Figure 4.18: Current Settings data structure.....	89
Figure 4.19: Scheduler Data Structure.....	90
Figure 4.20: Administrator Login Structure.....	90
Figure 4.21: Administrator Capabilities Data Structure.....	90-91
Figure 4.22: Table of Database Terminology.....	92
Figure 4.23: First Layout of Mobile Website.....	94
Figure 5.1: Relationships of All Project Entities.....	98
Figure 5.2: Customer Information MySQL table.....	100
Figure 5.3: User Settings MySQL table.....	100
Figure 5.4: Main Screen MySQL table.....	101
Figure 5.5: Scheduler MySQL table.....	102
Figure 5.6: Administrator MySQL table.....	102-103

Figure 5.7: Admin Login MySQL Table	103
Figure 5.8: HTML design of web pages.....	105
Figure 5.9: Main Page (Mobile) Class Diagram.....	110
Figure 5.10: Scheduler Class Diagram.....	111
Figure 5.11: User Settings Class Diagram	113
Figure 5.12: Administrator Settings Class Diagram.....	115
Figure 5.13: Layout of Entire Mobile Website.....	116

Section 1: Introduction

1.1 Executive Summary

Computer and Electrical Engineering are diverse fields with applications that affect everyone in the world on the daily basis. With the rate that technology continues to grow and improve provides many beneficial changes to the way society operates as a whole. In today's day and age we are able to have access to many electronic applications from anywhere in the world due to the internet and phone applications. One issue in Florida that is always under constant improvement is energy efficiency in Heating, Ventilation, and Air Conditioning systems (HVAC). For the past 20-30 years engineers have worked on making these improvements to HVAC systems to become more energy efficient. Especially in Florida based on the fact that the weather is constantly hot and humid for most of the year, thus resulting in majority of HVAC systems residential and commercial to be running year round, leading to many consumers having expensive energy bills. To address this issue, last year, a fellow senior design group designed a HVAC system that presented to be one of the most energy efficient systems yet. This HVAC system used components from a traditional HVAC system with enhanced options that allowed the user to set temperature and humidity set points, introduce fresh air into the building, and select from multiple tolerance levels that ranged from "Max Comfort" to "Max Energy Savings." This HVAC control and feedback system was controlled with a LCD touch screen that is capable of wireless access. The capability of the main control unit to be accessed wirelessly and with the technology that is available today with Internet and cell phones, we took the enhanced features of this HVAC system to the next level. Florida is very common to seeing extreme weather changes throughout one day and in residential cases almost all users are out of their house for a good portion of the day making consumers unable to make the necessary changes to their HVAC system when they are away. In both business and residential cases the providing company of the system can make improvements in the maintenance and operation of their system by having constant access to their HVAC system.

To make improvements to this HVAC system our group is proposed to make this system be accessible for its users through the Internet and phone applications, making a HVAC system with the capability to ascend the efficiency that energy is consumed with this system. Our plan to accomplish this goal was to first recreate the redesign of the HVAC hardware designed by engineer Tim VanderMay of Proton Design Inc. Our hardware consisted of all new parts including the microcontroller which was a PIC32 microcontroller. We also used a Wi-Fi board, and Ethernet daughter board, all hardware parts are by the creation of Microchip. Once hardware simulator of the HVAC system has been established we created TCP/IP stack software with the inclusion of Microchip's TCP/IP stack in order to

give our hardware the ability to connect to the Internet and send out any information from the microcontroller.

With the establishment of our hardware simulator, the PIC32 Microprocessor was programmed to emulate the functions of the HVAC system. Once that was done we created a database server to retrieve and store any information received from the HVAC system. An Internet browser was created using MySQL database allowing users to read information stored on the database server from their HVAC system to review or make changes to. In addition to having a web browser for users with this capability we also created a mobile phone web application that will give users the same abilities as that of the web browser; we wanted our mobile phone application to be compatible for the major smart phone application markets (Android and iPhone) and instead of making an application for each market, we pursued a mobile phone website that will give off the appearance of an phone application making it easily compatible for both markets. Also, the appearance of the HVAC system on the web browser and phone application was the same user interface as with the original HVAC system. Finally, the database server was able to retrieve any changes that were made over the Internet browser or mobile phone application and relayed it back to the hardware simulator in order for the hardware to make the proper adjustments. The final goal of our project was to have the HVAC system be accessible from anywhere.

Section 2: Project Description

2.1. Project History

Our project which is called “The Efficient HVAC Control System,” gives the residents the ability to control their HVAC units more conveniently and more efficiently. This project is considered the third generation for the same project to make it better and better. Generation 1 was done by Cory Glass, Derick Holzmacher, Andrew Mertens, and Joshua New in 2010 summer and 2010 fall semester. They were the first group who were aware of the high cost of on homeowners electricity bill and difficulty to control the HVAC units, so they started to make a new control unit in order to balance the cost and efficiency and most importantly - the comfort of a user’s home or office. They built the first generation of the HVAC Control System, called iTemp, which was equipped with a touch screen which can show the indoor and outdoor temperature and humidity to the user and lets the user choose actions like cooling or heating, as well as changes in humidity if they feel it is necessary. The first generation have already equipped with the function of comfort setting which can let the user choose the mode they want. For example, if user wants to have the maximal comfort, he or she just needs to choose maximal comfort and let the micro-controller decides when to start the AC unit and how long it last. However, generation one had

encountered some difficulties and some parts were not working correctly. That why the second generation worked to improve the prototype of group 1. Below, in Figure 2.1, you can see the work of the previous senior design team.



Figure 2.1: Generation 1 HVAC Control System (iTemp)

Generation two, called the “Efficient HVAC Control System” project was done by Steve Jones, Mathew, Arcuri, Elroy Ashtian, Jr., and Jerthwin L. Prospere in spring 2011 to summer 2011 semester. They worked on solving some of the issues of the generation one hardware. For example, they enhanced the user friendly interface through a wall mount touch screen thermostat. They also added the web connectivity by allowing the user to change the settings via the Internet. This will make the product more convenient and useful for future markets. It also adds some new functions, letting the user have more control over the AC system. It allows user to set a schedule for the units, indicating when the system should begin and stop running as well as controlling other settings. For example, user can set when to start the AC unit and how long it run and also set the temperature and humidity thermostats. In order to make the maintenance more easily, group 2 adds some features that is restricted and is only accessible to the technician who has the valid password. It is not only allowing flexible configuration of devices but also easily to troubleshoot the problems. If a user wants to have different scent in different rooms, he could choose the function “MOOD SCENTS”. They also add more control flexibility to the project. For instance, the generation one project has only the user main menu for user to change temperature and humidity setting for the room. On the other hand, generation two add more features in there, like the scheduler, user settings, and

administrator screens to the main project. Scheduler allows user can make arrangement of the HVAC system beforehand instead of changing settings manually every time. User setting allows user can have basic control over the system, for instance, user can choose the system mode, such as on or off buttons, or the user can also check the tolerance level of each difference comfort options. They also added an option for technicians who had valid password to access the advance settings. For example, they can change almost all the parameters of the HVAC system, like the tolerance level and how long to spread the scent when user chooses "Mood Scents" options. Below is a another sample of the second generation project in Figure 2.2.

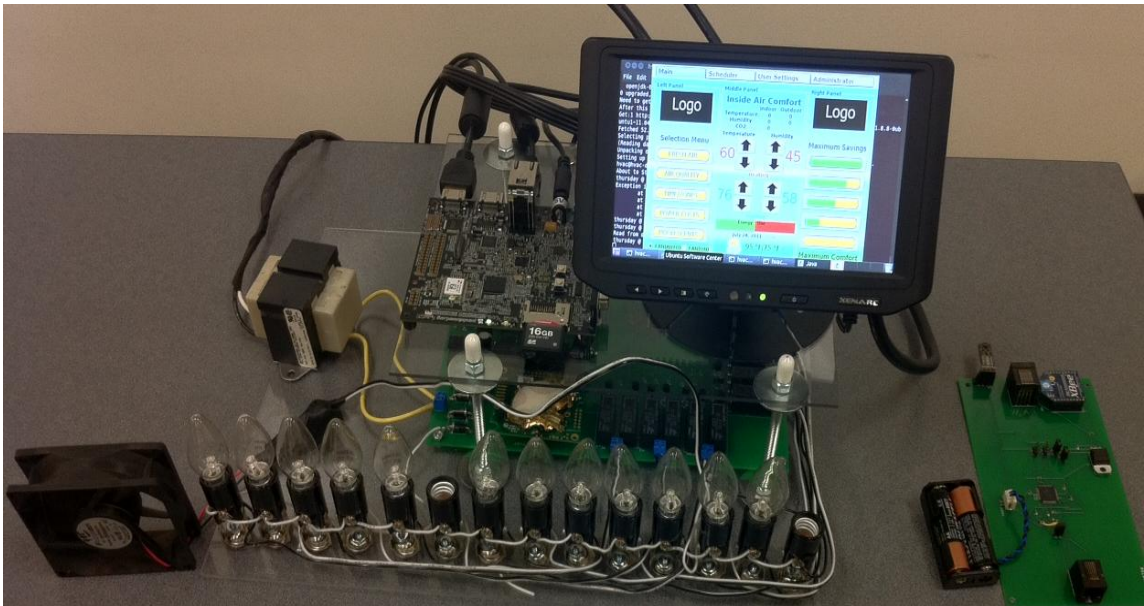


Figure 2.2: Generation 2 (Reprinting Permission Requested from UCF senior design Group 2 Spring 2011)

Below is the pros and cons of the previous two projects as well as the intended advantages to be added for the current generation of this project.

Generation 1 Advantages

- First come up with the idea and start working on the prototype
- Have set the clear project requirements & specifications
- Use touch screen to allow residents can more easily control the unit

Generation 1 Disadvantages

- Software is slow.
- Lacked full functionality. (Some of the function is not working properly)

- Unorganized code
- Don't have a hardware simulation for AC units in order to test the control unit.
- Blown regulators
- Wifi chips have problems

Generation 2 Advantages

- Make the efforts to full functionality and extend more functions
- The codes become more organized and are working correctly
- LCD touch screen become more sensitive to fingers
- Created a hardware simulation to do testing

Generation 2 Disadvantages

- Having trouble for internet connectivity.
- Limit the accessibility for only using Internet
- Not able to do updating hardware by Wifi
- The Panda board is too powerful for the design and expensive
- The LCD touch screen is really pricy and can't be massive production.
- The Wifi chip didn't work really well

Generation 3 Advantages

- Add the fully internet control and also the smart phone application which can do the same as internet.
- Have the ability to update the hardware by Wifi
- Choose a much cheaper LCD touch screen to decrease the cost.

Generation 3 Disadvantages

- Hardware incompatibilities caused hardware simulator to be much simpler than planned.
- PIC32 microcontroller had a very high learning curve.

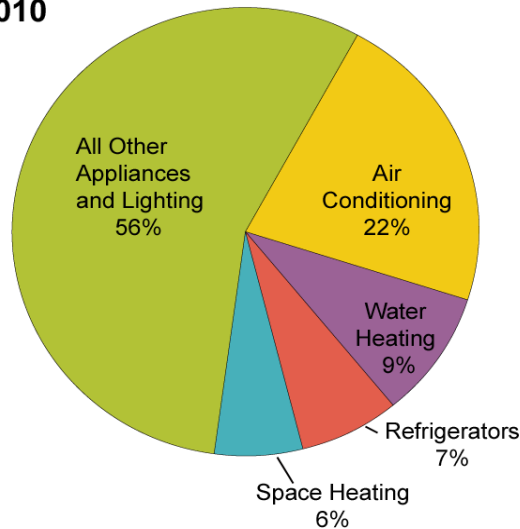
2.2 Motivation and Goals

Most people have experienced that even though they didn't turn on the air-conditioning often, they will still be surprised about the electric bill at the end of the month. Or that you're never satisfied with the efficiency of your AC units. Or sometimes you forgot to turn off the AC before you left the house. Instead of just wasting money, we figured out a way to solve all the problems at once. From Figure 2.3, almost 30% of residential energy consumption is used for air

conditioning and space heating. It includes the energy that was wasted so it gives us lot of room to save electricity. Our idea on this project can be implemented and expanded to many different areas, for instance, refrigeration, lighting, other appliances or even water heating. From Figure 2.3 below, we can save lots of electricity and money waste at the same time. It makes us more interested in HVAC system.

Left behind by two Senior Design teams from before us, is a prototype for such a system. Originally called iTemp, and now called The Efficient HVAC Control system, gives residents newly found control over their indoor climate. It conserves electricity by providing new options for indoor climate, such as controlling both the temperature and humidity of the area. It uses outdoor and indoor sensors to detect humidity and temperature outside and inside the building, and it communicates this information to a central control unit that allows users to detect this information. The product is almost mature for mass production, even though it needed to have some improvements like internet connectivity.

How Electricity Is Used in Homes, 2010



Source: U.S. Energy Information Administration, *Annual Energy Outlook 2011*, Table 4, Reference Case. Projections based on the Residential Energy Consumption Survey 2005.

Figure 2.3: Residential Energy Consumption (Reprinting Permission Requested from U.S. Energy information and administration)

We were also motivated by the group ahead of us because we have meeting with two previous group members. They told us about how they made the generation two and what part is good enough for implementation and which part need to be redesigned or invented to make our product to meet the needs of our sponsors. We are standing on the shoulder of others so we can see bigger pictures of the

whole project and were glad to have a chance to participate in this project. Remote control of the HVAC system is one of our motivations behind a smart control system, because many people have smart phones that allow them to have an internet connection wherever they go. It will become a trend that allow user to control their HVAC systems while they are away from the building and away from their main controller. Creating a Smart phone application and web page are the best way to apply this concept and it also easy to be used by the user also.

We went and took the previous HVAC system, debugged it, and designed it further and increased the efficiency of the system. Adding web interface through Wi-Fi was a main goal as it is allowing user adjustable controls to use the Internet and we also added a new interface which is only allowed the control by technician for better diagnosis and repair the AC system. We also extended our design to the mobile phone so that we can not only control through computers but also through the mobile phone. We created a new smart phone application which can work on android and/or iPhone to control the AC units anytime you want. Our goal was to make a website that can work not only on the computer but also smart phones. Using this strategy, we can save more work on developing the application because the android phones and i-Phone applications use different languages and also save money to buy the software for designing.

A new innovation for our project was to try to figure out a way to update our firmware by wireless connection. It is more convenient compare with the traditional updated methods, such as using external wire to update or just taking all the units back to company then updated it manually. Using the wireless to update the firmware can not only save time but also save manpower and most importantly save cost for company. However, our schedule did not allow us time to design this year, and must be held back for future research.

With the high cost of oil and thus energy more efficient ways of consuming electricity becomes more important topics now. Especially in Florida where using AC units almost every day for cooling purpose. We hope our system can reduce the bill and increase the comfort of customer that why we thought the HVAC system would be a good project to work with. Besides, the project will cover some good understanding of software skill helping us for future preparation of working. For example, we used Dreamweaver to design the webpage which is suitable for not only the resolution on the computer but also mobile devices.

2.2.1 Carbon Dioxide Safety

The need for Carbon Dioxide (CO₂) monitoring systems is not easily understood by many, but Natural Air Energy wants it to be a permanent part of the project. Any biology student can tell you that Carbon Dioxide is perfectly natural is many environments. Why would buildings need detectors for Carbon Dioxide? An

interesting problem in homes and workspace environments is that sometimes buildings and workplace environments produce situations that can raise carbon dioxide concentration to dangerous levels. The most common carbon dioxide sources are from poor ventilation and from leaking man made sources of carbon dioxide. For example, carbon dioxide canisters are used at gas stations, 8 restaurants and fast food chains (used to disperse soft drinks.) If an area that uses these canisters experiences a leak and the work area is not properly ventilated, the carbon dioxide gas could silently and swiftly leak into the workplace. After that, the carbon dioxide lowers the oxygen level of the area, creating illness and possible asphyxiation (suffocation). In turn, household sources of carbon dioxide come from activities as simple as heating, cooking, using household sprays or consuming electricity. Any biology student can tell you carbon dioxide comes from something as simple as breathing. Carbon dioxide is odorless, naturally occurring, and is heavier than air, meaning it sinks to the bottom of an area like a building and doesn't move if the temperature of a building is low enough.

Therefore, HVAC system companies such as Natural Air Energy take an interest in creating CO2 detectors for household or business use in response. They are almost always a part of an HVAC system, and do not independently exist to check CO2 levels. For example, a modern HVAC system will use a CO2 sensor to determine that there are a lot of people in an area, and will adjust ventilation settings so that an area gets fresh air accordingly.

2.3 Sponsors and Subcontractors

The motivation behind our group proceeding to do this project is the opportunity presented to us by our sponsor Will Carson and the company he helped found Natural Air Energy Saving Systems. Natural Air Energy Savings Systems was founded in 2008 and is located in Winter Haven, FL as an AC contractor that is focused on energy savings and inside air quality. Natural Air Energy presented us with the assignment of making their technology in HVAC systems accessible via internet browser and phone application. This opportunity provided us to work on a project that would be on the verge of being applied in the real world and would be a very outstanding asset to have added on our resumes, based on Natural Air being one of the fastest growing companies in the state of Florida with over 2.5 million a year in annual sales.

This is not Natural Air Energy first occasion working with a senior design group at the University of Central Florida, which has led our sponsor group to being comfortable with presenting us the opportunity to work on a project of this caliber. The previous work that Natural Air Energy has done with these previous senior design groups that has led to our group trying to further advance the technology were the iTemp and HVAC Control and Feedback system 2.0. The first project that Natural Air Energy presented to a senior design group was the iTemp which

was a HVAC system capable of being controlled by a touch screen, it was a first generation type and overall not very successful. The second generation of this project, the HVAC Control and Feedback system 2.0 continued on from the first project presenting many more capabilities in addition to the first generation as well as having wireless access. A major key factor of this project was the patentable technology of tolerance levels that a user could have set starting from maximum savings to maximum comfort and settings in between. Success of this project led to Natural Air Energy wanting to give this HVAC control and Feedback system to allow users to access their system through the Internet browser and phone applications, since their plan is to have is the HVAC system mass produced and have available to the public and businesses in the near future.

In order to make this technology available to the public consumer the HVAC system has to be capable of being mass produced at an efficient and profitable manner. Once an assigned portion of our project it has now been relayed to a subcontractor Tim VanderMey founder of Proton Design Inc. Tim's assignment was to redesign the second generation of the HVAC system with the most efficient technology available today and also have it where it can be mass produced to the public. Already, Tim has brought many changes to the second generation design by the previous senior design group, however, circumstances prevented VanderMay from going forward with the design. We decided to use parts of his original design to construct a simulator that would serve as our testing environment for the project.

2.4 Objectives

The software that we made is going to allow all of the capabilities and features that are available to the user and administrator through the LCD touch screen of the HVAC system available through the Internet and phone applications using the same user interface. Our plan of procedure that was used for accomplishing this objective was to use machine to machine technology, by having a sensor that is able to read and set all the data from the HVAC system and transport it to be available over the Internet. The sensor was also be able to set and read data accessed through the Internet and relay it back to the HVAC system.

The hardware for the HVAC system created by the previous group was going through redesign with our subcontractor. However, we have been made aware of the hardware that is planned to be used in the redesign process. We used a PIC 32 microchip in our hardware replica of the HVAC system. The PIC 32 microchip was connected to an LCD touch screen that would simulate controls similar to the HVAC system touch screen of the previous group. In this replica we provided the microchip with Wi-Fi accessibility in order to connect with our software that would provide the touch screen to be accessed over the internet and phone applications. Even though our replica would not control a real HVAC system, we proved that our software was able to connect to our device over the internet and

phone applications and make the necessary working changes to the hardware that would be intended for in the real HVAC system.

There was two ways that we would have liked the system to be accessed through the Internet; the first is to have our system be browser accessible. Making a browser that can be accessible for the system using a data structure based off pseudo code that will display all the data and capabilities given to the user when they use the LCD touch screen. This data structure will allow users to view their settings and make necessary changes they want to make and in addition to that will allow the administrator to make administrative changes to the HVAC system providing the administrative password. Also, we set up a web server that would retrieve all the data from the HVAC system and keep it stored on the server for when the customer accesses their account the information that they would see would be accurate. If they submit any changes to their system the server would read the changes and relay it to the HVAC system updating it to the current settings that the user has submitted for.

In addition to providing web browser access to this HVAC system we also made this system be accessible through a phone application. The two major markets that we are aiming to have these phone applications for are the Android and iPhone application markets. An initial thought was to make applications that were compatible to the Android and iPhone market, while this would of proven to be fairly reasonable for the Android market it presented to be more of a challenge to make an application for the iPhone market in addition to having to be charged in order to make the application as well. Instead our intentions was to make a mobile website that will give the appearance of an application, this way it would be more easier to have the application be accessible for both the Android and iPhone market. Today's technology with many smart phones being touch screen and our HVAC system being controlled by a LCD touch screen, we created a user interface for our phone application. This lets the customer who uses the phone application to see everything on their phone exactly how they would see it if they were using the touch screen on the HVAC system itself.

2.5 Design Specifications

2.5.1 Original Hardware Specifications

The following specifications were developed over first several meetings within our group members as well as sending that specification to our sponsor. Using the same standard as generation two and added more features on the software side. However, the hardware part was outsourced to Tim VanderMay who worked in Proton Design Inc. as Subcontractor because the sponsor's needs. However, we made some parts of the hardware for demonstration purpose.

- Make it ready for optimum mass production reduce the price of product to around 500 dollars (optimum design to achieve the lowest cost production of a reliable product)
- Achieve flash updating of our operating firmware, via a direct site plug-in; a land line phone connection; by Wi-Fi connection from outside the building
- Monitor current air temperatures in multiple locations with highly accuracy +/- 2%.
- Monitor current air humidity in multiple locations ranging from 0% to 100 % with accuracy +/- 2%.
- Send the most updated information to the control unit and also display on the LCD touch screen every 1 minute.
- Have the ability to change the thermostats and humidity on touch screen.
- Goal is the reduce the usage for electric power by at least 5%
- Microprocessor will also check the CO₂ Concentration of the room, if it is greater than 600ppm which is typical of indoor air and is an acceptable and safe level. The control unit will send the warning message to the user.
- Have the ability to have internet connection remotely from an AC service office to the customer's building, to be able to read a problem and/or change the operation of an installed unit
- Create a smart phone application to control the HVAC system
- Create a mobile webpage that allow user to change the setting for HVAC system and the webpage will be updated the temperature and humidity information of indoor and outdoor by every 10 seconds.

2.5.2 New Software Specifications

The new software specifications that our group set out to accomplish with this HVAC system were presented to our group in our first meeting with our sponsor Natural Air Energy Savings Systems and subcontractor Tim Vandermay of Proton Design Incorporated. These software specifications will become the main focus of our group during this project due to the fact that majority of the hardware responsibilities has been removed by the sponsors.

- Set Up Machine to Machine communication
- Establish a web server to relay information from HVAC system to web browser and mobile phone applications and vice versa
- Create a Web Database to allow input and output of data using MySQL database
- Create Web Security in order to provide protection of users HVAC system information over the internet

- Create a mobile phone website that would appear as an phone application for both the android and iPhone application market using Dreamweaver since it is easily compatible with both application markets
- Make web browser and mobile phone application one user interface to allow user to see exactly what they would see on the touch screen of their HVAC system
- Firmware over Air (FOA) to allow firmware updates to occur when readily available
- Create TCP/IP Stack software to allow Hardware to be able to communicate to internet

2.6 Smart Phone Application Description

2.6.1 Primary Goals

The primary goal of this project is obviously the creation of the network database and its main website for communication with the HVAC system. The next step is the creation of a mobile phone application to compliment the website and system. The phone application was designed such that it would look exactly the same screen as the main HVAC system, with all the same functionalities and features directly controllable from the phone itself. The final result is that consumers will be able to adjust their air and heat settings using the HVAC system no matter where they are. They can use their computer or their phone to operate their HVAC system, even if they are not home. All three interfaces will be extremely similar, no matter what format they happen to access. In fact the mobile phone will be nearly identical to the system itself, especially if the phone has touch screen capabilities.

2.6.2 Requirements and Ideas for Development

For the purposes of this project, the development of the phone application was designed primarily on the iPhone and Android systems. At first, creating two versions of the mobile application was considered. This largely depended on the resources needed to obtain the technologies as well as the main differences between them. Multiple languages or interfaces were needed to be learned, the decision was made to focus on only one technology, as it increases the time needed to learn these methods as well as the languages needed to design the main website.

One consideration of the mobile apps is the cross-capability of the apps and the Internet websites. Due to the widespread use of different technologies in today's

world, a system that can run in many different formats was advantageous. Just as different people use different browsers such as Google Chrome, Mozilla Firefox, and Internet Explorer, there are different mobile systems as well, such as the Android and iPhone web browsers. A system that can run in as many of these technologies as possible is very marketable. The more cross-capability the HVAC system can obtain, the greater the number of customers it can support. This also gives consumers more and better reasons to purchase it.

2.7 User capabilities

It was important for us to declare what is the right for user to change and what is the right for technician to change first; otherwise, it will cause many problems in the first step. But we made the best efforts to allow user have the maximal control over their comfort and flexibility.

Figure 2.4 below is a class diagram depicting all the capabilities of a user. When the user boot the device and they should see tabs on the top – Main, Scheduler, User Settings, and Administrator. Main is the menu that contain the three different panels (Left,middle,and right panels). In left panel, you have 3 selections - Fresh Air, Power costs, and Mood Scents. In the middle panel, you have not only the indoor and outdoor temperature and humidity but also the setting for temperature and humidity that allow the user to change as their interest. For example, by choosing 55 °F as heating standard and 80 °F as the cooling standard, the AC will be activated for cooling when the temperature inside is above 80 °F . If the indoor temperature is lower than 55 °F, the AC will be activated for heating. This part is the most important parameters that allow user to change with great flexibility and will be used most often. In the right panel, user can choose the five different levels of mode that user want. According to the balance of the budget of user, he or she can choose whatever they feel is good for them and then the system will implement different mode to control AC units.

The Scheduler is a convenient function for the user to make the task for the system beforehand. A user can choose the time for a particular day which will range from 12am to 12pm and choose the duration for the task. If you want the HVAC system turns on for 60 minutes, you just simply choose 60 minutes for the duration. And in the middle of the screen, you will be able to choose the setting for the temperature and humidity as a reference point for heating and cooling at that schedule. After adding the schedule, the screen will show it on the middle and you can see the time, duration, and setting on it.

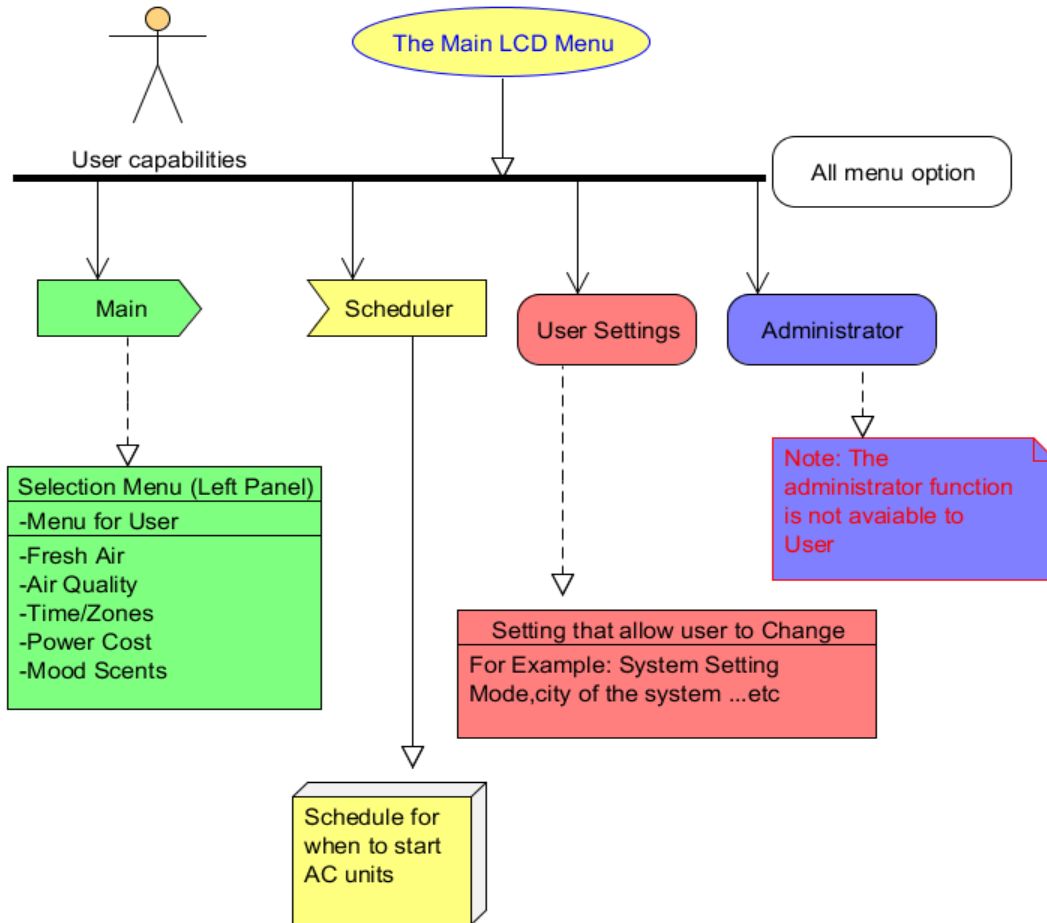


Figure 2.4: User Capabilities Diagram

Next to the Scheduler option is the User Settings box. The User Settings allows user to select turn on the HVAC system. If user chooses turn off, the connection between control units and AC units will be disconnected; otherwise, we should choose turn on selection. In here, a user can also see the setting for different mode and make the decision which one is the best choice. But the user cannot change the setting for tolerance level because this part belongs to the administrator's job, not the user's. On the bottom of the screen, a user can choose the location where the HVAC system is installed and also the city. Figure 2.5 shows the final design of generation two's HVAC system. The third generation sought to duplicate this as much as possible regarding our own project.



Figure 2.5: Screen display in generation two

2.8 Administrator Capabilities

Having a HVAC control and feedback system that can control all of its functions through LCD touch screen for users is one great capability. Taking this capability to the next level would be incorporating where an administrator could access the HVAC control and feedback system and make changes to the settings of the system. Also if there was an error with the HVAC system this capability would allow an administrator to have access to the system to attempt to make corrections to any errors that may have occurred. To gain access to the administrator capabilities there is a tab on our touch screen system to reach this function. However, the administrator tab is password protected in order to prevent anyone from accessing this tab and changing the settings to this system.

Once access has been gained to the administrator tab the user will have the option to make changes with the installed system components, selection options, and the tolerance levels, displayed in figure 6 below. The installed system components option allows the user to turn on or off the installed components of the HVAC systems. The components that our HVAC system will commonly use are AC1, AC2, Fan, Natural Air (NA), and Dehumidifier (DH). The selection options tool allows the user to make adjustments to following options fresh air, air quality, time/zones, power costs, mood scents. The tolerance level option allows the administrator to adjust the five preset tolerances that this HVAC system provides for its user. The Fresh Air option allows the administrator to select the duration that the Fresh Air can be turned on for ranging from a time difference of 5 to 40 seconds. Air quality selection option presents the administrator to choose which of the installed selected filters in the HVAC system to be turned on; in

addition to this the administrator can set the time that the filter will operate from a time of 5 seconds up to 40 seconds. Time/Zones selection option allows the administrator to select which zones in their building to currently be responding to the HVAC system and which zones to not be, as well as the time that these effects to occur. The power costs selection option allows the administrator to view the estimated cost of how much energy they consumed over a time period, there are no changes that could be made to this option it is only for the administrator to view. Mood scents selection option allows the administrator to choose which mood scent to have activated at the time and the interval of when the mood scent would spray. The five predetermined tolerance levels range from the most cost efficient tolerance level to the most comfort level and three more specific tolerance levels spanning in between being cost efficiency and comfort level. For each specific tolerance level the administrator is capable of adjusting the temperature difference along with the duration of time for each level. Finally once the administrator has made the necessary changes that need to be made to the HVAC system they will have the opportunity to save these changes, along with the power to kill the application and log out of the administrator tab. The administrator interactions with the project can be seen in Figure 2.6 below.

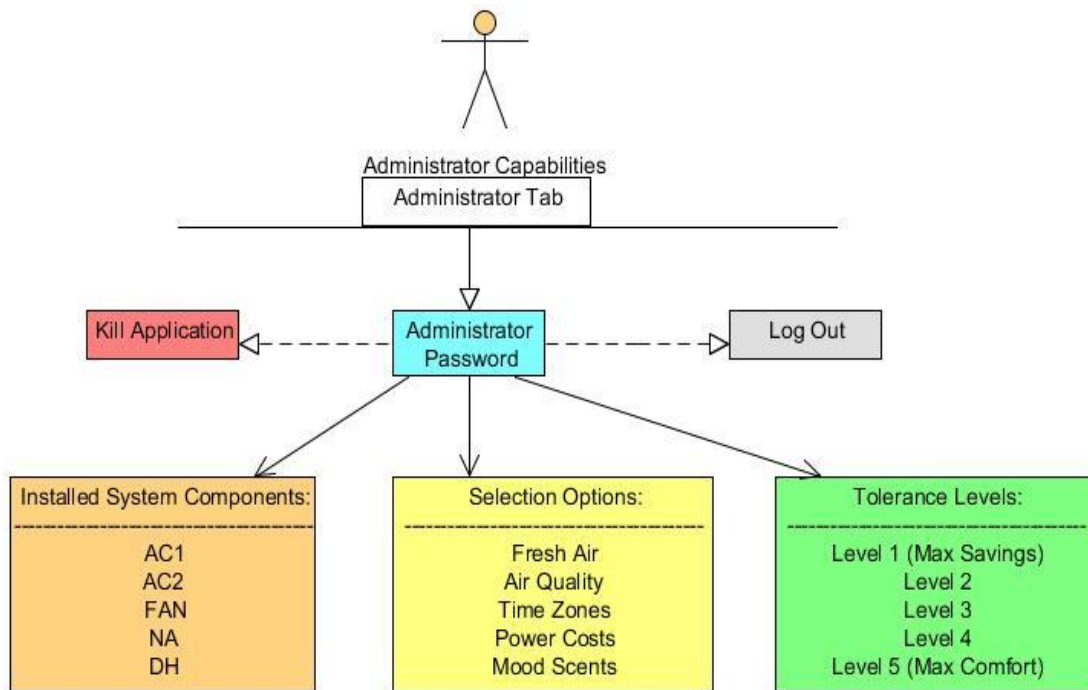


Figure 2.6: Administrator Capabilities Diagram

In reference to what we accomplished with these administrator capabilities of this HVAC system in relation to our project is to allow administrative changes to be made to the HVAC system without being in contact with the main control unit. Having this application is accessible via internet or through a phone application

will give homeowners the chance to make necessary changes to their HVAC system when they are not home and also for corporations and business buildings. Taking this capability to the next level it would also allow maintenance issues to be able to be taken care of more easily because if there is an issue to occur with this HVAC system and it is reported the company can take a look at the system to discover the error without being on site for either home or business. This way if the problem does not require sending maintenance man on site and could be resolved over the internet would save homeowners and businesses money for labor expenses. However if an error does occur resulting in situation needed to be corrected on site, the company would be aware of what the issue is when they do choose to send out maintenance to resolve the matter, making the trips more efficient and reducing the time it would take in order to correct the problem.

2.9 User Administrator Comparison

The capabilities that both the user and administrator propose to having over the HVAC system are almost exactly identical except the administrator is allowed to make changes to certain settings of the HVAC system that the user is only capable of just overlooking. First, starting with the capabilities that both the user and the administrator possess, they both are capable to make any changes available to either in the main tab menu, which ranges from setting the cooling and heating maximum temperature as well as the cooling and heating humidity. Also, well as making changes to the selection options available in the main tab consisting of Fresh Air, Air Quality, and Mood Scents. However, the changes that can be made to these selection options are only those available to be changed at the time. In the next tab, Scheduler tab both the user and the administrator possesses the power to set the times and temperatures for each day that they want their HVAC system to operate at. For the User Settings tab, the user and administrator can choose whether they want the HVAC system to be on or off, in addition to choosing the location that their HVAC system is in by selecting a state and given city within the state. Both parties are also capable of viewing the settings that each tolerance level is currently set at.

As stated earlier many of the options available on the HVAC system are both accessible to the user and the administrator, but there is an administrator tab on the HVAC system that only the administrator has access to because it is password protected. This tab allows the administrator to kill the application of the HVAC system, and set the degree difference and the time duration in seconds of each tolerance level to be activated; the user can only view these settings in the user tab. Administrator can also choose which of the selection options can be on or off and how they operate, as stated earlier in the Main tab. Whichever option has been selected on will be available to make changes as in the Main tab, otherwise it would be unavailable. Below in Figure 2.7 is a table of the tabs on

the HVAC system that the User and Administrator have access to, depicting the different availabilities between different users.

Tabs	User	Administrator
Main Tab	✓	✓
Scheduler Tab	✓	✓
User Settings Tab	✓	✓
Administrator Tab		✓

Figure 2.7: User/Administrator Comparison

2.10 Roles and Responsibilities

In order to divide the labor wisely and equally, it was necessary to assign our labor according to the individual's strengths and interests. The group consists of two computer engineers and two electrical engineers. Each individual focused on certain and separate parts according to our strengths to make sure the work was finished on time. Dividing the labor enabled each member to specialize and focus on their parts of project as shown in Figure 2.8 below.

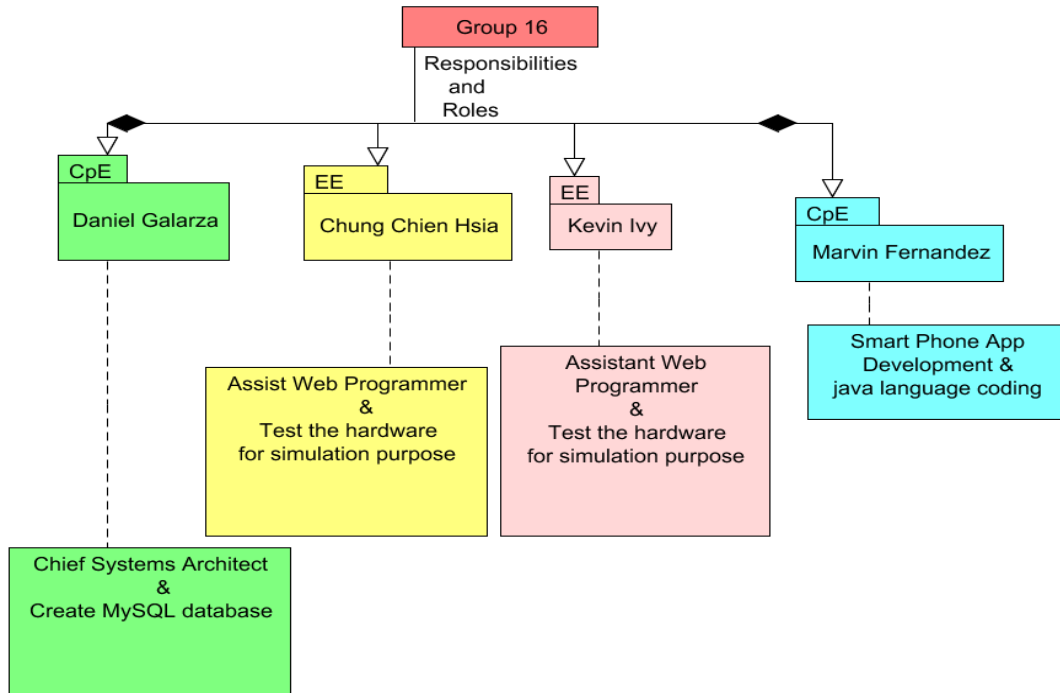


Figure 2.8: Roles and Responsibilities

Daniel is a computer engineer and has a Bachelor of Art in Digital Media. He has some experience building websites in HTML and CSS and making them capable of accessing databases with PHP. This makes him the ideal choice for working with the Internet components of the website.

Daniel Galarza

- Create MySQL database
- Develop web programming with PHP for dynamic web pages.
- Use AJAX to update the screen automatically
- Teach Kevin and Chung HTML, CSS, some PHP for creating web pages
- Select, purchase and utilize all necessary web services.

Chung Chien Hsia is an electrical engineer. He has good circuit analysis and circuit design and also has good trouble shooting techniques. Other than the hardware training, he also has the basic C/C++ knowledge and knows of some Java language. His interest is focus on power systems and analog filter design.

Chung Chien Hsia

- HTML for web pages design
- CSS to improve the web pages
- Learn and understand PHP for database interaction.
- Microchip operation and simulation

Kevin Ivy is also an electrical engineer. He has a good history of dealing with circuit analysis and circuit design, as well as the testing of circuits. Outside of his history with hardware design and analysis, he has received basic training in C language. His interests are to pursue work related in the field of telecommunications systems.

Kevin Ivy

- CSS to improve the web pages
- Learn and understand PHP for database interaction.
- Learn to use an SQL based server to develop the HVAC data structure
- Microchip operation and simulation

Marvin Fernandez is another computer engineer. He is approximately equal experience in both electrical systems (circuit design, circuit analysis) and computer programming. Has experience in both C and Java languages. He has an interest in further developing his programming skills , as well as his website development skills.

Marvin Fernandez

- Smart Phone application

- Java language coding
- Supporting role in the web design and development

2.11 Coding Breakdown

The project is almost completely software based, so all four group members was responsible for software coding at some point. Kevin and Chung are Electrical Engineering students, thusly, they were in charge of the hardware aspect of the project. Daniel is not only a Computer Engineering student, but has experience with a BA in Digital Media, Internet and Interactive Systems Track. He has experience building websites with HTML, CSS, PHP and MySQL. This makes him the most ideal member for researching and developing the web infrastructure needed for the project. It was his responsibility to help Kevin, Marvin and Chung learn web programming languages to help him build the web sites and background infrastructure for the project. Daniel does not have much experience with XML, so to get everyone on the right foot with the most advanced standards in web design; he will teach them XML while learning himself, so the whole website can be built with XML from the ground up. This helps set up the website for AJAX in the future as well. They started off by building the basic look of the web pages for the browser based control system. Daniel worked on the server based infrastructure and use the more advanced languages, PHP and MySQL. Marvin has more experience with Java. Java is a popular language for building smart phone applications by far. He was in charge of the smart phone application for controlling the HVAC system. During fall semester, Daniel learned Java, and by spring semester, he was able to assist Marvin with anything he needs to produce the application. If the phone application is done with Mobile Web Site architecture instead, all four members worked together to create it with Dreamweaver. The team produced a simulated HVAC system to imitate the properties of the upcoming edition of the HVAC hardware designed by Tim VanderMey of Proton Design Inc. Rather than recreating the entire touch screen HVAC system, the new hardware simply imitated the next system's internet connectivity. Daniel, Kevin and Chung did this by programming the PIC32 microprocessor in the C language.

2.12 Project Milestones

Each topic on the graph represents a important marking point for the project's schedule and the relative time it should have taken to accomplish that section. These topics made up the primary points of interest that determined our progress for the next following months. It was important to stay as close to this plan as possible and to not fall behind. A tentative schedule is displayed below in Figure 2.9 for both semesters.

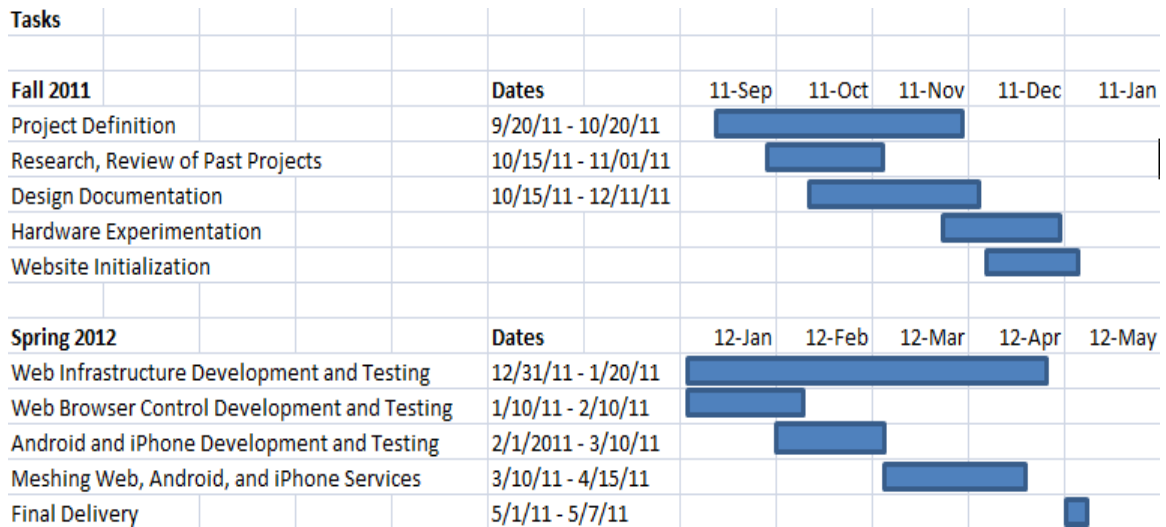


Figure 2.9: Milestone Schedule

2.12.1 Fall Semester

Fall Semester was strictly a research part of the project. It took almost the entire semester to truly identify the project, due to difficulties contacting the client and sudden changes in Natural Air. Late into the semester, the team was suddenly informed that all the hardware of the project was outsourced to Tim VanderMey of Proton Design Inc, because Natural Air wanted an updated version of the project as soon as possible to show to shareholders. Instead, Natural Air wanted the phone application to change from being an optional part of the project, to the top priority. This was not deemed a bad thing, as the project simply changed, rather than simplified. In the initial definition of the project, the initial mission of the team was to produce a more mass producible HVAC system while building an internet service for the system, with the smart phone application being an optional part of the project. Instead, all the hardware of the project has been outsourced and replaced additional software requests. Now, the smart phone application is the goal we fulfilled. Initially, Daniel and Marvin became the technology leads for the team, with Kevin and Chung following their choices. But then in November, Team 16 got the terrible news from Tim VanderMey that his contract with Natural Air would be on hiatus. Tim gave the team his choices in hardware for this theoretical third edition of the HVAC hardware, with the PIC32 microprocessor at the core. Kevin and Chung were suddenly given the reigns in inspecting these hardware choices over winter break.

2.12.2 Winter Break Preparation

Although the fall semester will be exclusively for the summer, Team 16 intends to take full advantage of the month-long break between the two semesters. In order

to hit the ground running in January, steps were taken to set up all the necessary pieces of the system to be created. First, the hardware intended to simulate the next edition of the HVAC touch screen hardware was added to the project late into the fall semester, and will require extra study. This hardware was studied by Kevin and Chung so it can later communicate with the web components, with Daniel assisting with them brushing up on C, the PIC32's language. Daniel will purchase the necessary web services for the project and request reimbursement, potentially including a domain name (arbitrary or chosen by Natural Air,) and a web hosting service, most likely GoDaddy.com. All four members will get familiar with the latest edition of Dreamweaver. Daniel created the template for the web based HVAC controller, by hand or with an open source template. Kevin and Chung worked with Daniel in getting familiar with basic HTML, creating forms for each page based on Chung's design choices for user input. Daniel then decorated the site with a basic CSS sheet to make the webpage mimic the design of the original HVAC touch screen system, because Natural Air greatly enjoys the aesthetic look of the touch screen's GUI, and did not want it changed. Marvin will also use Dreamweaver, and familiarized himself with its latest features for developing mobile web page applications, such as its adjustable resolution options. Daniel and Kevin worked together converting Kevin's abstract data structure for the HVAC system, into a true SQL based data structure.

2.12.3 Spring Semester

The spring semester began the execution of the design developed in the fall semester. In January, Marvin started the development of the MySQL Database, while Daniel developed the necessary web pages with HTML, CSS, PHP and SQL. By the end of January, the web pages had the basic structure resembling the touch screen interface from the previous senior design group's project. At the same time, the browser HVAC controller was developed. What is important is the flow of the testing (see Project Testing.) The web server was tested to see if it can receive every possible status contained within the HVAC system (reading). From there, the browser HVAC controller was refined so that it can *send* commands to the HVAC system (writing). A mobile version of the website was tested so that a user can control their HVAC system from their phone as if they were in front of their HVAC system's touch screen panel. First the standard web browser was tested, then the mobile phone version was tested to make sure it could imitate all the same functions as the desktop browser application.

2.13 Financing and Budget

The table below depicts the budget for this project. The previous version of our projected budget was \$1500, in order to assemble a new version of the HVAC system. The most important part of our budget is the first one in the table. In order to add internet infrastructure to the HVAC system, we must pay for two

different kinds of web hosting services. First, one web hosting service will be chosen to hold the web browser based HVAC control system and MySQL database. A second web hosting service may be chosen to store the database infrastructure. If the group were strictly building a website with not server based programming, a simple web editor such as Dreamweaver, or even the free Mozilla Komodo Edit would be suitable for the project. However, this project requires heavy PHP and SQL programming. Daniel used this software during studying Digital Media in 2007, and recommended acquiring this software for the group's purposes. A basic Wireless G Router of any make and model is part of the budget as well. This was eventually free since Daniel owned one. In the spring semester, the group bought parts that simulated different parts of the design similar to the new prototype hardware for the HVAC system created by Proton Design Inc. This means that the group may have to simulate the HVAC system with an entirely software based imitation of the HVAC touch screen controller. A wireless router would be connected to a PC or laptop to detect incoming commands and send and receive data to and from the MySQL database.

Late into the semester, a wide variety of hardware was added to the project as well. The following hardware additions are based on the direction Tim VanderMey of Proton Design Inc. is taking with the redesign of the hardware. They allowed the team to mimic the hardware scenario that is most likely to be the newest edition of the HVAC hardware. At the core of the needs is the PIC32 Microprocessor, which serves as a significantly cheaper alternative to the previous processor, the Panda Board, which was \$275.00. To give the PIC32 internet connectivity, an I/O Board, Ethernet Daughter Board, and Wi-Fi expansion were also a part of the budget so we may simulate the new hardware's wireless internet capabilities. This brings the original cost of the project up from \$450.00 to a new total of \$924.00. This new budget reflects new equipment to meet new circumstances with our sponsor and requirements for the project. Not only did the project require hardware, but a very important mentor, Tim VanderMey of Proton Design, will no longer be a part of our project. The team must achieve Natural Air's objectives without his assistance. This new budget includes hardware choices made by Tim VanderMey before he concluded services with Natural Air. These hardware choices are intended to make a more cost effective and mass producible version of the project created by Team 16's predecessors. These parts are the PIC32 USBII Starter Kit, Ethernet Daughter Board, PICtail Plus Epson Board, and I/O Board Expansion.

Our project requires a website, so Dreamweaver was purchased for coding in HTML, CSS, PHP and SQL/MySQL. Originally, PHP and SQL would be coded by purchasing Zend Studio, but Dreamweaver has advanced so much in PHP development that Zend Studio is removed from the budget. Also removed from the budget is the need for web hosting services. Natural Air gave the team access to their account with Host Gator, which will hold the web sites and databases we produce without interfering with Natural Air's currently existing

website. The newest additions to the budget are the parts that will mimic the work of the senior design team before us. They include the Temperature/Humidity sensors, personal fan, Mood Scent Dispenser, VPack 5 Volt, Jumper Pins, LEDs, and PCB boards.

Team 16 intended to finance this project two different ways. The main way is through Workforce Central Florida, which has offered to provide funding to Senior Design projects related to hot issues in the 21st Century, such as National Security, Conservation of Energy, Renewable Energy, Biochemical sciences, and so on. Our project applies to the conserving electricity, and thus was eligible for funding from Workforce Central Florida. Below in Figure 2.10, you can see the budget laid out to Workforce Central Florida.

Expense	Part Number	Provider	Price
Temp/Humidity Sensor (3)	SHT21	Liquidware	\$143.51
LM73-Digital Temperature Sensor		National Semiconductor	\$.53
Personal Fan			\$30.00
Mood Scent Dispenser			\$15.00
VPack5.0V_AA		Bodhilabs	\$11.95
Jumper 2 Pin (10)	PRT-09044	Sparkfun	\$3.20
Basic LED (10)	COM-09592	Sparkfun	\$3.20
Secondary PCB		4PCB	\$61.32
Main PCB (2)		4PCB	\$66.00
Electrical Test		4PCB	\$65.00
Wireless Router	N/A	Cisco Linksys	\$50.00
PIC32 USBII Starter Kit	DM320003-2	Microchip Tech.	\$55.00
Fast 100 Mbps Ethernet PICTAIL Daughter Board	AC164132	Microchip Tech.	\$50.00
ZEROG 802.11 Wi-Fi Board	AC164136-4	Microchip Tech.	\$60.00
LCD TFT 7" 800X480 White Touch Panel	NHD-7.0-800480WF-CTXI#-T	Newhaven Display International	\$107.00
Graphics Controller PICtail Plus Epson S1D13517 Board	AC164127-9	Microchip Tech.	\$119.99
Graphics Display Truly 7 800x480 Board	AC1641279	Microchip Tech.	\$269.99
I/O Board Expansion	DM320002	Microchip Tech.	\$72.00
Dreamweaver	N/A	Adobe	\$130.00
Shipping			\$65.00
Total			\$1378.69

New Budget			\$1378.69
Old WFC Budget			450.00
Difference Requested			\$928.69

Figure 2.10: Senior Design Team 16 Budget

Section 3: Research

3.0 Research Related to Project Definition

Daniel has experience with web design and web programming, because he has a Bachelor of Arts in Digital Media, in the Internet and Interactive System track. Therefore he took advantage of his experience and researched the old languages he has studied in the past and how to implement them in the project. He also researched TCP/IP, AJAX and REST, because he was uncomfortable with those architectures. He has the most experience working with the internet, knows how to make websites require passwords, but in terms of security that is the initial extension of his knowledge. He is also receiving a military position in the Cyber Operations branch of the military, and will make this a part of his career. In combination of the fact that Natural Air wants the product to be as secure as possible, he also researched system security. Kevin and Chung, being Electrical Engineers, researched high-level software developing strategies, such as UML diagrams for representing the behaviors and capabilities of users, administrators, the database, and all software and hardware interactions. They researched the history of the project in general, where it came from, and where Natural Air came from. Despite the project nearly being entirely software, the software developed must interact with the hardware developed, so they are also researching the changes the hardware has experienced in its history, and how it's currently being updated. They both researched the properties of the PIC32 microprocessor in particular. Marvin researched wireless communication, because the team does not know how the HVAC system can connect to the internet. In particular, he researched the ZigBee and Wi-Fi protocols that the new HVAC prototype will use to access our web architecture. He also researched smart phone application development. Below, in Figure 3.1, is a summary of the team's respective fields of study for this project.

Daniel Galarza, CPE	Project Implementation, Web Programming, Databases, System Security, Wireless Communications
Marvin Fernandez, CPE	Smart Phone Applications, Wireless Communications
Chung Chien Hsia, EE	Project History, Hardware, PIC32 microprocessor, Web Programming
Kevin Ivy, EE	Databases, Hardware, PIC32 microprocessor

Figure 3.1: Researching Strategies Summarized, divided by Team Members

3.1 Research Strategies

One of the most important aspects of the project is the research phase. Different technologies will be evaluated and examined to decide which will best fit the needs of the project. Major project decisions were made at this phase, such as which specific technologies to use. There were factors that determine the value of a specific part. These include cost, ease of use, the need to learn new languages and/or interfaces and capability with other devices or technical software. The decisions made here affected the next phase of the project, which is the actual design plan for the project. It was vital to review as many alternatives for each part of the project as feasibly possible. Otherwise, potential hardships can arise from using equipment that isn't entirely suited for the best performance of the project. Since the research to be performed must be as broad as possible, each member of the group must contribute to the research phase of the project design. Most of the information needed for research was split up within the group according to the members' roles within the project. Then all the research was put together to decide which parts are actually going to be used for the project. A specific design was then made that covers the technologies to be used in the project.

3.2 Existing competitor products

One of the ways to begin research on this project was to look at and study other products in the market that are similar to this one. The next sections will cover other products the group has researched and studied and compare the similarities and differences between them.

3.2.1 Honeywell Prestige 2.0® Comfort System

Even though our HVAC system has lot of new features adding by us or the sponsor, there are still some competitor products of our HVAC system. William Caron argued that they had no competitors, and that was the reason they wanted their intellectual property undisclosed. Daniel argued to the team that an absolute lack of competitors had bad implications that there were truly no competitors; there may be no market for what Natural Air was trying to produce. However, it did not take long for Team 16 to find similar products when searching for them online. Some parts of the design are little similar to competitors product and other parts are different. It was better for us to know the existing competitor products and analyze the difference between our design and their products. We found some products that were designed by Honeywell International Inc. have some similarity with our HVAC system. The model that we think is the closet to our HVAC system is Prestige 2.0® Comfort System which have High Definition, full

color display with touch screen monitor so it is easy for user to control using fingers. It can control the HVAC system over the internet from a computer, smart phone or tablet PC. They also equipped with wireless outdoor temperature sensor to detect the outdoor temperature and humidity. It also includes with a portable remote control for their HVAC system to control temperature anywhere in the home. It has 7-day programming that allows user to program for each day of the week. The company claimed it can save energy up to 33% on annual energy costs or up to \$200 per year. The monitor display is as Figure 3.2 below. One thing need to pay attention is that the figure only shows black and white but the real product is HD with full colors.



Figure 3.2: The residential use main menu (Permission Requested from Honeywell International Inc.)

Main menu contains 4 tabs on top of display so user can choose Home, Fan, System, and Menu. It displays the temperature and humidity inside the building and also displays the temperature and humidity outside the house. On the right of panels contains the adjustment for temperature setting. If you tap on the date on the screen, it will allow you to change the time. It allow user to choose the different setting for fan by simply tap on the tab on top. It contains three modes, and first one is automatic which fan runs only when heating or cooling system is on. The second mode is on that the fan is always on. The third option is circulate that fans runs randomly roughly 35% of the time. The reasons that the company claims that they can save up to 33% energy cost is because they have include the default setting as figure below. However, one of the drawback for this setting

is that the temperature setting is fixed for the maximal setting. Our design would not have that problem. Below in Figure 3.3 is a sample of these features.

Residential Use	Period	Start time	Heat (Mon-Fri)	Cool (Mon-Fri)	Heat (Sat-Sun)	Cool (Sat-Sun)
	Wake	6:00 am	70°	78°	70°	78°
	Leave	8:00 am	62°	85°	62°	85°
	Return	6:00 pm	70°	78°	70°	78°
	Sleep	10:00 pm	62°	82°	62°	82°

Commercial Use	Period	Start time	Heat	Cool	Fan
	Occupied 1	8:00 am	70°	75°	On
	Unoccupied 1	10:00 pm	55°	85°	Auto
	Occupied 2*	12:00 am	70°	75°	On
	Unoccupied 2*	12:00 am	55°	85°	Auto

* Period 2 is cancelled by default. If you activate it, the values shown above are default settings.

Figure 3.3: Energy Save Settings for maximum cost savings

If user presses the system tab on the top of menu, then a user can choose from 5 different system settings-Heat, Cool, Automatic, Off, and Emergency Heat. Those will give the user more flexibility of their HVAC system. If you choose Heat option, thermostat controls only the heating system and Cool option only allow thermostat controls only the cooling system. However, if user chooses Automatic, thermostat selects heating and cooling automatically. If you choose the off option, cooling and heating system will be turn off but not the fan. Fan will still operate normally if fan is set to on or Circulate mode. Emergency Heat option use thermostat control auxiliary heat.

Menu contain many functions include holiday/vacation scheduling, setting of dehumidification, ventilation, and Installer options. In ventilation, user can choose from three operating settings- On, Off, and Auto. The installer option is really similar to our administrator option that only allows the qualified technician to change the parameters and settings there with the valid password.

Most parts of the main menu are really similar to our design. For example, our main menu will display the inside and outside temperature, and the adjustment to temperature setting. However, we will also allow user to change the humidity setting by making a new selection buttons just next to the temperature settings. This will make the control more convenient compare with the competitor products. Besides, their products will require user to choose cooling, heating, or automatically. However, our design will work automatically, user just needs to set

the range of the temperature for cooling and heating. Then the AV unit will be turn on for cooling or heating according to the current temperature in the building.

Even though, the product claimed that their HVAC system can be controlled by internet of smart phone devices, they required to buy the internet Gateway. It will give the user remote access to the thermostat via internet or from smart phone. However, they didn't show this features in the manual or the website. This is reason why our project had some features that different than theirs. First, we don't need to purchase any internet gateway because as long as you have your personal router, you were good to go to control the devices. Second, their saver setting depended on the default setting that regulates the fixed temperature for cooling or heating. It is not so flexible and our design went with another route that can best fit to user needs.

Their product has some features that we don't have for now. They can allow users to lock the menu for advanced security. It contains 3 lock options – unlocked, partially locked, and fully locked. If user chooses fully locked, then any access to HVAC system will be blocked. Partial locked will only allow user to be changed. Unlocked is the default setting that user can change anything they want. This function can prevent unauthorized change of HVAC system and it is increase the safety of the residents. Overall, we all want to achieve the same goal which is save money, save time and more security to the residents.

3.2.2 BAYweb

Daniel introduced the project to Dr. Sumit Jha, who expressed immediate interest in the topic of modern, energy conserving HVAC systems. Dr. Jha revealed to the group that he not only took an interest in modern HVAC systems, but that he owns one. Dr. Jha purchased HVAC Thermostat products and services that are superior to everything Group 16 is trying to achieve.

BAY Controls is an LLC company located in Maumee, Ohio that sells such products and services. First, customers purchase one of a variety of thermostat devices depending on the HVAC hardware their building has. Then, with the 29

thermostat successfully installed, the user makes a free account on www.bayweb.com. This account allows a user to control their air conditioning unit from home, to set up a seven day scheduler for their HVAC system, lock and unlock the controls on their device, and view graphs of their power consumption throughout a variety of time intervals. All these web services are provided free of charge to customers who invest in the thermostat and internet module for their building. An internet module is connected to the business owner or homeowner's router (therefore requiring a broadband internet service as well as a router in the home. Not only can this application be accessed from a desktop browser, but also a mobile device. BAY Controls LLC started off selling these units exclusively to business venues, but they now sell units to individual homeowners, and Dr.

Jha is one of them. Below in Figure 3.4 is a sample screen of their web HVAC application.

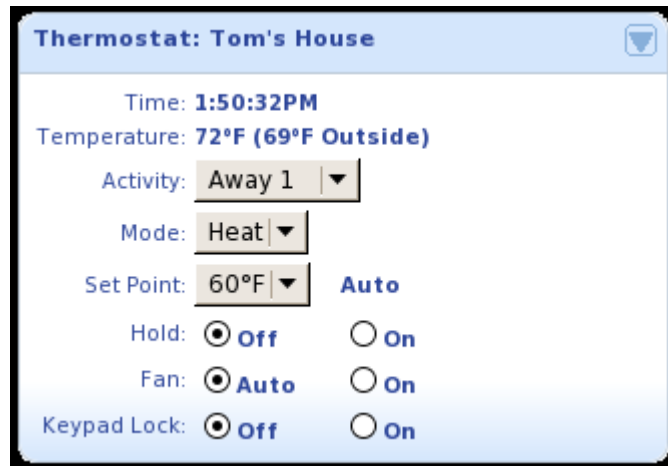


Figure 3.4: BAYweb.com web HVAC application (reprinting permission granted from Bay Controls LLC.)

3.3 Web Infrastructure

Completion of this project requires a strong understanding of how the internet stores, retrieves and sends data. The HVAC system needs to be able to send the temperature, humidity and settings to the Internet so that users can then control their home comfort via a smart phone. First, the project must define the clients and servers in the entire system. This requires that the entire system to send and receive data through the Internet, so understanding the nature of how all the entire systems devices will connect to the Internet will be a major part of producing the results needed to complete the project.

We determined that there are three clients and two servers for the entire system. Clients are the devices or software that connects to the internet, and servers being the web services and software need to send and receive information. If the clients are the devices that need to be able to connect to the Internet to send and receive data, the first client is the HVAC system itself. Originally, it was part of our group's objective to give the old HVAC system connectivity to the internet. Instead, Tim from Proton Design Inc. is redesigning the hardware to improve upon the work of the previous senior design team to give the device internet connectivity. Eventually, we developed our own hardware the connection testing. The second client is any smart phone (with an iPhone or Android operating system) that downloads the application with the purpose of accessing the user's respective HVAC system. The third client is a browser running a web application from a computer to manipulate the HVAC system in a similar fashion to the smart phone application.

The project requires two different servers for sending and receiving data through the internet. For the project to succeed, the HVAC system and the smart phone application must be able to communicate with each other. Linking these two separate systems is the first server. This server's purpose is to send and receive data from the HVAC system to the browser and smart phone application. This server will consist of a MySQL database. This database will contain every variable, input and output that the HVAC system contains. These variables include the temperature and humidity of the home. It also contains the current user settings applied to the HVAC system, and newly received inputs from the user via the browser or smart phone application.

The second server holds a different purpose than the database. This server holds a website application so that users can control the HVAC system from a computer the same way they would if they had access to the touch screen directly. This server stores all the HTML, CSS and PHP files that will compile into the user website. The website will be given its appearance and structure with HTML and CSS, so that it looks as similar to the original HVAC system's touch screen interface as possible. The PHP files serve as the programming necessary to communicate the user's commands to the first server, as well as retrieve updated information. This means the first server must be able to communicate between the HVAC system, the smart phone application, and the web application, while the second server must be able to communicate between the web application and the second server. If possible, both services together on the same server would be ideal, saving the team, and ultimately the client, money.

3.4 Choosing Service Providers

In a traditional dynamic web page design, only a server and domain based service would be needed. Proton Design suggested that this traditional model is insufficient for meeting the needs of the project. For Daniel, the web programmer of the Senior Design team, this new upcoming model of hardware internet communication is his least understood subject. These new styles of services could be useful in making the project much simpler, as these services have things like uptime guarantees and maintenance.

3.4.1 Thingspeak

Thingspeak, a service from ioBridge, is a free, open source hardware communication service that uses the web programming language Ruby on Rails. Being open source and free to the public sounded like it would be ideal for our project, however, its documentation revealed that its infrastructure allows a mere eight fields of data. The project would require many more, as the data structure being developed for a single customer could reach near one hundred variables of data. Close inspection of Thingspeak leads to a better understanding

of why it could be useful to our project. It provides a variety of web applications to collect and display data. It provides simple code to perform posts and queries to its database, as well as code to display data in the form of its many applications. Below in Figure 3.5 is a sample of an HTML embedding of streaming data from the account Daniel created.

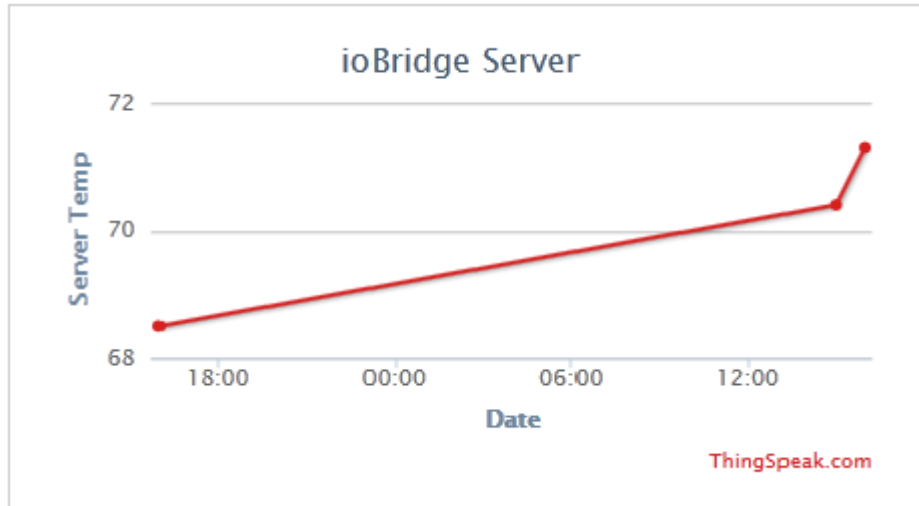


Figure 3.5: Sample Thingspeak embedded graph – reprinted with permission from ioBridge

3.4.2 ioBridge

ioBridge is one of the more famous “Internet of Things” services. It offers a wide variety of hardware and software options for clients seeking internet connectivity between hardware devices and the internet. It sells many hardware modules with different purposes depending on the appliance. It even has temperature sensors and HVAC systems intended to work with its online infrastructure. Its online services come in two different forms: OEM and Professional. OEM allows existing customer hardware to connect to the Internet where uploaded data can be observed via a database and provided JavaScript based software. OEM services cost \$11.99 per three months, and are a reasonable expense. If our client wants to permanently keep our work at the conclusion of this project, they can purchase an unlimited account for one flat fee of \$149, which would be suitable for a long term commitment. Professional is the alternative service, and is not suitable for this project because it is intended to be used in conjunction with the hardware they sell. This project team had no intention of buying sensor hardware, because that is not part of our project objective. Instead, it’s subsidiary Thingspeak was a better preference.

3.4.3 Pachube

Pachube.com is another big name in hardware internet connectivity. It sells a variety of intriguing hardware sets for use exclusively with Pachube. For example, the Current Cost Bridge allows individual users to detect their power consumption and upload it to Pachube's existing infrastructure. In a sudden stroke of luck, as of October 31st 2011, Pachube became a completely free service. On paper, this sounds like a fantastic choice in the short run because it is free and has many nice software features for detecting data from hardware. The problem is that if the client chooses to expand upon our work using Pachube, there will be a point where it reaches the website's limit of 100 API transfers per minute. Pachube is going through a very recent transitional phase, so perhaps it will sell more advanced services for Natural Air in the future.

3.4.4 Go Daddy

Godaddy.com is a popular web hosting service provider that provides all the standards in web development hosting for traditional websites such as domain names, hosting, and MySQL databases. It is quite inexpensive, at \$4.99 per month in increments of 3 months, one year, two years, or three years. This fee provides unlimited bandwidth, ten MySQL databases and 10GB of space, which is more than enough to complete the project and allow expansion. Even if Go Daddy is not used for the needs of a database, a traditional website is still one of the parts of the project, and Go Daddy would make an excellent host for it. A single domain costs a onetime fee of 11.99.

3.4.5 Hostmonster

Hostmonster is another popular web hosting service, and one that Daniel is familiar with. It sells hosting services at 6.99 per month in increments of 1 years, 2 years, and 3 years. Unlike Go Daddy, Hostmonster provides one free domain name as long as it is unused. Additional domain names can be created for ten dollars each. We were going to go with Hostmonster for our web needs, but Go Daddy turned out to be less expensive and provided more payment options. Daniel used this service once and had issues with their billing cycles, so the team will not use this service.

3.4.6 Host Gator

A potential opportunity that Natural Air has provided the group is the ability to use their own web hosting services on Host Gator. Natural Air eventually allowed us to use it, so Daniel set up folders on the domain www.naturalaireenergy.com, and created hidden folders so that the Carson' potential customers do not see the web applications that are in development.

3.5 The Internet of Things

As technology advances, the way the Internet is used has changed. In the nineties, the traditional client server model dominated the Internet. It still does, but in new ways. The difference is now, in the 21st century, inanimate devices big and small connect to the Internet regularly to provide users with information. Blu Ray players, Sony's successors to DVDs, connect to the Internet to regularly update their consistently expanding firmware. Cellular phones can check email, a traditional use of the Internet. A home security system can automatically contact 911 if an alarm is triggered in a home without homeowners calling the police, just in case there is a true emergency. The core to making this possible is Wi-Fi. Wi-Fi is a variation of the internet protocol called IEEE 802.11. It allows devices large and small to connect to a wireless network within specified ranges varying from a computer connecting to a household router, to a computer connecting to a campus wide network connecting users to the internet. Proton Design Inc. refers to this as the Internet "of things." Small devices using the Internet the way regular human beings would.

The HVAC system was redesigned specifically so it can access the internet the way a personal computer would. In order for it to send its status to a web browser or phone application, and especially for receiving commands from a cellular phone, it needs to be able to access the internet regularly. The device needed to be able to connect to a MySQL database to regularly send the temperature and humidity to the building where it's installed. It must also connect to that database to receive new commands such as adjusting the room temperature. How can a device like an air conditioner use the Internet? Part of it sits outside a building, and inside is a small device that users manipulate to control it. What sort of system can get all the hardware and software components communicating with each other over the Internet?

To begin, the air conditioner itself must be able to connect to the HVAC system. This can be done with a simple radio. Proton Design Inc intended to give the hardware a simple 900 MHz radio to allow all hardware to communicate with the HVAC system interface. The radio will be used by the outdoor sensor to send the status of the outdoor temperature and humidity to the indoor system. Next, there is the HVAC system itself. Its hardware was redesigned to give it ZigBee wireless communication protocols. Just as the Senior Design team before us favored Zigbee, Proton Design fully intends to use it to its fullest capabilities. ZigBee is a wireless network protocol intended for small devices that need to consume as little electricity as possible. In keeping with Natural Air's desire for conserving electricity whenever possible, this wireless protocol is perfect for allowing the HVAC interface to access the internet while consuming little power. In fact, there is an edition of ZigBee called ZigBee Smart Energy, which is intended for detecting electricity use. ZigBee was an integral part of the design of the new HVAC system's microcontroller hardware.

3.6 Web Browser Interface

Before the Smart Phone application can be made, the Web Infrastructure needs to be thoroughly tested through other means. Although starting with making the smart phone application is very much possible, it is not a good design practice. Before the smart phone application can be made, it was a better idea to make sure all the web services are working first. This was done with a web browser application that serves the same purpose as the smart phone application. Such an application will serve the same purposes as the smart phone application, but will be operational on a desktop or laptop personal computer.

The term “web application” refers to the fact that it will be a website to the core, built with technology typically used for web pages. The website will be built with HTML, CSS, PHP and MySQL. The HTML and CSS code serves to give the application it’s look and structure. It was used to create web pages that will be completely equivalent to all the different screens possible on the HVAC system’s touch screen. The main menu, administration, and additional features on the HVAC system will be available on a web browser. CSS was used to give the application its look and feel. Due to time constraints, CSS was only used to give the web page its basic structure so that a user can easily understand the menus and options. Structurally, it will resemble the HVAC system’s screen; however CSS will allow us to give the application all the same colors and aesthetics of the touch screen. Due to time constraints, the aesthetics of the application were low priority. Instead, making sure the application is simple to understand and use was the priority to the application. Luckily, there was time, so the application was updated to be more pleasing aesthetically.

Also taking priority over the aesthetics is the web programming behind the structure of the website. A MySQL Database was the core to making the entire project work as intended. This database sends and stores data at regular intervals, and immediately, a smart phone was not ideal for testing all its capabilities. The web application had PHP and SQL coding not visible to users, existing to make a more robust website that can successfully access the database. This allowed the website to reveal the status of the building’s temperature and humidity, as well as all the current settings. It should be no different than if the user was in their home, using the HVAC touch screen.

First, the website was constructed with HTML and CSS giving as much of the look and feel as the HVAC touch screen that time allowed. Then the PHP and SQL was added to at least allow the user to see the status of their HVAC system. Finally, additional PHP programming was added to allow the user to make changes to their HVAC system without the user needing to touch the original HVAC system’s screen. When all of these features had been successfully created, the team moved on to creating the smart phone application, with full confidence that if a web page can use the HVAC system, then a cellular phone can use the HVAC system.

Ironically, the way the web application works, in a sense we would have created an application for the phone. Modern smart phone technology includes web browsers that allow users to send and receive data no different from a browser on a desktop or laptop computer. However, this is less than ideal, because manipulating a web page browser on a phone can be very inconvenient when a website is not optimized for mobile phone use. The two major options for a smart phone application will be either to optimize the original browser application for mobile phone use, or to make a Java based smart phone application from scratch. The second is more ideal, because a Java smart phone application will be optimized for use with a touch screen, and therefore more professional, and easier for users to utilize.

3.7 Coding Languages and Protocols

3.7.1 HTML

HTML stands for Hypertext Markup Language. It is the most basic, barebones language for producing a website or web application. Children using Live Journal or My Space in the turn of the 21st Century learn HTML and they didn't even realize it. As such, learning HTML is a very basic stepping stone for web development. It is a mere "word processor" for the Internet, creating things like text, paragraphs, and bulleted lists. For this project, it was one of the very first languages used in developing the software to complete our objectives for the HVAC system. Therefore, it was used to create all the text, links, buttons and forms for the web application, with the intent to mimic the HVAC touch screen's features as quickly as possible. The results created by the HTML coding was initially entirely static and unable to display what is going on inside the HVAC system's temperature and humidity sensors (see PHP and MySQL below.)

3.7.2 XML

XML stands for Extensible Markup Language. It sounds like an advanced version of HTML, but it serves very different purposes from HTML. While HTML is meant to display data to users (such as text), XML is meant to transfer data. XML helps create "metadata," by adding groupings, tags and attributes to data in HTML based documents, providing additional readability and grouping of data. Metadata is unique from the standard idea of data, in that it merely gives properties of that which it tags in between brackets. You see metadata every time you click on an mp3 file of a song you like and fill in the genre. Daniel and Marvin ultimately learned XML in preparation for using AJAX.

3.7.3 CSS

CSS stands for Cascading Style Sheets. It was the second language needed for the project, and of all the technologies and languages needed for the project, it

has the lowest priority, but was needed regardless, because it is the language that gives HTML Object Oriented Programming capabilities. It allows every paragraph, line of text, button and form its shape, size, color, and behavior when a user clicks on it. If completely unused, a website will be completely white, and the font and size will vary tremendously based on the user's browser, and browser's setting. While these things seem purely aesthetic, when the PHP and SQL programming is added to the application, the appearance of the HTML based text and forms will fluctuate wildly, with words spilling on top of other words because text boundaries and sizes are unspecified. CSS is needed to give the HTML structure so the web application is easy to understand regardless of browser or anything the MySQL database throws at it. However, styling the page so that it looks nice is a very low priority compared to making sure the page looks functional. There are plenty of free-to-use style sheets available on the Internet, and they will be considered because our prototype web applications are for educational purposes at the core.

3.7.4 PHP

PHP is the one of the backbone languages for programming on the Internet. HTML alone is merely word processing for the Internet. It cannot send and receive data. In other words, it creates static web pages. PHP on the other hand can create dynamic web pages. With PHP, web pages can change what is visible to users based on what information is provided by the user, or what information is provided in the database. Without PHP, HTML can create the forms and buttons the users need to manipulate the HVAC system, but they will not do anything. PHP will provide us with the ability to interact with the database we create. In a sense, PHP is like programming in C++ for the Internet. PHP is invisible to people using the internet, because the language is run on the server, not the client web browser. The example below shows the source code of an HTML form on the left, and to the right is all the PHP code on the same file but invisible to the user. This also makes PHP powerful and effective for web security. Below in Figure 3.6 is an example of HTML running on a user's computer, with hidden server side programming occurring with PHP.

<pre> <html> <body> <form action=searchform.php method=GET> Search for: <p> First Name: <input type=text name=fname size=15 maxlength=15> <p> Last Name: <input type=text name=lname size=15 maxlength=15> <p> <input type=submit> </form> </body> </html> </pre>	<pre> <?php mysql_connect (<MySQLserver>, <username>, <password>); mysql_select_db (<dbname>); if (\$fname == "") {\$fname = '%';} if (\$lname == "") {\$lname = '%';} \$result = mysql_query ("SELECT * FROM <tablename> WHERE fname LIKE '\$fname%' AND lname LIKE '\$lname%' "); if (\$row = mysql_fetch_array(\$result)) { do { print \$row["fname"]; print (" "); print \$row["lname"]; print("<p>"); } while(\$row = mysql_fetch_array(\$result)); } else {print "Sorry, no records were found!";} } ?> </pre>
---	--

Figure 3.6: HTML Form (left) and its hidden PHP programming (right)

3.7.5 SQL and MySQL

SQL stands for Structured Query Language. PHP alone can provide dynamic programming capability to web pages. In a sense, it's like C Programming for the Internet. But alone, it cannot send and retrieve data from a database. Web databases communicate with SQL. Like the "Q" implies, SQL performs queries, also called requests, to send or retrieve data from databases, so that they can be communicated to websites in the previous languages listed. All the languages work together in the following fashion. Although learning SQL will be necessary, our actual writing in SQL will be limited to queries within PHP. The database was required to be built with SQL, yet it was not feasible for the team to do so by hand, which is done by carefully writing all the properties of the database table, every element, and all the elements lengths and properties.

Instead, a powerful tool called MySQL was used. Web servers with MySQL capabilities allow web programmers to specify what variables and database structures will go into the database. After designing the data structures needed for the project, the team was able to list out the MySQL variables that go into the HVAC system. The service will then automatically generate the database in SQL, while displaying the SQL generated to the designers so they can perform tweaks. As long as a developer can describe their abstract data needs in the form of strings, integers, and other basic variable types, MySQL can easily store it in a database. While hard-coding the database is possible, it's not effective or even feasible to do so when such a powerful tool is available to us. MySQL is the most popular open source database in the world, and was an obvious choice for our database needs. However, MySQL is no substitute for SQL. Individual SQL lines needed to be coded in conjunction with PHP each time the user needs to

interact with the database, such as downloading information or performing an update.

3.7.6 Java

Java is an Object Oriented Programming language useful in building applications. It is very popular for making smart phone applications. Marvin and Daniel, being Computer Engineering students, are familiar in Java. It has many powerful features that make it a powerful tool for developing Graphical User Interfaces, making it ideal for making our smart phone application. As of fall semester 2011, Marvin has learned Java and Daniel is currently learning it. It was the most likely choice for producing the smart phone application for the project.

3.7.7 JavaScript

JavaScript is a web programming language like PHP, created by Netscape Communications in 1995. A major difference is that PHP is server based, while JavaScript is client based. A user's own browser handles JavaScript code while PHP code is used by the server containing the code's files. JavaScript and Java sound similar, but serve different purposes. It was created in 1995 as a part of Netscape 2.0, out of the desire to coordinate HTML pages with embedded content. Initially, this was exclusive to embedded Java applets. But it can be used for many more things. Like PHP, it can also provide dynamic programming to web pages. If JavaScript is used, it would be used near the end of the project in conjunction with PHP (see AJAX below.) It was not used to interact with the database, because that is the role of PHP. PHP and JavaScript both require programming logic but serve different purposes in web applications. An additional use of JavaScript will be to create special interactivity that traditional HTML forms cannot provide. In particular, one of the HVAC system's most basic buttons is classic "thermostat" buttons for changing the requested temperature and humidity. There is no such form button to properly represent them in HTML. Instead, customized buttons will be created and programmed in JavaScript to represent them.

3.7.8 AJAX

AJAX stands for Asynchronous JavaScript and XML. Rather than a programming language, it is a system architecture. It is a combination of HTML, JavaScript, PHP and CSS that gives web pages the unique ability to self update without requiring a refresh from the browser. Typically, the content of a web page on one's browser is static. In order to change or update data, it needs the user to click on a link, and then the browser will access the server, which will then send data back to the client browser. AJAX is what allows websites such as Gmail, Facebook, and Twitter, to update itself regularly and endlessly without the need

for the user to select Refresh on the browser. It's what can also allow the user to go onto their browser, send a command to lower the browser and immediately get a response from the HVAC system. Without AJAX, every single command on the touch screen transferred to a browser will require a Submit button. Due to time constraints we had no choice but do have a submit button for everything, but ideally, we wanted to implement AJAX so the browser and touch screen behave and update just as quickly as if the user were right in front of the HVAC touch screen in their home.

3.7.9 REST

REST is short for Representational State Transfer. API is more modern than AJAX. It is another system architecture that allows instantaneously dynamic web pages for faster communication between devices. Like AJAX it uses XML for the basic web page coding. REST offers simplicity that AJAX does not have, and it will be considered if it can help us give the browser and smart phone application better capabilities. REST was considered for the interactive main page of the web application.

3.7.10 C#

C# (pronounced C Sharp) is an object oriented programming language. By the name, one would think it is an updated version of C. This is not the case. Instead, it is an alternative to Java for object oriented programming. It was created by Microsoft to improve upon the weaknesses of Java and C++. Tim VanderMey from Proton Design Inc. highly recommended this language for our project. It was considered an alternative to Java for the smart phone application. It could of also serve to build a simulated HVAC system that can send and receive inputs the way the actual control system would, because our project was mostly software based, yet we needed to interact with hardware which we might not always have.

3.7.11 Ruby on Rails

Ruby on Rails is unique web architecture. It can be used for a variety of purposes, such as quickly generating web applications in a short amount of time. It provides tools such as project managers, SQLite database production, and mass production of web application components (called scaffolding). It provides quick ways of coding in many parts of website building that many find basic, such as email.

3.7.12 C

C, one of the first programming languages many students learn, was the programming language of choice for the hardware. It does not have as many

options as Java, but is very reliable. The PIC32 microprocessor features a free, unrestricted C compiler, so C will be a natural choice for controlling the hardware aspect of the project. Unknown before the project research, C has libraries to connect to a database, no different from a web page. The C library needed for this is called `mysql.h`, and it provided the ability to connect to the database created for this project. Below is an example of how accessing an SQL database is just as possible for a C program as it is on an HTML/PHP based web page. This example below includes the necessary `mysql.h` library, establishes a connection and performs a single query to a database. This example below in Figure 3.7 includes the necessary `mysql.h` library, establishes a connection and performs a single query to a database.

```
#include <my_global.h>
#include <mysql.h>
int main(int argc, char **argv)
{
    MYSQL *conn;
    conn = mysql_init(NULL);
    if (conn == NULL) {
        printf("Error %u: %s\n", mysql_errno(conn), mysql_error(conn));
        exit(1);
    }
    if (mysql_real_connect(conn, "localhost", "zetcode",
        "passwd", NULL, 0, NULL, 0) == NULL) {
        printf("Error %u: %s\n", mysql_errno(conn), mysql_error(conn));
        exit(1);
    }
    if (mysql_query(conn, "create database testdb")) {
        printf("Error %u: %s\n", mysql_errno(conn), mysql_error(conn));
        exit(1);
    }
    mysql_close(conn);
}
```

Figure 3.7: SQL connection and query written in C

3.7.13 jQuery

jQuery is a relatively new web programming language and architecture, meant to be an alternative to JavaScript. In reality, it is a massive JavaScript library, meant for performing the same functions of JavaScript in a similar fashion. Consider CSS for a moment, in its ability to assign visual and (to a limited degree) behavior properties to elements in a website, whether they are words, colors or paragraphs. jQuery takes this idea a step further. Rather than add properties to web page elements, jQuery adds entire functions to web page elements. This is called Functional Programming. At the recommendation of another Senior Design group, Daniel tried using it initially as an alternative to creating alternative Cascading Style Sheets for the mobile phone application. He found jQuery to be incredibly powerful. jQuery has a mobile phone library, called jQuery Mobile, that styles webpages specifically to be used by mobile phone applications.

After trying jQuery, it took Daniel a significantly small amount of time (roughly a few days) to create all the necessary pages that the mobile application would

need, and with each of those pages looking and behaving like a real, hand coded phone application. jQuery Mobile allowed Daniel to quickly made phone-ready buttons, sliders and forms, each of which were touch screen ready. Daniel also used the standard jQuery library to handle all necessary AJAX protocols, which allowed the phone application to update the database asynchronously. jQuery proved to be an indispensable part of the project, and Daniel will study it further after graduation.

3.7.14 TCP/IP

Going deeper into the hardware side of the project requires an understanding of how hardware connects to the Internet in general. It was vital to learn and understand TCP/IP because the PIC32 microprocessor's creator, Microchip Technologies Inc, provides free TCP/IP stack software, which was vital to getting the simulation hardware to connect to the Internet. The fantastic part of studying this protocol was the fact that Tim VanderMey of Proton Design, and the previous Senior Design team, both intended to use PIC microchips for the same reasons. In that sense, in our project's study and use of the PIC32's free TCP/IP Stack libraries, the work of all three parties came together. Both Proton Design Inc and the previous senior design team emphasized the fact that this free software was important to the project, making it vital for study.

TCP/IP is one of the oldest and most important protocols of Internet development. TCP/IP is also known as the Internet Protocol or Transmission Control Protocol, and without it, the Internet would not have been possible. It dates all the way back to the 1970's in which the Defense Advanced Research Projects Agency (DARPA) developed it as the successor to ARPANET, the first packet-switching network developed during the Cold War. The way the Internet works in the 21st Century is the result of this protocol. For example, TCP/IP is what assigns every network client an IP address. Each IP address is composed of four numbers separated by a period. Each number is eight bits; therefore an individual IP address is composed of 32 bits. (Just like the PIC32 microprocessor.)

The protocol is divided into four abstract components, called Layers. The layer that Team 16 was most familiar with was the Application Layer, and was a comfort zone for team. It is the part of the protocol that includes applications that communicate with users, and the needs of users towards the Internet. Higher level languages and protocols reside in the application layer, such as HTML and HTTP, which exist to allow human users to interact with it, creating, manipulating, or retrieving data which can be understood by them. The web browser HVAC system, touch screen HVAC system, and smart phone application are will all be parts of the Application layer. The Application layer is considered the "highest level" of the layers, as it requires the highest levels of languages.

The Transport Layer, lower than the Application layer, allows data from sources such as a user's application or a server to be transported across a network. This layer observes and keeps track of the flow of data, its sources and destinations. It makes sure that data moves from one location to the other in order, and organized, so it can be easily understood as it moves from source to recipient. It does this using TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) packets. Each application that is part of the network is assigned one of thousands of different TCP/IP ports for transmitting and receiving data. The Transport Layer also manages lost data. The layer has timers for when packets of data are sent and when they are meant to be received. If packets of data do not arrive on time, they are resent through the network. The transport layer can also control the flow of data between applications. If one component of our system is more advanced and faster than the other (a smart phone versus the PIC32processor), the layer can have the flow of data slowed on one end to match the other.

The Internet Layer is lower level than the Transport Layer. It can be called the Internet in a sense, as it's what makes the Internet we know possible. It has three basic purposes. It gives packets of data the necessary implementation to reach its destination. It detects errors and diagnoses problems with connections. It is the lowest layer of the four that deals with connected devices in a network.

The lowest layer level is called the Host to Network Layer (also called the Link Layer, or Network Access Layer.) It is the only layer that does deals with the physical and logical connection of devices in a network. For example, it manages LANs (Local Area Networks), and Token Ring Networks, providing the necessary data transmission protocols based on the needs of the network. All these layers are represented in Figure 3.8 below, organized by hierarchy.

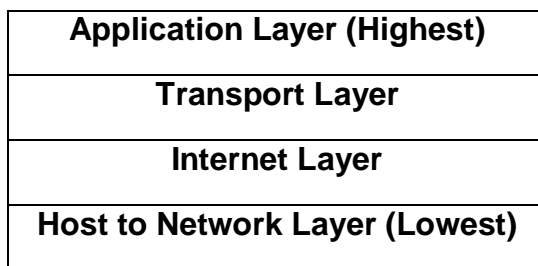


Figure 3.8: TCP/IP Hierarchy Summary

Microchip provided a free TCP/IP stack for their PIC18, PIC 24, dsPIC and PIC32 microprocessor lineups, already optimized and easy to use. The diagram below summarizes the structure of the TCP/IP Stack software. Figure 3.9 below shows the hierarchy of Microchip's TCP/IP Stack in a similar fashion to 3.8.

- Application Level Protocols: HTTP, FTP, TFTP, SNMP, DHCP, StackTask
- Transport Layer Protocols: TCP, UDP,

- Internet Layer Protocols: ICMP, IP, ARP, ARPTask
- Host to Network Layer Protocols: Ethernet MAC, SLIP

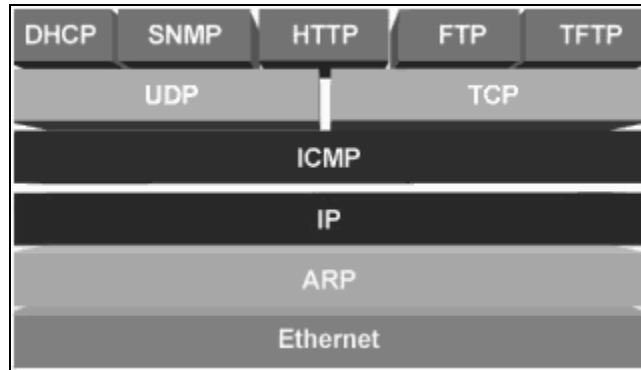


Figure 3.9: Microchip TCP/IP Stack for PIC18, PIC24, dsPIC & PIC32, reprinted with permission from Microchip Technologies

3.8 OSI

The TCP/IP architecture model is one of the oldest and most widely used. It has limitations due to its age, such as only being able to describe networks that use the TCP/IP model, such as the PIC32. This means it cannot describe modern wireless architectures such as Bluetooth or the famous 3G and 4G networks our smart phones use. The most famous alternative to TCP/IP is the OSI model. OSI stands for the Open Systems Interconnection, and the most famous feature that sets it apart from TCP/IP is that it is composed for seven layers instead of four. Furthermore it is adjustable to meet the needs of the hardware and software to create a network. An OSI model can describe TCP/IP but not the other way around, because any unnecessary layers of the OSI model can be unused, while the TCP/IP layer is limited to its four layers.

The Application Layer is the “highest layer,” just like in the TCP/IP model, and represents software with which users can interact with a software and hardware system. It uses high level languages and application software.

A new layer, the Presentation Layer, is below the Application Layer. This new layer is responsible for taking data from the Application Layer, such as inputs from a user on an application, and making it “presentable” to the next layers for proper display and communication. It is second in height to the Application Layer, and therefore associated with high level languages, yet it is also the lowest layer that developers work on data structures in a network. In other words, the MySQL database that Team 16 creates would apply to the Presentation Layer, because when user produces input, it is not understood by the MySQL database immediately. It must be submitted, formatted, and properly stored in the database. This layer has other excellent features, such as data encryption.

HTTP suddenly fits within this layer, unlike in TCP/IP.

The next OSI layer is also unique to OSI, and is called the Session Layer. The Session Layer is responsible for connections between devices. This includes establishing connections, maintaining connections once they are connected, and terminating them when communication is complete. These connections seem simple, but they can be as long as a web conferencing session between networks nationwide, or even live news broadcasts. In Team 16's case, they can be as short as a POST or GET command from a user submitting a form to a shopping cart. TCP/IP does not have any accurate representation of this layer, because it does not worry about the metadata (details) of a transport protocol. The Transport Layer from TCP/IP is also included in OSI, and remains just as vital to both models, and the Internet itself. Like TCP/IP, it provides reliable transfer of data by providing control over errors and proper flow of data.

The Network Layer comes after the Transport layer, and it allows data of various shapes and lengths to be properly transferred from a source, to a destination. This layer responds to the transport layer's requests, and performs its own requests to the data link layer, many layers above it. The layer is responsible for providing the models for different network connections; host addressing, and message forwarding. It includes such crucial hardware such as Ethernet or wireless routers. This layer is considered to have much in common with the Internet Layer of TCP/IP. But a major difference is that TCP/IP's Internet Layer can only describe the Internet itself, while the OSI Network Layer can describe a variety of different network protocols.

The layer below the Network Layer is the Data Link Layer. It is equivalent to the lowest layer on TCP/IP, the Host-to-Network Layer. TCP/IP would not normally represent the relationship between hardware and networks in this fashion. It works with the Physical Layer below it in transferring data between network vertices (or nodes) in LAN or WAN networks. It also detects errors in the physical components in the network, such as user computers and hardware. It is what allows hardware to transfer data to a network.

Finally, the Physical Layer is the lowest layer in the OSI model, to the point that it is even "lower" than the lowest level of the TCP/IP model. While TCP/IP does not talk about the physical properties of a network, OSI can represent hardware in a network. This can be as primitive as a computer's binary communication and the transfer of individual bits from a computer, to a router, to the other higher level layers of the OSI model. In total, OSI has three more layers than TCP/IP that can be added or removed as needed, providing additional ways to provide proper metaphors for the structure of a network. Figure 3.10 below displays the layer structure of the OSI model, as well as comparisons to the equivalency between OSI and TCP/IP.

Application Layer (Same)
Presentation Layer (New)
Network Layer (New)
Transport Layer (Same)
Session Layer (New)
Data Link (Same)
Physical Layer (New)

Figure 3.10: OSI Model Layer Structure, and comparisons to TCP/IP

3.9 Smart Phone Application Technology Research

Most of the research of the mobile phones was made using a comparative list. It listed many different kinds of software and technologies that are currently used to develop mobile phone applications. The lists compare these types of software by the languages used, mobile platforms supported, cost, and other features. From this list, several kinds of software were examined and considered as potential supplements to be added to the project design. Several potential software products will be covered in detail in the next sections.

3.9.1 Overview

One of the main goals of the HVAC system interface is to have accessibility to the system via the phone. An effective way to achieve this was to develop an app that displays all the same information as the main website. Thus the app would look exactly as the HVAC system touch screen would if the user was accessing the system from his or her home. The first thing to determine is what phones will be accessing the system. In the specifications section, it was decided to create the phone app based on the iPhone and Android, since they are the two most used phone devices for consumers. Thus, to create the app, the technology to create mobile phone apps is needed to begin development.

A benefit is that the system was mostly a copy of the main website in the Internet. So the completion and access to the website was desirable before development of the phone app begins. One of the main steps to begin iPhone/Android development was to have the right software to develop it. Software that can use UI and other technology for phones is the first step to develop the system. Once these software kits were obtained, programming can begin to ensure the app has all the same functionalities as the main website. This

means that every button and feature is included for users and administrators, as well as the same layout of the HVAC system itself.

Finally, testing was done to insure the phone app obtained the data from the HVAC system and it was accurate. Any changes a user might make must register to the system was tested for accuracy as well. All data must be accurate and consistent among all three interfaces which was tested as well. The testing format for the three parts of the project will be further covered in the testing section of the design document.

3.9.2 Specific Definitions

A software development kit, or SDK are tools used by a majority of software developers to create applications for mobile phones. SDKs allows third party users to develop applications and games for their specific technology. The SDKs are usually made up of several different software products. This includes a specific programming language, a IDE, a debugger, sample codes, tutorials and more. For the purposes of this project, the main two SDKs to focus on are the iOS SDK, and the Google SDK. These are used respectively in the iPhone and Android brand of mobile phones.

The iOS SDK is mainly used in the development of Apple's iPhones applications. The main language used for the iOS SDK is Objective-C. Objective-C is a modified version of the classic C programming language, and is used in most of the iPhone's apps. Objective-C is object oriented, and follows the rules of the regular C language. It's main difference is the inclusion of message passing, a style of programming that doesn't call methods, but sends messages to the receiver. The receiver decodes and interprets the message to obtain the instruction needed. However this also reduces the speed at which methods are computed.

The Google Android SDK is also designed for third party users. The primary language used for the Android SDK is Java. It also includes the IDE Eclipse and is cross-capable with most major operating systems such as Windows, Mac, and Linux. However there are various alternate SDKs developed that can be used in place of the main Google SDK. Each has its' own pros and cons and may or may not be considered, depending on the limitations of primary SDK.

3.9.3 Technologies Considered

In terms of actually developing the application for the phone, there are advantages to it already being a website. Most of the code for the phone app will be the same as the website itself. The only things that changed is the format and size of the phone app compared to the website, as additional coding was needed

to ensure the app behaves exactly as the main HVAC system device. Most of the coding would be in Java and it may have been necessary to use Apple's Xcode to create the application. Software that can implement Xcode will need to be purchased or obtained elsewhere. Another issue is that Xcode is usually only compatible with Objective-C, a unique language that Apple uses. The software to be used may have been needed to convert Java into something compatible with Xcode, if possible. Otherwise, an alternate means may have been necessary, because having to implement two different coding schemes for both the Android and iPhone is wasteful, inefficient, and bad design. Luckily, the Android SPK supports Java for its coding scheme in its programming and is free from their website allowing no need for budget issues.

Xcode

Some of the main tools needed to development on a iPhone are the iOS operating system, and the Xcode development kit. Xcode is the main IDE for iPhone applications and uses the Objective C programming language. Any changes needed to the website would need to be implemented in this IDE. The Xcode provides all the necessary functions for apps such as view controllers. One issue is that the Xcode may need the iOS SDK to run which would require buying the SDK from Apple. Also there was the need to learn another programming language combined with the languages required for the main website. An alternate software could have been substituted if applicable since this prototype app isn't being marketed to the public.

The main development software of Android requires the use of its SDK and functions primarily on Java for its code. It also uses Eclipse for its main IDE. This is useful as the technology is familiar; to shorten the time needed to learn new interfaces. However, while it uses the Java language, it doesn't use all of Java's APIs which is something considered during the coding process.

PhoneGap

One potential form of software we could have used is called PhoneGap. This software allows a website to be hosted on multiple platforms. This extends to not just the iOS, but Android and OS as well. This also allows the website to be configured simply by html and/or JavaScript. Finally the software was free to use allowing no budget problems to surface. Disadvantages for PhoneGap are that all applications become hybrid, the apps are not native, and layout rendering is done by the webview. Also the Apple SDK and Xcode is still required to develop a iPhone app.

Aqua

Such alternate software to consider is Aqua from UX+. Aqua is a free software development kit for mobile phones, which is suited for the purposes of this project. It runs on JavaScript or C/C++ and can use IDEs such as Eclipse and Visual Studios. It also runs on a variety of mobile platforms as well. It also provides a simulator which is useful for testing purposes.

The Aqua software kit main advantages was the ability to rely on only one programming language to design a phone app capable of working in multiple mobile phone operating systems. This allows more users to have access to the product and not just serve users who own a certain type of phone. Since it allows the use of widespread and common IDEs, there no need to spend time learning how a new interface works before actual coding begins. Another advantage is the simulator included in the product software. The application software can be tested and debugged for problems before the app is actually created, improving the performance of the product.

FeedHenry

The next software kit considered was FeedHenry. It is software based on cloud mobile application development. One of the main advantages is there is no need for SPK downloads to develop apps. It uses traditional languages HTML, JavaScript, and CSS, allowing flexibility in coding. It is also capable of supporting various mobile operating systems as well. It also includes the ability to plug in to Eclipse for familiar coding environment.

Dreamweaver

Dreamweaver was a famous web development software tool created by Macromedia, now owned by Adobe. Dreamweaver is a program made for website design. Containing both visual and coding representations of the website, it is a powerful tool able to easily edit and design websites. It features includes JQuery integration, supporting various JavaScript libraries. One major point is its ability to represent the visual aspect of several different devices, such as mobile phones, tablets, etc. It is capable of showing exactly how the website looks on different devices at the same time, so a developer can see the effects of every change he or she makes.

As mentioned before Dreamweaver has support to develop on many different platforms. This is done using the Multiscreen Preview panel. It displays the website as it would look under variously different devices. Each panel's display can then be edited without affecting the display outlook of the other panels.

Other languages supported by Dreamweaver include CSS3 and HTML5 for future coding. A Design view provides rendering support for the website, and a Live view allows for video support using Quicktime and tags. The Live view includes a Webkit rendering engine allowing for real time rendering support for website development. Dreamweaver, however, is able to support many more languages such as XML, XHTML, AJAX, PHP, and ASP. This variety provided a very flexible environment to completely develop the website.

Dreamweaver takes advantage of the PhoneGap technology in most of the development of mobile phone applications. This makes it easier to develop for both the iOS and Android mobile systems without having to code for each individually. So, once the code is developed in HTML and settled for the main website, Dreamweaver uses PhoneGap to create apps for the mobile phone systems.

Along with these features, Dreamweaver comes with other peripherals as well. There are several code hinting programs for the various languages included, to help developers who still may be inexperienced with website design. Forums are available for those who seek specific help on potential problems. W3C validation is also included with Dreamweaver to ensure the accuracy of a website. Dreamweaver provides plenty of resources to ensure a secure website development project.

In conclusion, Dreamweaver is very helpful in website development. Since it is applicable for websites in the World Wide Web, and most mobile websites development styles, it can reduce the amount of time needed to code both sections of the project by a significant amount. It will be possible to code for both sections at the same time, even if both websites may have various differences between them in the end. It also eliminates the problems concerning using multiple mobile technology formats. A single website can be designed, with minor changes made to fit each style of mobile phone out there roughly. This helps make sure any user will have access to the mobile website, which is a very solid advantage in a competitive market environment.

3.9.4 Mobile Web Site Development

One frustration that reoccurs during designing the mobile app is that a designing an app to run on various platforms is difficult due to different designs using different languages, browser, servers, etc. A possible bypass to this problem is to develop a mobile design of the main website. There are several ways to go about this, and hopefully one would suit the needs of the project.

One way was to provide users access to both the main website page and the mobile website page. If, for some reason, users are dissatisfied with the mobile version of the website, or the mobile website is unable to display on that user's

particular phone, they can always access the main website instead. A mobile website main advantage is that it is usually formatted better to fit on a possible mobile phone. Websites designed for desktops and laptops usually have a very clunky look in phones, and other issues like text sizes, heavy content, unnecessary scrolling, and small buttons. A mobile website designed specifically for a phone can be designed so users can have an easier time accessing the website without hardships. As stated, there are several things to considered if one wants to create a effective mobile website as opposed to a regular one. The next section will list several of the best practices used to design mobile websites.

3.9.5 Best Practices

There are several different engineering practices that have become efficient in developing mobile applications. These methods are considered to be the most efficient and superior way to develop with few hassles and have a well-developed product. There methods have developed as a result of cataloging hundreds of different companies and developers techniques and what gave the best results from these techniques. Thusly, these are known as the best practices of mobile website development. The following sections will begin discussing some of these methods which may prove useful in designing the mobile website of the project.

1. Using Cookies Sparingly

Cookies store data that a website uses, especially for a client using the website. However the cookie needs to connect to the server for every request sent, which may badly affect performance on a mobile website. So it is better to use as few cookies as possible to achieve ideal performance.

2. Replicating local data to server

Data unique to clients on a website should be stored to the server. This allows the same data to be accessed from the mobile and main website, and allows a way to save changes to the data in the event of a crash. Data that would have been lost or damaged can be recoverable from the main server. Since the project requires that a individual's data be accessed and consistent on both the main website and mobile app, it is important to ensure the data for a client is stored securely within the server database.

3. Mobile Security and Privacy

The mobile phone application needs as much protection and security as the main website, especially when the website contains personal information on a consumer. JSON is a common tool used to transfer data to a specific client. However, a direct execution of the data feed is not secure at all and is unwise to be left alone. This can be fixed using a JSON parser, instead of parsing the data by directly executing it.

4. Automatic Sign-In

After an user has signed in to the mobile website, an option to enable automatic sign-in may be beneficial. This way, an user does not need to type in the username and password every time he or she accesses the mobile application, especially since it is more difficult to type on a phone than on a computer. A cookie or local storage can store this data, but a password should be encrypted or have a hashed token for security.

5. Using File Compression

A simple way to save data space and improve performance is to use file compression. One way to do this is to use HTTP 1.1 compression software. Its algorithms for compression are widely supported, allowing flexibility and ease of use. However it is important to realize that not all formats benefit from file compression, which should be considered in the design.

6. Reducing Data Size

Another simple way to save data space is to reduce the size of the data itself. Smaller applications execute and run faster than larger ones. Reducing whitespace and code in the application can increase performance overall. More complex techniques such as globalizing variables can also improve and reduce data space as well.

7. Larger Network Requests

Connecting to the server usually is heavy on resources and time consuming. After the mobile website send larger data amounts saves performance and resources in the long run. Thus if possible, it is helpful to send few large data requests instead of lots of smaller ones. Other methods including prioritizing requests and combining small requests into bigger ones can help with this problem.

8. Using Sprites

Individual buttons that each have a separate image usually causes unnecessary HTTP requests. If there several buttons that have the same style and image, they can be combined into a single sprite. Thus, only a single HTTP request is needed to upload all the buttons to the website. This can also work on similar images as well. However, this shouldn't be used on informational images, as it causes additional problems.

9. Background Images in CSS style sheets

A way to avoid an additional HTTP request for background images is to include them in base64 encoded strings within the CSS. This adds a bit more space in the data but allows background images in the mobile website without causing severe slowdown for the user.

10. Cache AJAX data

Any data that uses AJAX accesses should be cached alongside the primary content of the mobile website. This is simple, as the standard caching techniques will work on AJAX data just as well as using it on the primary content pages.

11. Not Using Cookies Unnecessarily

Any static sources can be obtained using a different domain, sub-domain or path name from the main application cookies. This saves time and performance as static sources don't need the cookie information and thusly don't need to be updated along with the cookies.

12. Reducing Start-up time of a Application

An important part of a application's quality is the length of time it takes for the application to start. There are various techniques that can be used to reduce this time so the application can start much faster. Offline technologies such as HTML5 AppCache have the ability to take start-up data and stores it locally so it can start-up much faster. Partitioning large scripts and minimizing local storage queries also can help reduce start-up time significantly and be considered into the design.

13. Perceived Latency

Perceived latency can cause users to believe that the website has crashed when it hasn't, which leads to a bad experience overall for users. Reducing this so it doesn't happen will help improve a user's experience with the application. Enabling incremental rendering to the website allows the website to load useful information first so it is available to the user while their waiting for the page to load. Instead of loading the whole page for updates, it is better to update the pages dynamically by using the DOMs. Finally spinner progress bars can be used to show progress in loading the page, this way the user is ensured that the page is really loading instead of being frozen.

14. Multiple Interaction Methods

A mobile website can have various ways in which users will interact with the application. There are currently three main methods: Focus-based, Pointer-based, and Touch-based. Focus-based is the browser focus literally "jumps" from one element to another element. Pointer-based is more similar to the mouse in terms of its function. Touch-based is obviously based on the touch of the finger or stylus. It is important to know which type of method the website will incorporate, especially if it will use two or all three styles for any type of user who will potentially use the website.

15. Application Views

Most smart phones allow a application to change views depending on its orientation. Some websites do this by showing or hiding sections of content on their pages. This causes some problems however. The browser back button

ends up not working as it should and it is not possible to link to a specific view. A URI with a distinguishing fragment identifier can determine the view so it can assign the correct orientation to display.

16. Consistency of State

This practice states that any relevant data a user may input into the website is capable of being seen in other device, such as the mobile website. It is mostly done by saving all user information into a server which can access and update all the different forms of the website.

17. Meta Viewport Element

Some mobile phones have websites zoom in to compensate for their smaller size. A mobile website which is already designed for such phones may be distorted by this function. The meta viewport function is the main setting for the display and should be changed to 100% in order to avoid this problem in these phones.

18. Choice of Interfaces

It may nearly impossible to account for every possible interface a user may potentially use for a mobile website. If the website has multiple interfaces available, it may be helpful for users to decide which interface they would rather used. This gives the user more choice in the matter for a better experience. It would also be helpful to store a user preference in interface within a cookie. Otherwise, use the most appropriate UI on a user's first use.

3.9.6 Issues to Consider

Languages:

While more and more phones are becoming increasingly compatible with HTML and its variants, there is the possibility that a user will have a phone that can't support it. Thus that user will be unable to take advantage of the mobile application. A older language, such as WML can be used, but that will require more development time and resources. CSS3 is another popular choice of language but again, not all phones can support it.

One issue that can come up is images and the amount of storage space they take. While images present a pretty component for a website, they can also use up a lot of information space to display them. Since phones have much less storage capacity then desktops and laptops, this can lead to much longer loading times than what is necessary. A careful design of the mobile website will need to be considered to ensure that the mobile website isn't wasting more data than what is needed trying to load up too many pictures.

Incapable Technologies:

One important obstacle is that there are many different styles of mobile phones in the world. Even the two most popular systems, the iPhone and Android use different languages, browsers, and other designs. A mobile application tailored made to look perfectly for every phone will take far more time than the project will allow. Nevertheless, it was important that the website will display decently in most phones, if not better. Software that allowed development on multiple mobile phone formats will be invaluable to the project.

3.10 Wireless Communication

3.10.1 Overview

One of the main functions of the website is to communicate with the HVAC system wirelessly. This function of connecting wirelessly would ideally fall upon the server eventually selected to run and maintain the website and database. The server will store and update all data in the database which is then used and displayed by the main website. The first case is that the server can obtain data from the PC simulation of the HVAC system and display it on the website. Afterwards the connection between the system itself and the server had to be evaluated to ensure its performance reflects the same as the PC simulation. This connection had to be tested as the HVAC system may be different from the PC in how it connects to the online database. All of the wireless communication between the two eventually depended on the final model of the HVAC system and its technology.

3.10.2 Relationship of Web site and Database

The primary function of the database is to store all data that comes and goes to the HVAC unit. The web site to be created for the project has to receive this data and display it to the user on a computer. Thus models for the communication between the website and database is needed to accurate to obtain this information. The classic model of wireless connection is the TCP/IP model of wireless communication. Widely used in the Internet, the TCP/IP model provides a solid and reliable means to connect two devices wirelessly. This is useful for connecting the database to the web site. The same system can be applied to connecting the database to the HVAC system, or for the purposes of this project, the PC emulator of the system. Consisting of four or five different layers, it is a hierarchical structure, with each layer providing different functions.

The server to be used as the database is responsible for controlling the connections between the PC emulator/HVAC system and the web site itself. This is useful as it eliminates most of the work needed to create a TCP/IP communication system. Issues such as obtaining IDs for the devices and the most of the coding behind the system are taken care of, allowing time to focus on other areas of the project. One important factor considered was to ensure that changing the data on the Internet actually changes the settings of the HVAC system unit. There is no purpose for the web site if it is incapable of actually affecting the system from the Internet. The connection must have been able to send data both ways. One technology considered for this connection is the Zigbee Model, which uses a communication model similar to the standard TCP/IP model.

3.10.3 Zigbee Protocol Model for Communication

One of the possible software systems the system used was the Zigbee network standard. It is based for PAN networks using high level communication protocols. One of its biggest advantages is its long battery life and low power consumption, being cheaper than other technologies like Bluetooth. It is widely used in control and monitor applications, which fits the classification of this product. It is also very fast, able to go from sleep to active mode within 30 microseconds. It is specially designed to function on small microcontrollers such as the HVAC system will use, with easy to develop software.

The software is a modified TCP/IP model which includes the MAC, Network, Application, and Physical layer, while adding the ZDO (Zigbee Device Objects) layer. The ZDO layer is responsible for a number of things such as managing requests for joining a network, security, and device roles, and discovery. Together, with the Application layer from the basis for connecting to devices such as the server that contains the database. The Application Support Layer (APS) is the layer than connects the high level layers (ZDO, Application), to the low level layers (MAC, Network, Physical). Two services provided by the APS is the key-pair service, which configure and modifies object attributes, and the message service, which handles information treatment such that it avoids unnecessary application protocols and potential overhead.

For devices that establish a connection with Zigbee, a profile is created for that device. It contains binding tables that contain identifiers for the source and destination of the devices to send data. The device communicates using unicast if the network address is known; otherwise, it is contacted through broadcast communication. All communication is dependent on the binding tables, which is updated for every new node that attempts to contact a device within the network.

There are two different addressing methods to consider for devices. Direct Addressing uses the endpoint identifier and radio access. Indirect Addressing uses other relevant fields and keeps things simple and small on storage within the network. However it does require a network coordinator to manage associations between the network and device.

One of Zigbee biggest features is its security. The system mainly uses keys for gaining and allowing access for each individual device. The keys are primarily used in the individual network of each device. However, Zigbee does not take into account the existence of firewalls in devices. Since the HVAC system will be using a firewall, Zigbee will need to be aware of its presence. Zigbee also allows networks to share keys, but since separate HVAC systems have little need to interact with each other; this feature was not incorporated into the design.

3.11 Hardware Components

3.11.1 LCD Touch Screen

The requirement for our touch screen display was to be large in size so that it is easy to be operated. We also needed good resolution of the screen and sensitivity to touching. The chosen LCD touch screen panel is developed by Newhaven Display which was to be implemented into our design as the display for the system's user interface. This was one of the most expensive components in our design, so we chose this screen based on the fact that it provides longer lifetime and excellent display resolution for the screen as you can see in Figure 3.11. We had chosen to purchase a 7 inch Thin Film Transistor (TFT) LCD Touch screen with digital output.

The LCD touch screen can be utilized with either the finger of user or a pen to provide input into different stages of our HVAC control menu. On the main menu screen of the LCD screen where the user is able to set the desired temperature and relative humidity for inside the house in cooling mode and heating mode by touching using pen or finger on the buttons provided, checking the current temperature and relative humidity inside the house, view the current temperature and relative humidity outside where the remote sensor unit is located, and choose between several settings that range from maximum savings to maximum comfort which adjusts the system's tolerance for temperature and relative humidity variation between set points.

The best thing for the NHD LCD screen is the durability. According to Figure 3.11 it can be used for 1000000 touches if you use your finger to control it; otherwise, if you use the pen instead, you will be shortening the lifetime of the product to 100000 times. However, this is plenty for our design because it is not like your i-phone which you would use a few million times. The usage of the touch screen is

only when user wants to change the temperature or relative humidity in the house. The biggest consideration for us was the cost rather than the lifetime of the LCD screen.

Item	Min.	Typ.	Max.	Unit
Linearity	-	-	1.5	%
Circuit Resistance – X-Axis	450	800	1300	Ω
Circuit Resistance – Y-Axis	100	350	800	Ω
Insulation Resistance	10	-	-	MΩ
Operating Voltage	-	-	5	V
Chattering	-	-	10	ms
Transmittance	82	-	-	%
Activation Force	50	-	200	g
Pen Writing Durability	100,000	-	-	Characters
Pitting Durability	1,000,000	-	-	Touches
Surface Hardness	3	-	-	H
Haze	-	7	-	%

Figure 3.11: Touch Panel Characteristics for NHD-7.0-800480WF-CTXI#-T

We also were really concerned about the power consuming of the LCD screen, so this one was a pretty good fit to our design because it only use 5V to operate the screen compare to the the XENARC 706TSA which required the operating voltage ranges from dc 10V ~ 24V. The difference of power consumption between these two monitor can be easily calculate using $P=V/R^2$. It can reduce at least 4 times of power consumption. Other good feature for NHD LCD screen is their sensitivite, it only need 50g to let the screen sense the touches. In our design, one of the most important goal for our project was to have a sensitivity screen that allow user to control easily and it could be done with this LCD screen. We also needed to mention the resolution for the product is 800*480 pixels and has built in controller with white LED backlight. The physical specifications for NHD-7.0-800480WF-CTXI#-T is listed below on Figure 3.12.

Item	Specifications	Unit
Screen Size	7.0" Diagonal	inches
Physical Resolution	800xRGBx480 pixels	
Controller	Built-in SSD1963 (8 bits controller)	
Contrast Ratio	300:1	
Response Time	5-10 typically 5	milliseconds
Viewing Angle	Typically 70 for horizon	Degrees
Colors	262,144 colors	

Figure 3.12: Table with Specifications for 7 inch NHD LCD Panel

3.11.2 Environmental Conditions for NHD LCD Monitor

The touch screen monitor we purchased has some environmental conditions that had to be considered when we design our project. Especially the operating temperature that needed to follow; otherwise, you may burn your LCD screen without notice. Also, storage temperature also needed to be considered. It also includes supply voltage. The supply voltage range from 3.0 V to 3.5 V. However it is recommend applying typical voltage 3.3V for best performance. And the operating temperature can stand up to a minimum temperature of -20° Celsius and a maximal temperature of +70° Celsius. The storage temperature can withstand up to a minimum temperature of -30° Celsius and a maximal temperature of +80° Celsius. Even though it is a wide range for temperature but it is better to contain in the typical setting range. Figure 3.13 describes the environmental conditions that must be followed when we dealing with the NHD 7" LCD touch screen.

Item	Symbol	Condition	Min	Max
Operating Temperature	Top	Absolute Max	-20°C	+70°C
Storage Temperature	Tst	Absolute Max	-30°C	+80°C
Supply Voltage	VDD	Absolute Max	3.0V	3.5V

Figure 3.13: Table of Environmental conditions of NHD 7" LCD

3.11.3 Built-in SSD1963 controller

In order for us to control the NHD 7" LCD screen, we needed to use the built in controller SSD1963 which is a display controller of 1215 byte frame buffer which can support up to 864*48*24 bit graphic content. It is perfectly matched with our screen standard that support 800*RGB*480 pixels. It also equips parallel MCU interfaces in different bus width in order to receive command and graphic data from MCU. Its display interface supports common RAM-Less LCD driver of control depth up to 24 bit/per pixel. In order for us to know how to use the controller, we needed to know their features first. Below in Figure 3.14 for the feature of SSD 1963.

Display feature	MCU connectivity
Built-in 1215K bytes frame buffer	8/9/16/18/24-bit MCU interface
Support TFT 18/24-bit generic RGB interface panel	Tearing effect signal
Hardware rotation of 0, 90, 180, 270 degree	I/O Connectivity
Support 8-bit serial RGB interface	4 GPIO pins
Hardware display mirroring	
Programmable brightness, contrast and saturation control	
Dynamic Backlight Control (DBC) via PWM signal	

Figure 3.14: Table of Features of SSD 1963 controller

Some more advanced features of the SSD 1963 controller is they have a built in clock generator. This means the system clock of SSD1963 is generated by the built-in PLL. The reference clock of the PLL can come from two different places- either the external crystal oscillator or the CLK pin. Since the CLK pin and the output if the oscillator was cionected to PLL with an “or” gate as the figure below. The unused clock must be tied to VSS. Before the PLL output is configured as the system clock by the bit 1 of “set_pll” command 0xE0 and then the system will be clocked by the clock from reference. This enables the user to send the “set_pll_mn” command 0xE2 to the PLL for frequency configuration. When the PLL frequency is configured and the PLL was enabled with the bit 0 of “set_pll” command 0xE0, the user should still wait for 100ms delay for the PLL to lock. Then the PLL is ready and can be configured as system clock with the bit 1 of “set_pll” command 0xE0. Figure 3.15 visually shows the Clock Control Diagram below.

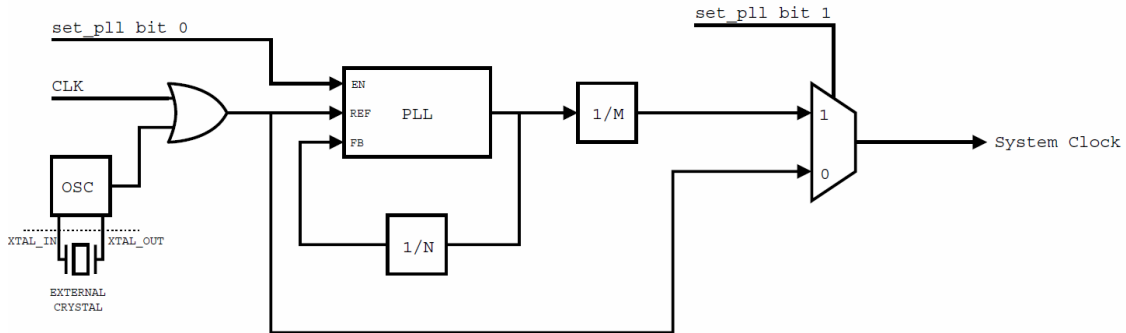


Figure 3.15: Clock Control Diagram (Reprinting Permission Requested from Newhaven Display International Inc.)

Another feature of SSD 1963 controller is that they have Deep sleep mode for power saving which was done by the “System Clock and Reset Manager” distributes the reset signal to the entire system in different states. It controls the clock generator and contains clock gating circuitry to turn on and off the clocks for different module. The system Clock and Reset Manager also manage the reset signals in different stages to ensure all module are reset to appropriate status when reset number is low in any state, like display state, sleep state, or deep sleep state. The Reset state is when the clock generator stop, and unable to receive any command from MCU(Microcontroller Unit) and also unable to updated Frame Buffer. The reset stage will turn off the display and all setting are resetting. In sleep state, the clock generator is on and enable to receive command from MCU and also able to update frame buffer. Of course, the display of LCD screen is off on the sleep state and all settings retain as original before sleeping. The deep sleep mode is when the clock generator stop and the controller can not receive any command from MCU and unable to update frame buffer. The display is off and all settings retain like sleep state. And last but not least is the display state which is the most important state for our design. In this state, all the function is on including display and all settings retain. The figure below show detail of state diagram for the SSD1963 controller. It indicate no matter which state the controller is at, as long as they get the reset number goes low, they will all go into reset state. The state diagram is displayed in Figure 3.16 below.

Figure 7-3: State Diagram of SSD1963

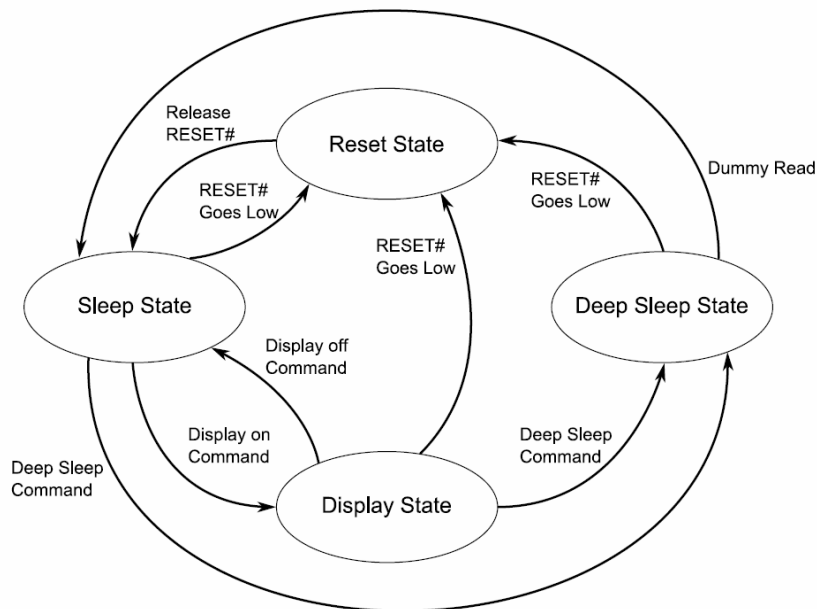


Figure 3.16: SSD1963 controller’s State diagram (Reprinting Permission Requested from Newhaven Display International Inc.)

3.11.4 LCD Touch Screen Comparisons

This research was really essential for us to realize what were the advantages and disadvantages of the elements that was chosen from previous group. In generation one, they choose to use Evervision LCD Panel which is 7" TFT LCD panel and it has to be purchased with a SLCD5 controller as a whole package with the price \$499 from Reach Technology Inc. The reason they choose this one is for the size of the screen and all other features like high resolution and also light weight of the screen. They need to use the external SLCD5 Controller board to load the Bitmap images to the screen onto the screen and place them where the developer instructs it to and can move or remove them as necessary. The LCD touch screen should contain VGA resolution and have at least 16 bit color (2^{16}) which means 262,144 color available for the generation 1 design. The environment condition for the Evervision LCD Panel is exactly the same as our choice NHD LCD screen. The operating temperature can stand up to a minimum temperature of -20° Celsius and a maximal temperature of $+70^{\circ}$ Celsius. The storage temperature can withstand up to a minimum temperature of -30° Celsius and a maximal temperature of $+80^{\circ}$ Celsius.

In generation two, they had chosen to use the XENARC 706TSA LCD screen which is developed by Xenarc Technologies Corporation. It is connected via HDMI and USB (for touch inputs) to the microcontroller. It is a 7 inch Thin Film Transistor (TFT) LCD Touch screen Display with both High Definition Multimedia Interface (HDMI)/Digital Visual Interface (DVI) inputs. The most important factors for them to choose this product over other are the durability. Unlike the 4-wire resistive touch screen which have lifespan of 1,000,000 touches, it was made with 5-wire resistive touch screen displays with is 35,000,000 touches. It can support high resolution from 640 x 480 and up to 1920 x 1080. But its native resolution is 800 x 480. The environment condition for the Xenarc LCD screen is a little better compared with our choice NHD LCD screen and Evervision LCD Panel. The operating temperature can stand up to a minimum temperature of -30° Celsius and a maximal temperature of $+85^{\circ}$ Celsius. The storage temperature can withstand up to a minimum temperature of -30° Celsius and a maximal temperature of $+85^{\circ}$ Celsius. It costs \$349. All other important comparisons is in the Figure 3.17 below.

Parameters	Evervision LCD	XENARC 706TSA	NHD	Units
Size	7	7	7	inches
Native Resolution	800RGBx480	800x 480	800xRGBx480	pixels
Supported Resolution	800RGBx480	640 x 480 ~ 1600 x 1200	800xRGBx480	pixels
Storage Temperature	-30 ~ +80	-30 ~ +85	-30 ~ +80	°C
Operating Temperature Range	-20 ~ +70	-30 ~ +85	-20 ~ +70	°C
Dot Resolution	N/A	2400 x 480	N/A	dots
Response Time	<=10 ms	6	5-10 typically 5	milliseconds
Lifetimes	1,000,000	35,000,000	1,000,000	touches
Surface Hardness	3	N/A	3	H
Display color	262,144	262,144	262,144	colors
Price	\$499	\$349	\$1	dollars

Figure 3.17: Table Comparison for LCD touch screen for three generations

From Figure 3.17 above, we can conclude that the XENARC 706TSA that used in the generation two is the best product on the performance. Even though its native resolution is 800x 480 pixels which is the same as others; however, it can support up to 1600 x 1200 pixels. This is really high resolution even you compare with the monitor with your computer nowadays. The range of storage and operating temperature is wider compare with NHD and Evervision LCD screen 5°C to 10°C on the maximal temperature or the minimal temperature which means it will reduce the failure rate of the product. The most amazing features for XENARC 706TSA is the durability of the screen. It is almost 35,000,000 touches vs 1,000,000 touches compare with other two models. It is really amazing part; however, it is hard to test for that and the price is a much expensive than NHD LCD screen. The other reason why we think the XENARC screen was not good for this project is because in HVAC system, a user will only use it occasionally when there is a need to change the setting of temperature or relative humidity. It is not like your smart phone that you will play around all the times. It was really not necessary to use a most expensive LCD panel because you will never use more than 1,000,000 touches. In this consideration, the price is the most essential consideration for our project. In order to have the potential to mass produce this HVAC system, we needed to reduce the price and increase the performance per dollar spends.

If we compare the Evervision LCD and NHD LCD screen, it is no doubt that we would choose to use the NHD LCD as our choice. Because they both have the almost the same performance but price was reduce dramatically. Besides, NHD has a built-in controller so you don't need to buy the external controller like Evervision LCD. It could further reduce the price. So even if the NHD doesn't have the best performance, it was the most suitable for our design. Figure 3.18 below is the built in controller SSD1963 from NHD LCD panel.

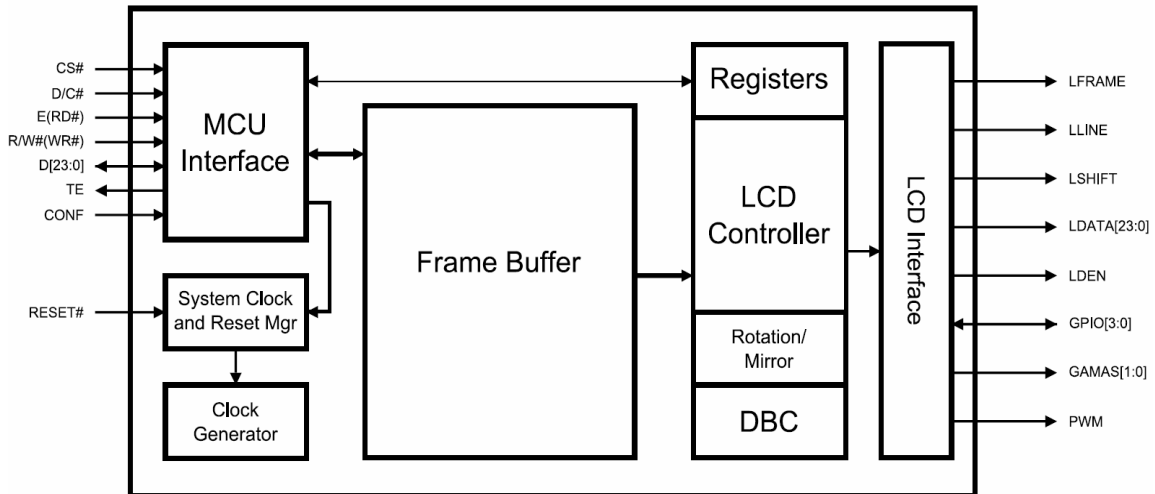


Figure 3.18: Block diagram of SSD1963 (Reprinting Permission Requested from Newhaven Display International Inc.)

The left side input is from the PIC32 Pin. The reset was directly connected to system clock for reset process. And the LCD interface will transfer all the data to the LCD screen.

3.11.5 Panda Board

The PandaBoard is a really powerful board using an OMAP4430 platform for any design. The OMAP4430, shown in the figure below, is a 1GHz Dual-Core ARM processor also has onboard connectors which can supports development of additional capabilities. It has a good advantage because PandaBoard doesn't need to buy an extra external board expansion. Using the PIC32, we needed to have an expansion board to do the same result. It can stream video up to full high definition (HD) 1920*1080 pixels with two HDMI display and support 2D/3D mobile gaming with the help from 1GB DDR2 RAM. As seen below in Figure 3.19, the PandaBoard provides a really powerful board with more connectivity options and support really high resolution. In our design, we needed to add Wi-Fi board because the PandaBoard only provide with Ethernet support. The price is \$279 which we got from the generation two budget section.

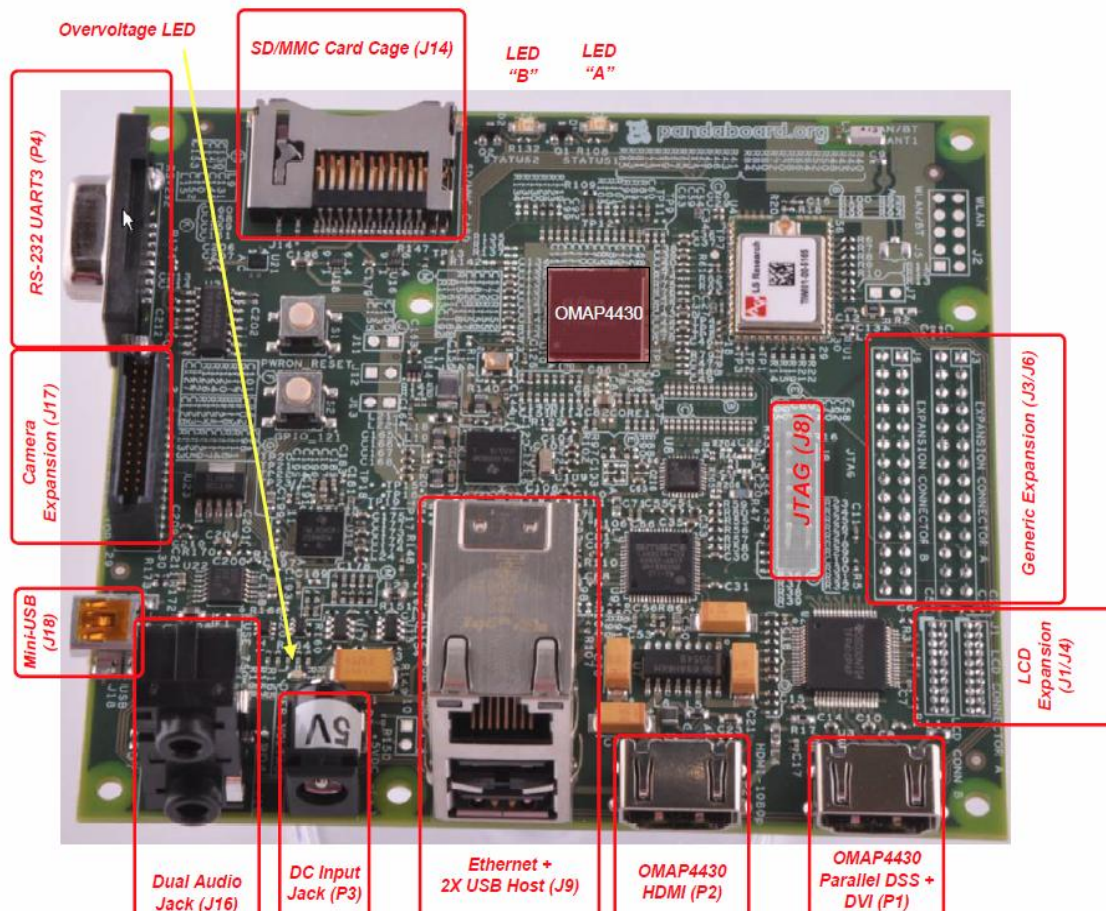


Figure 3.19: OMAP4430 PandaBoard (Permission Requested from pandaboard.org)

The PandaBoard provided two possible choices for us to use the OMAP4430 parallel display signals. The first choice is also the default configuration that routes them to a TFP410 DVI transmitter, whose output feeds an onboard DVI connector. The second choice routes them to a pair of 20-pin LCD expansion connectors (J1 and J4). It is better understood from Figure 3.20 below. In our design, we would have needed to use route two because our NHD touch screen didn't have the DVI connectors. The red route is the default setting that uses the DVI connector to connect external monitor in Figure 3.20 below.

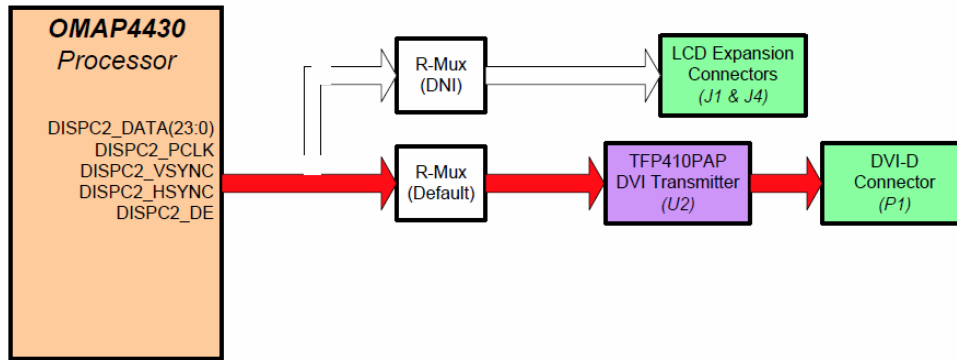


Figure 3.20: Display interface for PandaBoard (Permission Requested from pandaboard.org)

3.11.6 Explorer 16 Kit

Microchips® Explorer 16 Development board provides with a development environment for PIC24 and dsPIC33F microcontroller families which the generation one choose to use. They choose to use dsPIC33FJ256GP710 which was their main controller. It has 16 bit architecture and this 40 MIPS. It is the previous generation of PIC32 and PIC32 is compatible with PIC24 so it is better to use PIC32.

3.11.7 Comparison

Figure 3.21 below depicts a comparison of between the three different processing units; the Panda Board, the PIC32 and the Explorer. There is no doubt that PandaBoard is most powerful choice because it has the fastest CPU which has 1GHz. It is much faster than PIC 32 and PIC 24. It also contains huge RAM compare with other two choices. PandaBoard can support up to 1920*1080 pixels of resolution. It would be much better than the other two and it also has more flexibility among these three choices with onboard expansion pin and with DVI connector for display. However, PandaBoard is far too powerful for our project. First of all, even though it can support to stream video up to full high definition (HD) 1920*1080 pixels, our NHD LCD screen can only support to 800*480. In other words, we would never use the full function of their high resolution power in our design. Second, it has amazing calculation speed per second according to its CPU speed. Our project design does not require high calculation power because we don't have too many complicated algorithms. PandaBoard can support the DVI connection between board and LCD screen; however, it is not supported on the NHD screen. We would waste their full functionality for many parts. And the price is really expensive for so much wasted potential. In order to support its high performance, we needed to use a much better LCD screen which would raise the cost even higher.

Parameters	PIC32*	PandaBoard	Explorer 16	Units
CPU	PIC32 series	OMAP4430	PIC24 series	
CPU Speed	80MHz	1GHz	7.37 MHz	Hz
RAM	128KB	1GB		byte
Onboard expansion pin	No	Yes	No	
Support DVI Connector	No	Yes	No	
HDMI Displays	No	Up to 1920*1080	No	pixels
Price	\$55	\$275	\$299	USD

*means PIC32 development board

Figure 3.21: Table of Comparison between different board selections.

In our design, the PIC32 was the best choice for us, it has the CPU speed we need and it has just enough support for our LCD screen and was easy to implement the design for the touch screen.

3.12 Security

A request by Natural Air was to ensure the software we create be secure. The topic of security on the Internet is a mystery to many people. In order to satisfy the needs of the client, this subject required further study ahead of time. This project consists of network development, and will be composed of a database and clients, both internet browsers and the simulated hardware. All of these components needed to be connected securely. For example, future editions of our project may include sensitive information on customers such as credit card data. It is imperative that those elements do not end up in the hands of malicious “black hat” hackers. Even more petty attacks such as DoS (Denial of Service) could interfere with customers’ ability to use our client’s services.

Security risks can be broken up into two types: threats and attacks. Threats are “potential” security violations. Attacks are attempts to breach existing security measures. When one thinks of the terms “threat” and “attack,” one typically thinks of a malicious intruder trying to breach security and steal information. The truth is, attacks and threats can be performed accidentally by users just as much as maliciously. Threats come in four different types. Interruption is a threat in which an asset of a system becomes temporarily or permanently unusable. It could include the destruction of hardware or severing of a communication line. An example would be losing our database, rendering our phone and web applications useless. Interception is a more malicious threat, which involves an unauthorized party gaining access to confidential assets. Wiretapping is a classic

example people see in movies. User information being stolen from our database would be an example of interception. Modification takes interception a step further by making changes to confidential assets, such as modifying information in the database. A particularly tricky threat is fabrication, in which an unauthorized party creates counterfeit objects and inserts them into a system and making them appear authentic. An example would be an attacker adding their own information into a database so that they appear as a legitimate user, in the attempt to steal services or data.

Attacks are more malicious than threats, and there are five different kinds. A “breach of confidentiality” is the unauthorized reading of data or theft of information. The most common classic example is the stealing of credit card numbers off a database. For the sake of our project, this will not be an immediate concern, but when Natural Air integrates their business models into our work when the project is complete, care needs to be taken when customer information is put on internet databases. A different kind of breach is a “breach of integrity,” in which data is modified by an unauthorized party. This can result in users modifying the internal source code of a system. The last breach is the most serious “breach of availability,” which involves destroying data. An example would be a malicious hacker trying to bring down a website’s functionality for “bragging rights.” Theft of Service is the fourth attack, and is the unauthorized use of service for personal gain. An example would be a thief going into a web based system and installing software to make their own services on someone else’s server, stealing broadband. In fact, this happens every time someone sets up an unsecured wireless network in their own home and someone else uses it to go on the internet. This is common in urban residential areas such as tight apartment complexes. The final kind of attack is the popular term “Denial of Service” (DoS) attack, in which use of the system is prevented. The good news is that it’s easier to guard against accidental attacks and threats than it is malicious ones. In fact, guarding against accidental breaches in security is the first step on the path to making a system secure against malicious breaches.

This project requires the use of many HTML based forms with PHP and SQL code embedded in them. The advantage to PHP and SQL is the fact that they are server based languages. A user will not be able to view them by going into a browser and checking the source code (see section 3.7.2 for an example). It doesn’t change the fact that certain precautions must be used in coding. For example, one common source of vulnerability in web applications is Cross-Site Scripting (called CSS but different from Cascading Style Sheets.) Cross-Site Scripting is any code that interacts with a database containing hidden information and prints it to the user. In the language of PHP, that includes print, printf, and echo commands. An innocent example would be a user retrieving their account information on a shopping website, and having their database recorded Name, Address, and Phone Number displayed to them. But what about checking billing information? Under no circumstance should a user’s credit card be in full display, even in the comfort of his or her home computer. Otherwise someone else could

see it fully and steal it. In assuming this happens in the comfort of your own home, someone else could look at your browser history and see your credit card number in full display. A simple and popular solution to this is to write the display of the credit card by blocking out the first twelve credit card digits with asterisks and revealing only the final four digits.

A not so obvious security risk is buried within SQL errors. A database is the core of our project from which all components communicate, and it must also be secure. The vulnerabilities of a database come in two different forms. The first is in the connection setup between clients and the database. Connections to databases are initialized with sensitive information such as a username and password on the user's side, and the database server and table on the server side. If there is a failed connection to a database, without any precaution, a user could accidentally see sensitive details in the form of an SQL error message, including the database host address, authentications and target tables. This means that the way our applications connect to the database need to be inspected. This includes the web browser connections, the smart phone application connections, and the hardware simulation's connections.

The second major risk in using SQL is in queries to the database. A successful database connection, but with a failed query, could expose the database table layout, revealing user data. Corrupted data could also enter the database by means of such PHP functions as `mysql_connect()`. These problems are not exclusive to PHP, because our simulated hardware will be coded in C. C does not have any "standard" libraries for accessing databases, but the project will most likely use the `mysql.h` library, so as functions are found and used for accessing the database, they must be inspected for how they communicate to the database. For example, all of the C interactions with strings such as `strcpy()` need to be inspected for potential risks. The most common exploitation of SQL is called SQL Injection, in which hackers try to create their own POST and GET commands in order to insert malicious information into a database, or retrieve sensitive information.

As the later parts of the project may eventually use AJAX, the use of XML will require some extra study. The XML will be used to create templates and style sheets for the various pages of the web page. XML creates developer specified tags to HTML, and at first glance does not seem important. However, it is so versatile, that it is modernly used to interact with databases. It is also completely open sourced and easily used by any developer. A disadvantage to XML is that in its raw state, users can view it. Instead, XSL is a document that can transform XML into HTML, disguising it so that users cannot see it. AJAX uses JavaScript, and a classic feature of JavaScript is that a user can deactivate it on his or her own browser. Therefore, under no circumstance must security measures hinge on JavaScript.

One final case of a not-so-obvious security risk to the system would be hardware theft. The simpler case would be an account holder losing their phone, in which case a malicious user could tamper with their HVAC system. Some possible solutions would be for Natural Air to have quality customer support for the event a phone is lost, and the ability of a user to deactivate all smart phone access from their account via either the web HVAC controller, or the hardware touch screen system itself. An extreme case of hardware theft would be the touch screen unit being stolen, from either a homeowner or Natural Air's hardware distribution. This can be prevented by Natural Air keeping accountability of all unused hardware units it has not sold, and by having responsive customer service for its customers in the event of an emergency. All of these security risks can be summarized in Figure 3.22 below.

Project Component	Risks	Resolution
Phone Application	Phone Theft, Data Theft	Customer Service, Censored password, error inspection
Simulated and Future HVAC Hardware	Hardware Theft	Customer Service, Hardware Accountability
MySQL Database	Account Theft	SQL Error modification, Accountability
Web HVAC Controller	PHP, XML, SQL errors, Cross-Site-Scripting	Error inspection, Censored password, Customer service

Figure 3.22: Security Risk Assessment

Section 4: Project Design

4.0 Hardware and Software Design

This section will review the design of the project's hardware and software sections. Since most of the hardware will not be designed by this team, the hardware section will be less extensive than the software. While most of this design specifications will be mostly related to the software section of the project, there was still an important role the hardware would play.

4.1 Hardware Design

4.1.1 Hardware Redesign Purpose

One of the main differences between this current project and the previous generations is the making of a mainstream hardware design that is to be mass-produced. It may be possible that this system is the final product that consumers will begin purchasing. As such, it is extremely important that the network database and the HVAC system can communicate without any issues. What is also important is that the final layout of the revised system is displayed perfectly in the network, as well as having all the correct data. One of the most important aspects of the project was to test and ensure that the finalized system can connect and accurately send all the data onto an online database. This means that there must be a test field for the software to work on. A prototype hardware device was needed to simulate the final system and send data to the server. If the actual product is available during or before the testing phase, it can also be tested on to ensure accuracy and consistency.

The hardware used to design the HVAC Feedback and Control System 2.0 has originally been given to the subcontractor Tim VanderMey of Proton Design Inc. hired by our sponsor Natural Air Energy Savings System. The hardware used by the previous senior design group was also a working hardware simulator and the HVAC system was fully functional. However our sponsors from Natural Air wanted to take the technology from the HVAC System 2.0, which they also sponsored as well and have it ready for mass production. Tim VanderMey was given the responsibility making this technology ready for mass production, in the brief time that he has had this responsibility he has made significant changes to the hardware that are attainable to the goals from our sponsors. One of the main improvements that he has made in helping make this HVAC system mass producible is reducing the costs of parts, in addition to using better technology. Eventually however, the redesign of the hardware was postponed, which meant the only testing field we could use was the simulator we would design.

Even though the new hardware designs for this HVAC system is still under Proton Design Incorporated, our subcontractor Tim was able to provide us with the hardware that he plans to use for the redesign. Our main objective from our sponsor was to make this HVAC system accessible through the internet and phone applications and in order to make sure we can successfully achieve our goal we will need to make a simulation of the HVAC system using the same hardware that our subcontractor is going to use in his redesign. In order to show that the software we designed for our web browser and phone applications is compatible, we tested it with the hardware we use.

These are the new hardware items that we will use for our simulation:

- PIC 32 Microprocessor
- I/O Board
- WiFi Extension
- Ethernet Daughter Board
- LCD Touch Screen

4.1.2 I/O Expansion Board

The PIC 32 I/O Expansion Board is to be used in conjunction with our PIC 32 USB II Starter Kit with the sole intended purpose of serving as the mother board for the starter kit. This expansion board will provide a simple method of adding input and output expansions of the PIC 32 USB II Starter Kit and provide relevant connectors to power and debug the starter kit. The PIC32 I/O Expansion Board will serve the role as our motherboard for our hardware simulator of the HVAC system, providing the connection between our PIC 32 USB II Starter Kit, LCD Screen, Ethernet PICtail Plus Daughter Board and Kit Dev Zerog 802.11 Wi-Fi board to give our main controller unit the ability to connect to our web server for internet and mobile phone application access.

The programming that we established on the PIC32 USB II Starter Kit will relay functions and data from the LCD touch screen to the Ethernet Daughter Board and Wi-Fi board in order to send this information to our web server. Once all information has been relayed to the web server it will be capable of being accessed by a web browser and/or a phone application. Our web server was able to receive any necessary information or changes that were made from a web browser or mobile phone application and send it back to our Ethernet Daughter Board and Wi-Fi board and the PIC32 I/O Expansion Board will let our PIC32 microcontroller aware of changes that have been made over the web server and our microcontroller will relay these changes back to the LCD touch screen.

Key features for this PIC 32 I/O Expansion Board are:

- 132-pin Starter Kit connector
- RJ11 2-wire Debug Interface
- JTAG and Trace Headers
- Optional 9V DC inlet with 3.3 and 5 volt regulators
- Two 2x30-pin header footprints for optional MCU signal headers
- Two 120-pin PICtail Plus daughter card sockets
- 120-pin PICtail Plus edge connector

4.1.3 Wi-Fi Extension

In order to provide our hardware the ability to be able to connect to a web browser and phone applications we must have provided our hardware with the capability of being Internet accessible. To give our hardware this capability we used the Kit Dev Zerog 802.11 Wi-Fi board. This board was able to demonstrate Wi-Fi connectivity of 802.11 using PIC microcontrollers. Using TCP/IP Stack we configured this Wi-Fi extension board to be capable of Wireless Router Configuration in addition to Media Access Control.

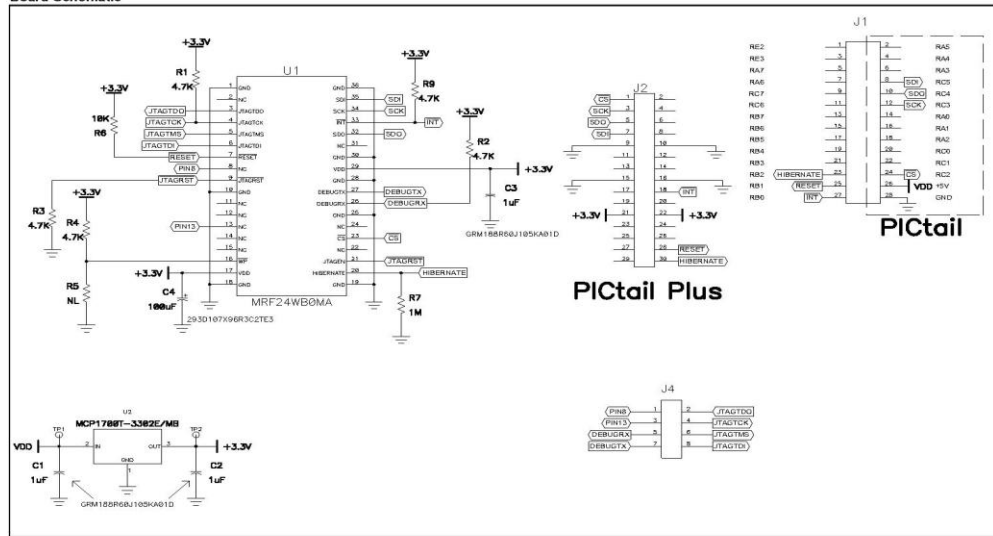
We established connection of the Kit Dev Zerog 802.11 Wi-Fi board to our PIC 32 I/O Expansion Board. Microchip the creator of this Wi-Fi board offers a free licensed TCP/IP stack to use in order to get the Wi-Fi board started. This stack uses multiple layers, where each layer accesses services from one or more layers below it. Many of these layers found on Microchip's TCP/IP stack are live in the sense that they do not only act when a service is requested but also when events like time-out or new packet arrival occurs, this is beneficial to what our group wants to accomplish with our web server in providing automatic updates to the firmware for the HVAC system in addition to supplying weather updates for the are the HVAC system is located in. Thus giving the HVAC system the capability to adjust its settings based on predictions that the HVAC system has received on what the outdoor temperature should be if the user chooses to have their system set up that way. Microchip's TCP/IP stack is modular in design and was written in C programming language, and implications to the TCP/IP stack to be in relation to what we want to accomplish with our software would only take up between 28-34 KB of code, resulting in plenty of space to be available in writing the code to program our PIC 32 Microcontroller. Below in Figure 4.1, the PICtail Plus Daughter Board's schematics are displayed.

Key features for Microchip's TCP/IP stack:

- Optimized for PIC 32 Family
- Supported Protocols: ARP, IP, ICMP, UDP, TCP, DHCP, SNMP, HTTP, FTP, TFTP
- Socket support for TCP and UDP
- Secure Socket Layers (SSL)
- NetBIOS Name Service
- Domain Name System (DNS)
- Ethernet Device Discovery
- Support for MPLAB C18, C30, and C32 compilers

MRF24WB0MA PICtail™/PICtail Plus Daughter Board

Board Schematic



DS1912A

Figure 4.1: Wi-Fi Extension Board

4.1.4 Ethernet Daughter Board

The Fast 100 Mbps Ethernet PICtail Plus Daughter Board is an Ethernet demonstration board for evaluating Microchip Technology's stand-alone 10/100 Ethernet controllers. It is an expansion board compatible with many PICtail and PICtail Plus host boards, including the PIC32 I/O Expansion Board which we plan to use this Ethernet Daughter Board with. Its semantics are shown in Figure 4.2 below.

Key Features of Ethernet PICtail Plus Daughter Board:

- ENC624J600 10/100 Ethernet Controller with factory preprogrammed MAC address
- 25 MHz oscillator and integrated magnetic RJ-45 connector
- PICtail Plus Daughter Board edge contacts for SPI or PSP connection to PIC 32 I/O Expansion Board
- PICtail Daughter Board headers for SPI connection for SPI connection to PICDEM.net and PIC18 Explorer Boards
- Configuration jumpers for PSP modes 1,2,3,4,5,6,9,and 10 under PMP control with automatic pinout remapping
- Current monitoring jumper
- Ethernet crossover switches
- 3.3V LDO regulator with shutdown for very low Sleep current

- 3.3V to 5V SPI level translation for the PIC 18 Explorer
- SSL encryption add-on software CD for the Microchip TCP/IP Stack

Ethernet PICtail™ Plus Daughter Board

Board Schematic

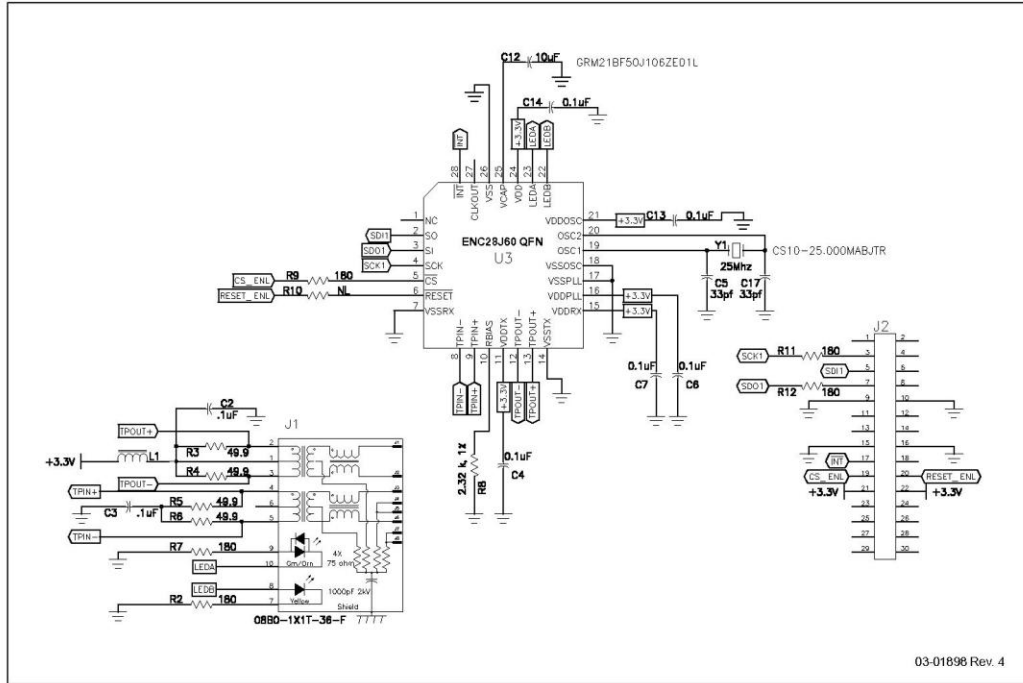


Figure 4.2: Ethernet PICtail Plus Daughter Board

In order to establish a connection of the Ethernet PICtail Plus Daughter Board to our hardware it must be plugged into our PIC32 I/O Expansion Board along with a suitable PIC32 USB Starter Kit, which we had the second version of. Our Ethernet Daughter Board was connected to the PIC32 I/O Expansion Board by inserting the Ethernet PICtail Plus J2 side edge contacts into the topmost position of the J4 or J2 slots on the PIC32 I/O Expansion Board. Pin 1 of J2 on the Ethernet PICtail Daughter Board should align with the Pin 1 slot on the PIC32 I/O Expansion Board. We tested to see if the connection between the Daughter Board and our Expansion Board is correct, by verifying that the green Ethernet link status LED automatically turns on when a successful connection between the Ethernet cable to our Daughter board and a link partner. We were able to use the TCP/IP stack software running on our PIC32 I/O Expansion Board in programming the Ethernet PICtail Plus Daughter Board to run the responsibilities efficient to our hardware simulator for the HVAC system.

4.2 PIC32 Microprocessor

The development kit we chose for this project is DM320003-2-ND which is the PIC32 USB II STARTER with the 32-bit MIPS M4K core that has 80MHz which can handle up to 1.56 Dhrystone million instructions per second (DMIPS.) More details is in the figure below. It also contains 512K flash with pre-fetch cache and 128KB Ram for data and program execution. It has USB 2.0 connection ability and also with the 10/100 Ethernet MAC with MII/RMII interfaces. They use the 5 stage pipelines to design the microcontroller unit with instruction fetch, instruction decode, ALU, data memory, and write back stages. It is a code-efficient architecture using C and Assembly language. It has independent watchdog timer and fast wake up and start up time. It also can enter into the low-power management mode (sleep, idle). Below is a close up of the PIC32 Microprocessor, as well as some properties of the device, reprinted with permission from Microchip in Figure 4.3 below.

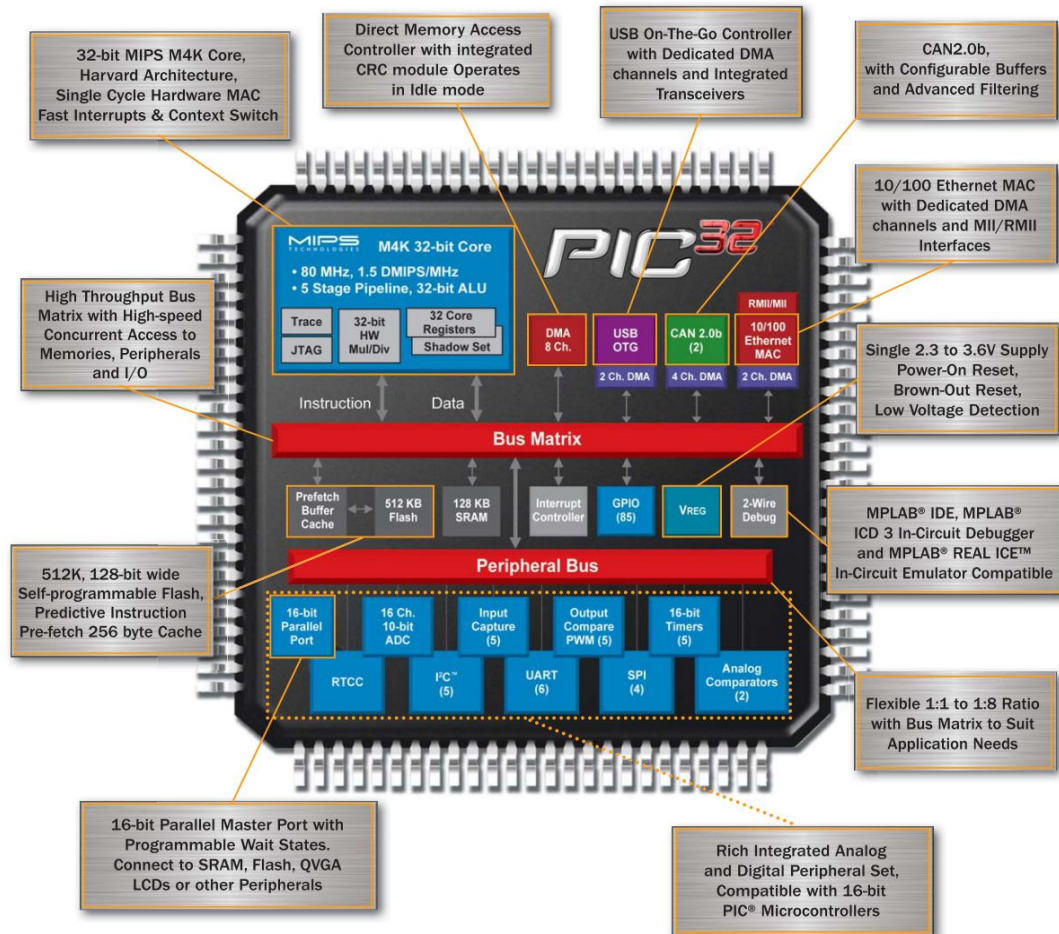


Figure 4.3: PIC32 detail information (Reprinted with Permission from Microchip Technologies)

The source code for USB, TCP/IP, graphics and file system are free and used common MPLAB development tools of the integrated circuit. The best part for the PIC 32 controller is they support controller-less graphics, which means they don't need the external controller but they are also compatible with external graphics controllers. They support with many different display resolutions and sizes in Table below but we needed to pay attention that we only list some important categories. Our 7" NHD touch screen has display resolution 800*480 which is supported by PIC32 on the first row of table. The next figure below, Figure 4.4, shows the variety of available screen resolutions that the PIC32 can support, as well as what the actual screen size will result from making adjustments in resolution, image wise.

Kind	Display resolution	Typical Size
WVGA	800*480	7"
VGA	640*480	5.7"
WQVGA	480*272	4.3"

Figure 4.4: Table of Support resolution and size for the PIC32

We need to mention why we choose PIC32 over PIC24 DA. In our design, we considered needing better resolution. As the display resolution and color depth increase, the frame buffer grows at the same time. If we have bigger frame buffer that can store the information for the screen to display directly into microcontroller RAM, we don't need to save into external SRAM or integrated into an external graphics controller. The PIC24 DA has 96KB frame buffer with cores 16 MIPS. On the other hands, PIC32 has 12KB frame buffer with core speed 80 MIPS. PIC32 can handle 5 times more instructions per second compare with PIC24. The biggest different between these two is that PIC24 with integrated graphics controller; however, PIC32 can drive a display without an external graphics controller which means it can direct interface to TFT displays. Both of them can also work with an external graphics controller to support or more advanced graphical features or larger screen size. It added more flexibility into our design. View Figure 4.5 below to read about the differences.

	PIC24 DA Integrated Graphics Controller	PIC32 Controllerless Graphics	External Solomon Systech Graphics Controller SSD1926	External Epson Graphics Controller S1D13517
Display*	WQVGA 480x272	WQVGA 480x272	WQVGA 480x272	WVGA 800x480
Graphics	HW Acceleration: Rectangles, Characters, Images	DMA on PIC32 + <5 MIPS	HW Acceleration, SD card, I/F, JPEG engine	SDRAM, I/F, Alpha-blending, Picture-in-picture
Frame Buffer	Color Lookup Table + 96 KB on MCU + Ext SRAM	128 KB on MCU + Ext SRAM	256 KB on Solomon Systech Controller	Ext SDRAM
Core MIPS	16	80	-	-
Power	Better	Good	Good	Good
Cost	\$	\$	\$\$	\$\$\$

Figure 4.5: Comparison between PIC24 and PIC32 ((Reprinting Permission Requested from Microchip Technology)

The other good features for choosing PIC32 is that Microchip technology Inc. provides many open libraries for us to use. It contains pre-made graphics objects, user interface for mTouch™ Sensing and also includes buttons charts, checkboxes, scroll bars and so on. Using PIC32, we could easily use the free software Microchip Graphics Display Designer (GDD) to design our user interface of our touchscreen display. GDD is a visual design tool that provides easy and quick way of creating Graphical User interface (GUI) screens for our applications. It gave many advantages for our design. First, it simplified the coding for our GUI design. All we needed to do is draw, resize, and delete screen objects. It will significantly eliminate the need to manually calculate the X/Y coordinates for the object placement. The software had the ability to import various graphical resources, like bitmaps images. It was very beneficial if we could import the bitmaps images directly and it will generate output source files that we could use directly.

4.3 Remote Unit

To enable our hardware simulator to function like an efficient HVAC system, we need to have a remote temperature and humidity sensor responsible for measuring the outdoor temperature and humidity. We had a secondary remote unit with a PIC24 microprocessor responsible for reading the outdoor temperature and humidity and transferring that data to our main unit. Due to this unit being remote and more than likely being placed away from a power source all the components on this unit need to be very low powered and was able to operate using a batter power source. This unit will be a PCB consisting of a PIC24F16KA101 microprocessor, SHT21 Temperature humidity sensor, and Linx RF modules LR series.

We chose the PIC24 microprocessor for programming the remote unit of the simulator because the previous groups botch chose this route for picking a microprocessor to control their unit and had success with this route. The

microcontroller is low powered so it can receive its supply voltage from a battery source and it carries the function of going into a deep sleep mode when it is not in use, consuming very little power. In addition to these attributes this PIC24 microprocessor also has a temperature range from -40 to 125 degrees Celsius allowing this microprocessor to easily withstand the conditions that it would have to endure in order to serve its purpose. Figure 4.6 below is a diagram of the PIC24's Pin Outputs.

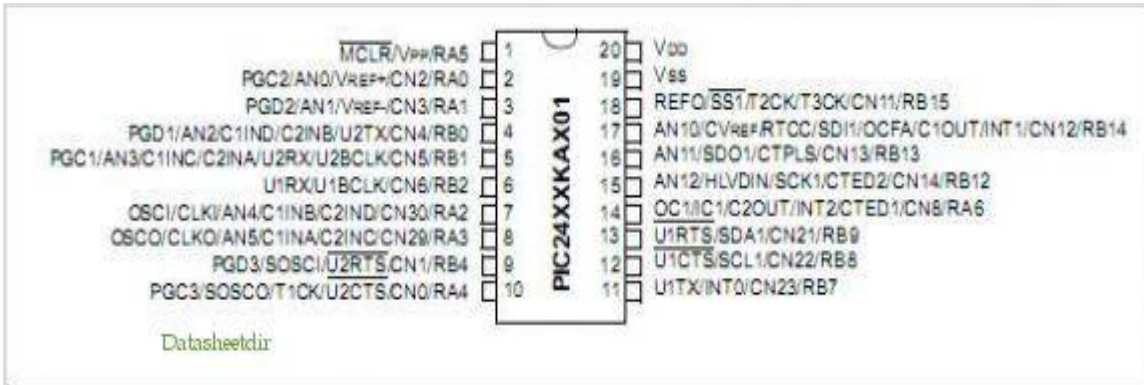


Figure 4.6: PIC24 Pin Out Diagram

Following the reasons behind the selection of the microprocessor, the exact same thoughts were put into effect when selecting a temperature and humidity sensor. The SHT21 sensor was the selection made based off the success that the previous groups had with using this sensor and programming it with the microprocessor selected above. The SHT21 sensor also displayed the qualities that we needed for our remote sensor by being small in size, requiring a low supply voltage, and having very low energy consumption. The SHT21 sensor is made for outdoor conditions with a temperature range from -40 to 125 degrees Celsius, and being able to provide an accurate humidity range ranging from 0 to 100%.

The method of transferring the data from the remote unit to the main unit was made by using two Linx RF modules LR series. This method is a completely different method than the previous groups used to transfer the data from the remote unit to the main unit, their method was with a Xbee wireless RF modules. This method would have been impossible to have done based on the fact that our main unit is using a Zerog 802.11 Wi-Fi board which would of not been able to communicate with the Xbee wireless RF modules since they use 802.15.4 networking protocol which does not communicate with 802.11. The Linx RF modules LR series can provide the functions needed for our remote unit being that it can communicate with another RF module LR series when paired with it for up to a distance of 3000 feet and can transfer data at 10000 bps, which was more than suitable for the functions that our remote unit requires and did not require the making of a wireless connection between our remote unit and our main unit. So the Wi-Fi component on our main board could be used for the sole

purpose of establishing communication between the hardware and the database. The Linx RF module LR series is also low in power consumption and carries a wide temperature range meeting the specifications needed for the components on the remote sensor. Figures 4.7 and 4.8 below provide a closeup of the RF modules.

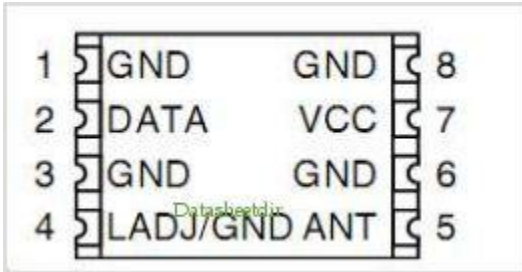


Figure 4.7: TXM315

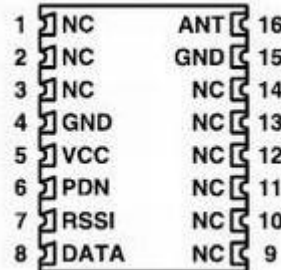


Figure 4.8: RXM315

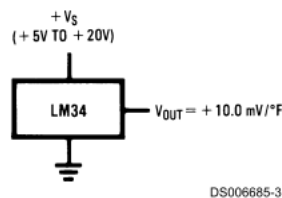
4.4 Hardware Components

Late into the semester, a wide variety of hardware was added to the project as well. The following hardware additions were based on the direction Tim VanderMey of Proton Design Inc. is taking with the redesign of the hardware. At the core of the needs was the PIC32 Microprocessor (We chose PIC32 starter kit which is \$55.00), which serves as a significantly cheaper alternative to the previous processor, the Panda Board, which was \$275.00 that contained too powerful features we would never use.

The PandaBoard is a really powerful board using an OMAP4430 platform for any design. The OMAP4430 is a 1GHz Dual-Core ARM processor also has onboard connectors which can supports development of additional capabilities. It has a good advantage because PandaBoard don't need to buy an extra external board expansion. Using PIC32, we will need to have an expansion board to do the same result. It can stream video up to full high definition (HD) 1920*1080 pixels with two HDMI display and support 2D/3D mobile gaming with the help from 1GB DDR2 RAM. However, all these functions are too powerful for us so we decide to go with PIC32 as our development platform. PIC32 contains 32-bit MIPS M4K core that has 80MHz which can handle up to 1.56 Dhrystone million instructions per second (DMIPS). It also contains 512K flash with pre-fetch cache and 128KB Ram for data and program execution. It has 16 analog pins that we can use for our project because Chung chose to use some analog device for our indoor temperature and humidity sensors and also the CO2 sensor.

4.4.1 Temperature Sensor

In order to detect the temperature indoor and outdoor, we chose the LM34 (Precision Fahrenheit Temperature Sensors). The LM34 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Fahrenheit temperature. The LM34 thus has an advantage over linear temperature sensor calibrated in degrees Kelvin, because user did not require subtracting a large constant voltage from its output to obtain the convenient Fahrenheit scaling. Below in Figure 4.9 is the basic usage of the LM34.



**Figure 4.9: Basic Fahrenheit Temperature Sensor (+5° to +300°F)
(Permission Requested from TI)**

For demonstration purpose, +5 °F to 300 °F was good enough for our design, especially in Florida. And for voltage, we give them 5V in our design. The other advantage for LM34 is that it does not need any external calibration to provide typical accuracy of $\pm 1/2$ °F at room temperature and $\pm 1 1/2$ °F over a full -50 to +300°F temperature range. The LM34's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It has very low self-heating, less than 0.2 Fahrenheit in still air and is rated to operate over a -50° to +300°F temperature range. Below in Figure 4.10 is the bottom view of the LM34 temperature sensor.

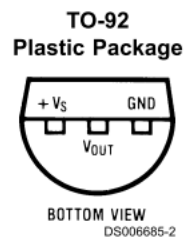


Figure 4.10: LM34 temperature sensor (Bottom view) (Permission Requested from TI)

Many electrical signals around us are analog in nature. This means a quantity varies directly with some other quantity and the first quantity is mostly voltage while that second quantity can be anything like temperature, pressure, light...etc. In our design, LM34 temperature sensor would output voltage varies according to the temperature it measured, so if we could measure the voltage, we should be able to measure temperature also.

But most of our microcontrollers are digital in nature, not in analog signal that we want to use like voltage. It means they can only differentiate between High and Low level on input pins. For instance, if input is more than 2.5V it will be read as 1 (High) and if it is below 2.5 then it will be read as 0 (in the case of 5V system). So we cannot measure voltage in that way directly. In order to solve the problem, we used a ADC unit (Analog to digital Converter). It would convert a voltage to a number so that can be processed easily by digital system like MCU.

The most important specification of ADCs is the resolution. This specifies how accurately the ADC measures the analog input signals. In the PIC32 we used, it has 10 bits but normally ADCs can be 8 bit, 10 bit and 12 bit. In order to calculate the right factor for our input to get the right temperature output. We will demonstrate the process step by step below.

For example, if the reference voltage of ADC is 0 to 3.3v (we use the PIC32 as example) then a 10 bits ADC resolution will break it in 1024 divisions so it can measure it accuracy up to $3.3/1024=3.2$ mV so you can see that the 10 bit ADC can't distinguish difference between 1mV to 3.2 mV but it is good enough for our design because 1 °F is 10mV difference. The value we read from ADC on PIC32 is what we need to do "real voltage = ((ADC reading)/1024.0 *3.3*1000); //Voltage in mV" since 10mV = 1 degree from Figure 4.9, to get temperature we must divide it by 10 from previous equation. "t= ((ADC reading)/1024.0)*3.3*100); //It is in degree Fahrenheit. " After simplifying, it becomes "t= (ADC reading*0.33)."

4.4.2 Humidity Sensor

In order to detect the humidity indoor and outdoor, we chose the CHS-MSS from TDK Corporation. It has a compact size so we can reduce the size of the total sensor and it has low current consumption. If the output DC is 1 V, it means at 100% RH. Voltage and the humidity have linear relationship so the relative humidity can be read directly with a voltmeter. Figure 4.11 shows details on the humidity sensor.

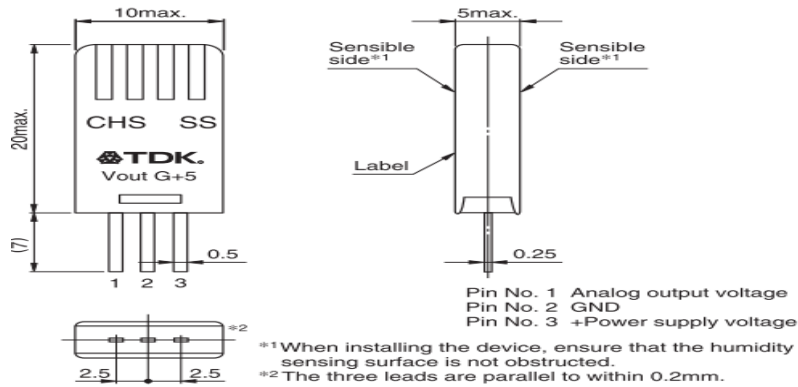


Figure 4.11: CHS-MSS humidity sensor (Permission Requested from TDK Corporation)

In order for us to calculate the right factor for our humidity input, we need to take closer look on Figure 4.12. This figure displays the mostly linear relationship between the output voltage and the relative humidity.

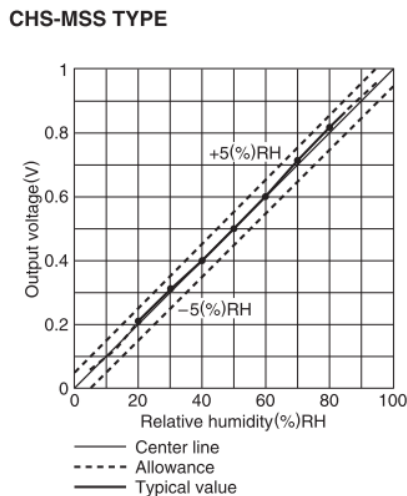


Figure 4.12: Sensor Linearity (Ta=25°C, Edc=5V) (Permission Requested from TDK Corporation)

4.4.3 CO2 Sensor

This part was added by sponsor recently because it can increase the potential customers who would interest for this particular function. This is because it has many potential uses including saving lives. In order to detect whether the CO2 concentration was over normal range or not, we chose to use the CO2 gas

sensor module 27929 from Parallax Incorporation. It is interfaced with through 4-pin parallel header and only requires 2 I/O pins from MCU which can be easily implemented in our PIC32. The sensor was used to provide a means of comparing gas sources with normal CO2 concentration which was set when one did the calibration and being able to send an alarm signal when the CO2 source became excessive. The sensor is displayed in Figure 4.13.

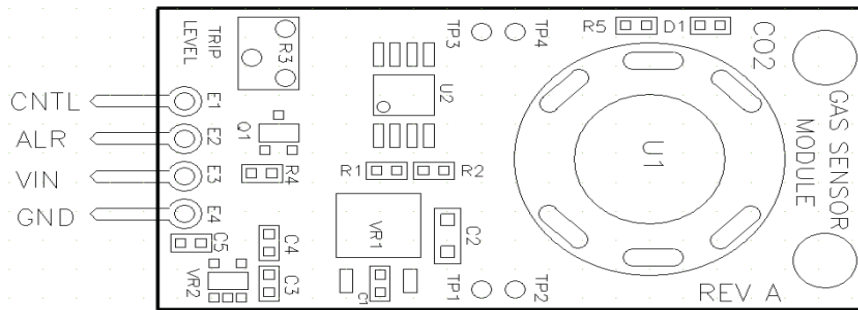


Figure 4.13: CO2 Sensor schematics (Permission Requested from Parallax Incorporation)

In Figure 4.13, there are 4 pins for this CO2 sensor-CNTL, ALM, VIN, and GND. More details are displayed in Figure 4.14. In our design, we turned on the CO2 sensor all the time for demonstration purpose, so the CNTL needed to be set high at all times to turn on the heater do detect the CO2 concentration. So we connected the CNTL and VIN to a 9V DC voltage in our design. And we connected the ALM pin to the analog pin on PIC32 so when the alarm was triggered it will send the active high signal to the microprocessor.

Pin	Name	Function	Level
E1	CNTL	Heat switch input from MCU, active High	0-12V DC
E2	ALM	Alarm output to host MCU, active High	0V/3.3V
E3	VIN	Input Voltage	+6.5V-12V
E4	GND	Ground	0V

Figure 4.14: Pin description for 2792 CO2 sensor

4.4.4 Testing the Indoor and Outdoor Sensor

It was really important for us to test our hardware especially indoor sensor like humidity, CO₂, and temperature sensor. In order for us to test the temperature sensor, we first needed to decide how to test it. First, we put something hotter than the indoor temperature on the top of LM34 sensor so we could see the temperature increase on the computer screen. In order to let the result print on the computer screen for simulation purpose, we need to include the `<stdlib.h>` and `"#define PIC32_STARTER_KIT"` into our codes; otherwise, we couldn't print any results on the screen. We used our breath to increase the temperature by continuously expiring hot air on the top of temperature sensor so you could see the changes of the temperature easily on the screen. On the other hand, we did some testing on decreasing the temperature by putting the cold drinks close to the surface of the temperature sensor. But as we were doing it, we should be extremely careful not to let the water drop on the circuits which may cause some problems.

Other than testing temperature sensor, we tested the indoor temperature sensor as well. But it was hard to control the humidity inside the building to test it. So we will just use our breath to increase the humidity temporarily by expiring the humid air from mouth to the sensor. And you should be able to see the changing on the screen. When we did the testing, the indoor humid sensor should increase a lot. Normally, indoor humidity is roughly 30% to 46% RH so when you breath the

humid air on it, it should increase to roughly 75% RH (This data is the one we did the testing before in the senior design lab). We didn't do much testing for decreasing the humidity because we do not have access to the AC on the school. It was really hard for us to do the testing for that particular part but if the testing for increasing humidity is working, it should be work as well.

CO₂ sensor testing was a little harder compared with the temperature and humidity sensor. Because before you could do any of testing, we must first needed to do the calibration for the CO₂ sensor to set the normal concentration of CO₂. First, we needed to place the sensor in a clean air environment and supply power (9V) to the activate sensor for at least 5 minutes before making any adjustments. Second, we needed to use voltmeter to measure the voltage between TP1 (+) and TP2 (-) in Figure 4.12. It should be between 1.5V to 3.3V. Then we adjusted the potentiometer R3 until the voltage across TP3 (+) and TP4 (-) read approximately just below the reading from TP1 and TP2. The LED should go out now. Third, applying the gas source to the sensor, the light should be lit up

by now to show the excessive of CO2 in the surrounding. Then we removed the CO2 source and the light should have gone off within 60 seconds. If not, we needed to redo all the calibration again.

After the calibration, now we could do the testing for CO2 sensor. First, we were a little worried on how we could get the CO2 gas, but we found out it was pretty easy. We used cleaning duster which they used CO2 to clean the electronic device and we used that as our CO2 source. Before we supplied the CO2 to the sensor, the light on the module should be off which indicated the CO2 level was normal and the computer screen should show the CO2 concentration in “Normal” indication. After the CO2 gas was added close to the sensor, the light on the sensor should be on and the screen would show “Warning! Warning! Warning! The CO2 concentration is too high” on the screen to indicate the excessive CO2 exist in the environment. And after around 1 to 2 minutes the light should be off and the indication would become “The CO2 concentration is normal” again.

Figure 4.15 is the PCB board layout for the indoor sensor. It includes the indoor temperature sensor, indoor CO2 sensor, and indoor humidity sensor. For the outdoor sensor we used the purf board to make all the connection to reduce the costing for fabricating PCB.

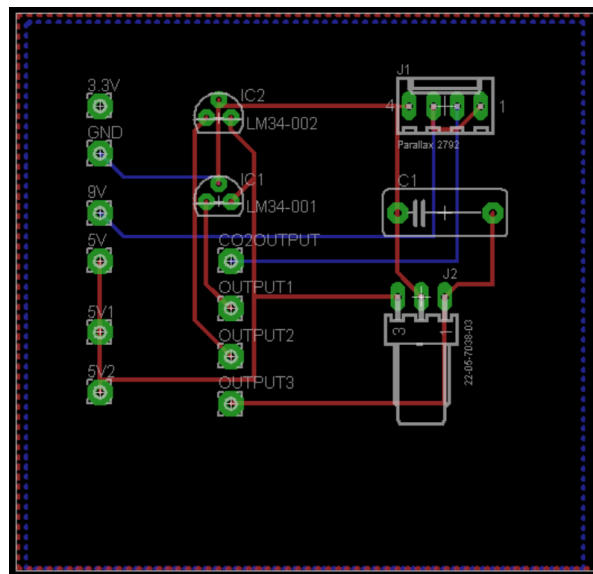


Figure 4.15. Indoor sensor schematics

4.4.5 Testing Machine to Machine Communication

Testing of the machine to machine communication from the hardware to the MySQL database was done two ways through an Ethernet connection as well as a wireless connection. The hardware was able to make this connection using the Ethernet Pictail and Wi-Fi Pictail board connected to the PIC32 microprocessor through an I/O Expansion board. To ensure that the hardware has established connection to the Internet we opened the sample website that Microchip provides that the hardware can only communicate to on any web browser, connections to the Internet is tested for both wireless and Ethernet. To allow the communication between the database and the hardware we programmed the microprocessor to act as a TCP client and when it receives the readings from the temperature, humidity, and CO2 sensors it will relay the values to an PHP script site designed for the sole purpose of putting these sensor values into their appropriate places on the MySQL database when button 1 is pressed on the PIC32 microprocessor. Showing that this process does operate smoothly we had the values from all of the sensors displayed on MPLAB, which is the software used to program the microprocessor and we pushed the button and updated our desktop browser and mobile web browser as well and saw if the values we had shown on MPLAB are the same as the browsers. Testing of this machine to machine communication was successful for both Ethernet and wireless connections, however for the wireless connection the hardware had to be connected to a wireless router with the specific name of “MicrochipDemoAP” just for testing purposes although the hardware could be configured to establish a connection with any wireless router in the future.

4.5 User Account Data Structure

In giving customers access to their accounts online we'd set up a prototype data structure consisting of pseudo coding that will provide all information regarding that user in regards to their account. The first part of the user account data structure provided the account information for the user to confirm that the information they are seeing belongs to their HVAC system. Each customer would have an individual customer number associated with their account on our data structure; we provided this number for our customers using the integral data type structure. After the user is provided with their customer number we next provided the first and last name of the account holder for the HVAC system, to provide these name we will use string characters to display the first and last name of the account holder. The next part of the account information that was provided for the user is the account number using a double precision number. Finally, the user was provided a username and password to access their particular account.

Figure 4.16 shows the basic data structure for a customer's account, should they "sign up" with Natural Air Energy.

Cust_Info

int	Cust Num
string	First Name
string	Last Name
Double	Account Num
String	Username
String	Password

Figure 4.16: Customer Information Data Structure

Following the customer information section of our data structure the user would be able to look at the user settings of their HVAC system. Starting off with the system status the user would have the opportunity to choose whether they want to have the HVAC system on or off, we set this up with int data type. A similar int type was used for the Reboot option. For this section of our data structure in user settings, the only change that the user would be able to apply in this section is to whether they want the status of the HVAC system to be on or off. The final option of the table is for the user to input their location using the State and Zip code inputs which are a varchar and int respectively. Figure 4.17 displays the database's structure for these user settings.

User Settings

int	Custnum
int	Reboot
int	Shutdown
varchar	State
int	Zip

Figure 4.17: User settings data structure

After, we provided the user with the account information to verify that they are looking at the correct account for their HVAC system, they were provided with the current settings that their HVAC system is currently set on. The first setting on the main screen settings is the mode of the HVAC system. A user would choose whether to be in cooling, or heating mode. They could also choose to simply turn the system off. The next elements under the current settings of the database is their temperature and humidity settings, which will be represented using integers. The user is able to determine the choice of what to have their HVAC system set on by looking at the outdoor temperature and humidity which would be listed under this represented by integers. Both the indoor temperature, humidity, and outdoor temperature, humidity are displayed in this table as integers. After the user can see all the temperature settings they can look at which system components of the HVAC system are currently set on or off. For this data structure we used integer and time characters to list each system component's value. A user could also see what their tolerance level is currently set at; we

would use an integer to represent the tolerance level ranging from the values of 1 to 5. The number one would represent the tolerance level for maximum savings and number five would represent the tolerance level for maximum comfort, the number two, three, and four would represent the tolerance levels in-between maximum savings and maximum comfort. Also included in the main screen settings part of our database is the user would be given the choice to turn their mood scents on the HVAC system on or off once again using Boolean data type. Figure 4.18 lists all the variables that can possibly describe the settings a user can manipulate on the basic screen.

Main Screen

text	Mode
int	Temp
int	Humid
int	Tolerance
int	Cust_num
int	Fresh_air
int	Air_quality
int	Scent1
int	Scent2
int	Scent3
time	scent_time
int	Current_Temp
int	Current_Humid
text	CO2
int	Outdoor_Temp
int	Outdoor_Humid

Figure 4.18: Current Settings Data Structure

The next part of our data structure covers the options available to the user under the scheduler tab in the HVAC system. The first element in the table is the Custnum, which shows the current user. The second element is a unique data type called the Trans Id. The Trans Id is short for Transaction ID, a automatically incrementing number that increases the number of rows in the scheduler table. This was necessary because in order to loop through all the database with constantly changing rows driven by customer's inputs. It provides a index that keeps track of the number of rows in order to avoid SQL query errors. Afterwards is the day and mode values for a new schedule to be assigned. The options that the user can make for a particular day are the temperature and humidity for that day of the week. Integer data type was used to fulfill all the data regarding the temperature and humidity settings for that day of the week. The final parts of the table include the duration, air quality, and fresh air options, all represented by the time data type. Figure 4.19 thoroughly explains all the variables and variations that need to go into a user's 7-day scheduling system.

Scheduler

int	Custnum
int	TransID
int	Day
time	Starttime
text	Mode
int	Temp
int	Humid
time	Duration
time	Air_Quality
time	Fresh_Air

Figure 4.19: Scheduler Data Structure

As stated earlier in the section there are certain aspects of the HVAC system that any user cannot make changes to and the only way changes could be made to these settings is by the administrator of the HVAC system. For the administrator to be able to make administrative changes to their HVAC system they have a set password giving them access to these settings that only they can change. The administrator has the same capability on our data structure by using the same password as on their HVAC system. We used a text data type for the administrator to enter their username and password. Figure 4.20 shows the basic structure for the Admin Login table. Figure 4.21 below summarizes the structure of the Administrator's MySQL table, differentiating it from a regular user's options.

Admin Login

text	username
text	password

Figure 4.20: Administrator Login Structure

Administration

int	AC1
int	AC2
int	Fan
int	NA
int	DH
int	Tolerance1
time	tempTime1
int	Tolerance 2
time	tempTime2
int	Tolerance3
time	tempTime3
int	Tolerance 4
time	tempTime4
int	Tolerance5

time	tempTime5
int	humidTolerance1
time	humidTime1
int	humidTolerance 2
time	humidTime2
int	humidTolerance3
time	humidTime3
int	humidTolerance 4
time	humidTime4
int	humidTolerance5
time	humidTime5
int	custnum
int	scent1
int	scent2
int	scent3
int	heap

Figure 4.21: Administrator Capabilities Data Structure

The final table is the Administrator settings that the administrator can edit only. Once the administrator has gained access to administrative capabilities they would be allowed to determine which installed system components should be on or off, the settings for each tolerance level of their HVAC system, and which mood scents that they would like to be activated. Int and time data types was used for the administrator to set the installed components of the HVAC system on or off. Changes could be made to the tolerance levels on the data structure using integers to represent the tolerance levels ranging from one to five, as well as for the temperature or humidity difference for each tolerance level and the duration that each tolerance level is activated. The data structure would only allow the administrator values ranging from one to five for the degree difference of the temperature tolerance level and one to ten for the degree difference of the humidity tolerance level. The administrator also has the selection of choosing which mood scents to have on using a int data type to state whether this mood scent should be on or off as well as a setting for HEPA. Once the administrator is done making all the administrative changes that they wish to make for their HVAC system they will have the option to save these changes so they can become effective to their HVAC system immediately.

4.6 MySQL Database

Creating a web database is an important process of providing all the necessary information consumers would need from their HVAC system in order to make appropriate changes to their HVAC system as needed. A convenient way of going about making this web database for customers is to use a SQL database query language, in particular using a MySQL database. In the previous section

we created a web database that we believe to be suitable for customers of an HVAC system and we expressed this database using SQL statements. Our plan for this semester is to have the sample web database created and use MySQL database server in order to bring this web database to life. The guidelines and definitions behind a MySQL database are summarized below in Figure 4.22.

Database	A repository to store data
Table	A part of a database that stores data related to an object, thing, or activity
Attributes	The columns in a table and each row to a column will possess the same attribute
Rows	The data entries stored in a table and provide a value to each attribute
Relational model	A formal model that uses databases, tables, and attributes to store data and manages relationships between tables
Database Management System (DBMS)	A software application that manages data in a database and is based on the relational model. Also known as database server
SQL	A standard query language that interacts with a database server
Constraints	Restrictions or limitations on tables and attributes
Primary key	One or more attributes that contain values that uniquely identify each row
Index	A data structure used for fast access to rows in a table
Entity-relationship (ER) modeling	A technique used to describe the real-world data in terms of entities, attributes, and relationships
Normalized database	A correctly designed database that is created from an ER model

Figure 4.22: Table of Database Terminology provided from O’Reilly’s Web Database Applications with PHP and MySQL

The process of creating our web database with MySQL database servers is to have our database managed by a database management system (DBMS) or database server. This database server would support a database language to create and delete databases and to manage and search data. The database

language that is used by almost all database servers is SQL, a set of statements that define and manipulate data, which is our database main objective of being able to define correctly all the data stored on their HVAC system to our database server. A MySQL database server that we created would carry the responsibility of managing our database, MySQL runs as a sever process or service and supports several different clients including a command-line interpreter and a Hypertext Preprocessor (PHP) function library. One MySQL server has the capability to manage multiple databases for multiple applications and each can store different data organized in different ways.

Within a database server we created database tables that served the purpose of covering specific sections within our database. For example in part of our database that we created in the earlier section we created a database table in reference to a customer of an HVAC systems primary account information. This database table included customer number, first and last name of the account, and the account number as well as their username and password. Going into further detail of a database table we have multiple attributes (also known as columns) in this table and each attribute carries its own name. From the previous example referencing to our database table the attributes of this table would be Customer Number, First Name, Last Name, Account Number, Username, and Password. The next detail in a database table would be the rows of the table, and rows of database tables bring values to the attributes in the database table. Returning to the previous example, the rows in our customer account database table carry different values based on each attribute. Customer number attribute carries the value of an integer, which would be the data type used in order to insert the number value. First Name and Last Name provide the value of identifying the primary account holders name of the HVAC system, the data type used to insert this value would be a string characters. The Account Number would also carry the value of a number, however it's data type would not be an integer since account numbers are generally large random numbers with no relation to the account before or after it and that is why we would use a double as our data type for identifying this value. Even though we did list data types that would be used for this example, this is just a small list of data types that could be used in creating a database using MySQL. Each database table in a database server is identified by using a primary key for that particular table, in our example of the customer account information the primary key of this database table would be the Customer Number. Our group expanded on these factors in the completion of the rest of our database using MySQL.

4.7 Smart Phone Application Design

Figure 4.23 below shows an overview for the primary pages to be developed for the smart phone application. While the goals was mimic the main pages of the main web site as much as possible, there was changes made between the two.

The second goal is to ensure the website works on mobile phones, even if this means changing the format of the mobile website from the main website.

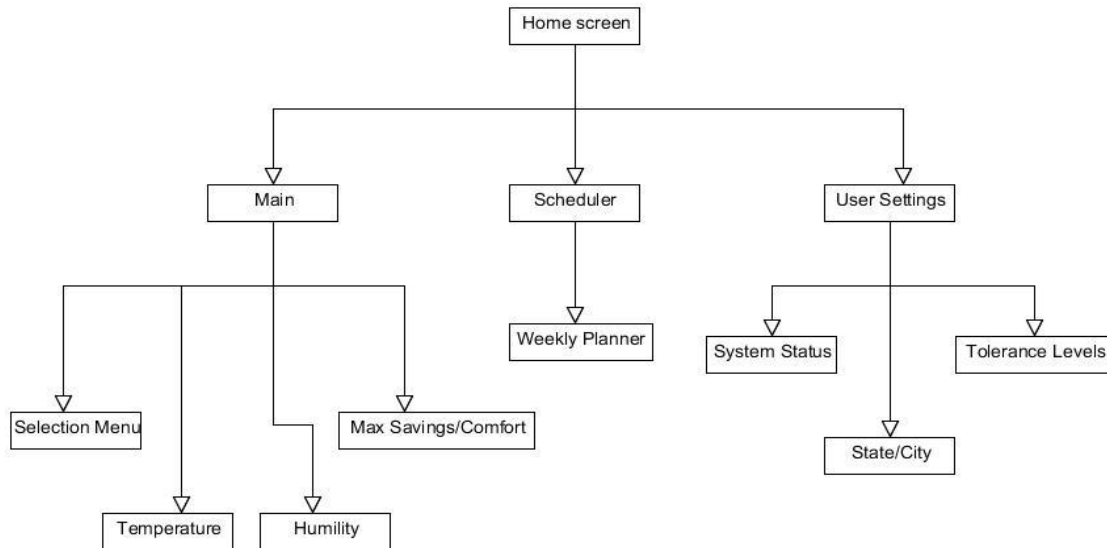


Figure 4.23: First Layout of Mobile Website

The home screen was divided into three sections, each of which will have a separate tab: the main screen, the scheduler screen, and the user settings screen. The first screen to display when a user accesses the website will be the main tab. The other tabs are accessible from their corresponding links.

Main Screen:

The main screen is the primary screen users interact with. It displays all the important data users will need, such as the temperature and humidity data, the current energy consumption and savings, and their respective buttons to adjust them. All of the usual settings from the selection menu are displayed here as well, such as the Air Quality and Mood Scents settings. It was imperative to have this screen be as similar to the HVAC system's original as possible. This way, consumers will usually only need to access this screen most of the time to perform their desired changes.

Scheduler:

The second tab leads to the scheduler page. In here a user can make a schedule for the system to follow for each day of the week. After picking a particular day to modify, a user will find all the settings available in the main tab. After confirming these settings, the HVAC system activates these settings on that every particular day. A user can control these settings up to the particular time

the settings will last, be it all day or only thirty minutes. All schedules created by the user are displayed on the appropriate day button display, in case the user forgets later on and needs a remainder.

User Settings:

All remaining settings not found in the main tab are displayed here. The important option to reset the system is found here, in case the system crashed or froze in some way. The status of the system energy and savings options is displayed here as well. Finally miscellaneous options such as city and state locations are set here as well.

Administrator Settings:

A page for the administrator settings was also added to the smart phone application. The possibility of an administrator having to access the HVAC system from a mobile phone is not entirely unreasonable. As such, it functions similarly to the original system. It requires a login password to access the settings the administrator can use. Here they can choose settings such as to kill or terminate the system itself, or personally modify settings for the system. Leaving from this tab will automatically cause the user to log out, regardless of whether the log out button was pressed or not.

4.8 Smart Phone Web

A vital component of our project is to make our HVAC system be accessible using a smart phone. In today's technological world, a majority of the smart phones in society consist of two major markets, the Android and the iPhone. These markets use applications for their smart phones to provide consumers with the items that they would like to access on their smart phone. However, the process of making applications for each of these markets can prove to be fairly difficult based on the fact that both markets require different programming languages to make applications applicable on either market. The Android market is the easier of the two to provide an application for due to the fact that you can make an application on their market free of charge and can use coding language that is quite simple, the most commonly used programming language is Java. However, for the iPhone market the process of making applications is not as simple as Android based on the fact that the iPhone market does require consumers to pay a fee in order to put an application on their market, in inclusion with that the language required to make an application on their market which is Objective-C, a programming language that adds Smalltalk style messaging to the C programming language.

To alleviate the difficulty of having to use multiple programming languages in order to make applications suitable for both the Android and iPhone application

market we made a mobile phone website. This mobile phone website will give off the representation of a phone application, while actually being a smart phone on a mobile website. This method removes the process of having to make an application that has to be compatible to an iPhone operating system in addition to an Android operating system. The mobile phone website also plays a helping role in giving smart phone users who are unfortunately not a part of any of the major phone application markets, but possess a smart phone with the capability of accessing mobile phone websites. Also, by creating a mobile phone website that can give off the appearance of a phone application will only require our group to use one programming language, since there are programs available that will allow you to make mobile phone websites that are compatible with both Android and iPhone markets.

A particular program that we used to accomplish the feat is Adobe Dreamweaver, which is a proprietary web development application formerly known as Macromedia Dreamweaver. Adobe Dreamweaver allows users to preview the websites natively in a preview pane or in locally installed web browsers. It provides transfer and synchronization features and a templating feature that allows single-source of shared course and layout across entire sites without server-side includes or scripting. The behaviors panel also enables use of JavaScript without any coding knowledge, and integration with Adobe's Spry Ajax framework offers easy access to dynamically-generated content and interfaces. This was beneficial in making our mobile phone website because we wanted to accomplish one user interface, making our mobile phone website show our consumers exactly what they would see on their LCD touch screen of their HVAC system. Dreamweaver can use third party extensions to extend core functionality of the application that can be written by any web developer. Dreamweaver is supported by a large community of extension developers who make extensions available both commercial and free for most web development tasks from simple rollover effects to full featured shopping carts. Dreamweaver also allows files to be edited locally then uploads them to the remote web server using File Transfer Protocol (FTP), SSH File Transfer Protocol (SFTP), or Web-based Distributed Authoring and Versioning (WebDAV). Any difficulties or issues that appear with the mobile phone website were addressed by our web programmer made corrections to the website and upload the corrections to the web server allowing quick maintenance to any technical difficulties of our mobile phone applications.

Version 5 of Dreamweaver supports syntax highlighting for the following languages:

- ActionScript
- Action Server Pages (ASP)
- ASP.net
- C#
- **Cascading Style Sheets (CSS)**
- ColdFusion

- EDML
- **Extensible HyperText Markup Language (XHTML)**
- **Extensible Markup Language (XML)**
- Extensible Stylesheet Language Transformations (XSLT)
- **HyperText Markup Language (HML)**
- Java
- **JavaScript**
- JavaServer Pages (JSP)
- **PHP: Hypertext Preprocessor (PHP)**
- Visual Basic (VB)
- Visual Basic Script Edition (VBScript)
- Wireless Markup Language (WML)

In addition to these languages, web programmers can add their own language with syntax highlighting. These choices of programming language gave us the option to choose what we believe to be the most optimal programming language in running our mobile phone website.

Section 5: Design Summary

5.0 Design Summary

In this section, a complete and detailed summary of the entire project and everything it encompasses will be given. It will be separated according to the main parts of the project. Each area will be described in more detail in the following sections.

5.1 Design of HVAC system emulator

The first part of the project design is the creation of a emulator designed to model after the actual HVAC system. This was emulated on a PIC32 microprocessor featuring all the inputs and outputs the system will send to the online database. These inputs and outputs are described as reads and sets within the system. Reads are data that the online database receives from the system or emulator and will display on the web site. Sets are data that updates the information on the database when there is a change in the system.

5.1.1 Layout of Project

The simulator for the project was a key part for ensuring the success of the website and server. It can be said that the software works perfectly by simply

reading whatever is in the server and displaying it appropriately, but without an actual device to test it on, its claims means very little.

The PIC microprocessor was able to provide actual data for the server to store, allowing the websites to access this data and display on their respective pages. From this, it will be possible to determine if the data is displayed correctly in the website. The relationship between these entities are displayed in the Figure 5.1 below.

The server as a whole, connects all of the entities of the project together. Both the websites and the microprocessor connect to the server database. The database is responsible for having all the data the microprocessor obtains from the air-conditioning and heating unit. The microprocessor sends this information and all its updates to the database. Then the main and mobile website read the information from the server and update their information on the appropriate pages. This works in reverse as well. When something changes within the mobile or main website, the server is responsible for sending this new information to all the other entities, which will update them appropriately.

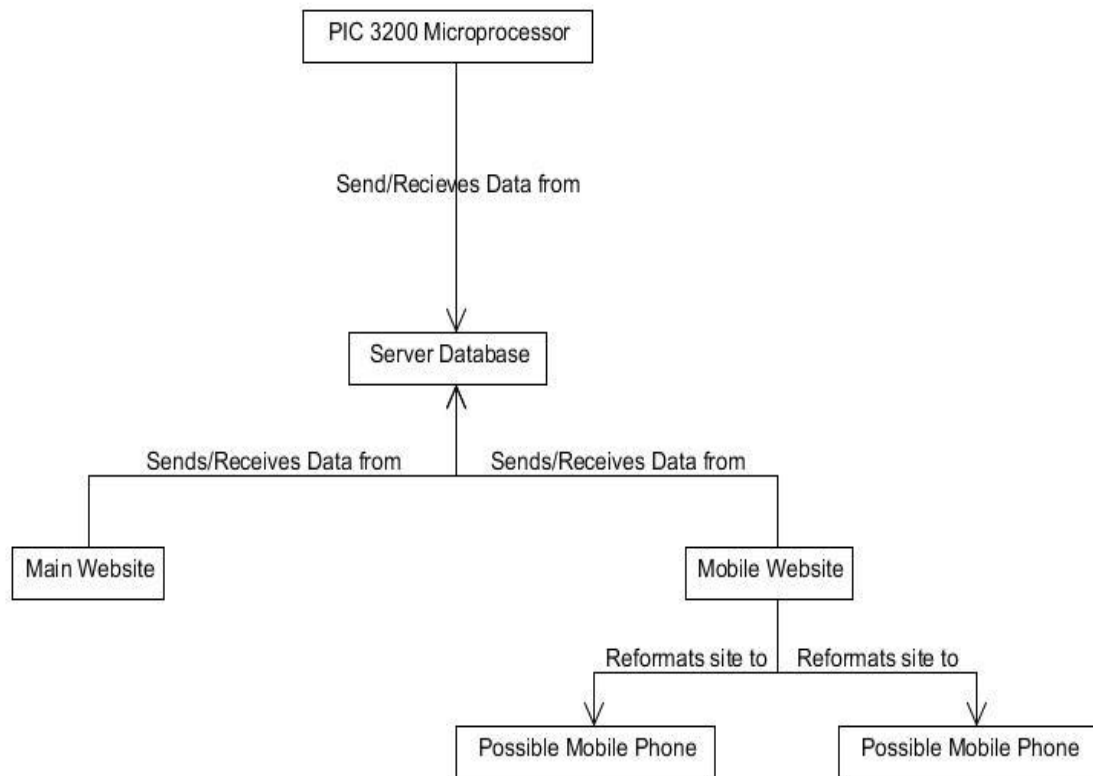


Figure 5.1: Relationships of All Project Entities

5.2 Design of Web Database Structure and Website

The second part of the project is the design of the web database that is to be used to store the data. A server was needed to provide the database to use, as well as provide additional features such as security. The specifics of the server to be used will be discussed in the next section.

5.2.1 Structure of the Database

The database has a very specific format in regards of how it retains the information stored in it. After all, the main purpose for the database is to store all the information regarding the site so the main website and mobile website can pull this information and present it to the user. The server identifies this information using a structure that displays all the objects within a particular page. Each object has several characteristics that describe the object. Some of these characteristics will be described below.

1. Type

This simply tells the server what type or primitive the object will be. All basic types from ints to strings to Booleans are used to describe the field entry. Since the server primarily uses the coding language MySQL, it needs this info to create the object.

2. Field

The Field entry serves as the actual name for that particular object. This entry is what separates the object from other similar objects, and what the websites will eventually display on their pages. The server will take the name of the field entry and that name is what is eventually displayed on the websites.

3. Null

This entry tells the server whether there is information to process when the website is first accessed. It is basically saying whether is a entry is null or not at the beginning of the connection. A yes means the object has no information yet in the server. No means the object always has information in the website and can never be null.

5.2.2 Customer Information Structure

The first structure in the database discussed is the Customer Information object. Here is where a user basic information is stored in the server. Each customer has their own object that identifies them with the system. This in turn, allows them to access all of their personal settings as well. There are six fields here. Cust Num is the number of the customer within the system. The next two fields contains their full name. The Account Num is their private number identifying them with their version of the website, with all of their settings. Finally, the username and password allow the user to have their own account without the interference of any other user effecting their settings. Figure 5.2 below shows the layout of the table for Customer Information.

Cust_Info	
int	Cust Num
string	First Name
string	Last Name
Double	Account Num
String	Username
String	Password

Figure 5.2: Customer Information MySQL table

5.2.3 User Settings Structure

The next structure represented is the User Settings page. Figure 5.3 describes the structure of the user settings page below. Features such as the reboot and shutdown objects are created here. Also included in the table is the location settings for that user. Below is a listing of all the variables required for storing data for user settings.

User Settings	
int	Custnum
int	Reboot
int	Shutdown
varchar	State
int	Zip

Figure 5.3: User Settings MySQL table

5.2.4 Current Settings Structure

The next structure object is current settings of the main screen. This structure gives the potential settings for the entire main page of the website. Several menu options are represented here, such as the outside and inside temperature and humidity, the CO2 display, and settings to control different devices, such as the Fresh Air and Mood Scents. Objects that don't require an immediate value when accessed are assigned as NULL in the server until they are changed by a potential user. Below, in Figure 5.4, is a listing of all the variables required for storing data for current settings.

Main Screen	
text	Mode
int	Temp
int	Humid
int	Tolerance
int	Cust_num
int	Fresh_air
int	Air_quality
int	Scent1
int	Scent2
int	Scent3
time	scent_time
int	Current_Temp
int	Current_Humid
text	CO2
int	Outdoor_Temp
int	Outdoor_Humid

Figure 5.4: Main Screen MySQL table

5.2.5 Scheduler Structure

The next structure created by the server is the Scheduler page of the website. Each object is referred once no matter which day is chosen. As usual, the corresponding objects that don't necessarily require a value are given NULL statements. For the most part, most of the settings in the scheduler mirror the settings of the main screen structure. Figure 5.5 gives a prototype version of the page structure below.

Scheduler	
int	Custnum
int	TransID
int	Day
time	Starttime
text	Mode
int	Temp
int	Humid
time	Duration
time	Air_Quality
time	Fresh_Alr

Figure 5.5: Scheduler MySQL table

5.2.6 Administrator Structure

The last main object structures in the server are the Administrator structure and the Admin Login structure. It is shown in Figure 5.6 and Figure 5.7 below. The Administrator structure creates the layout for the Administrator page in of the website. It includes all of the settings available to the admin when they access this page. The Admin Login stores all the administrators and their password information that allows them access to the admin page. Obviously, the password number will be NULL, until a specific password is given to the account for an administrator to access. Important exclusive settings here include the tolerance levels for both the temperature and humidity as well as the hardware components a user may have in their home. The admin has the power to determine which hardware components will be available as well as the specific tolerance settings the user will have.

Administration	
int	AC1
int	AC2
int	Fan
int	NA
int	DH
int	Tolerance1
time	tempTime1
int	Tolerance 2
time	tempTime2
int	Tolerance3
time	tempTime3
int	Tolerance 4

time	tempTime4
int	Tolerance5
time	tempTime5
int	humidTolerance1
time	humidTime1
int	humidTolerance 2
time	humidTime2
int	humidTolerance3
time	humidTime3
int	humidTolerance 4
time	humidTime4
int	humidTolerance5
time	humidTime5
int	custnum
int	scent1
int	scent2
int	scent3
int	heap

Figure 5.6: Administrator MySQL table

Admin Login	
text	Username
text	Password

Figure 5.7: Admin Login MySQL Table

5.3 Design of the Web Site to host the system

The third area of the project is the design of the web site itself. It's primary job is to obtain the data from the database and display it onto the website itself, using the same interface as the HVAC system. The website was built using the typical resources such as MySQL and HTML. The design specifics are mentioned below.

Many languages came into play in terms of the design of the web site to host the system. HTML, PHP, MySQL, and JavaScript all played a role into the creation of the site. The site was designed so most of the common browsers will be able to

display it, such as Mozilla Firefox, Google Chrome, etc. The site contains full functionality of the touch screen system that the main HVAC system will be using. This required the use of the more complex languages PHP and JavaScript.

5.3.1 Design on Web Pages

The design of the web pages that is compatible with the computer and smart phones systems is one of the biggest goals of our project. In order to achieve this goal, the HTML played an essential role for this part, especially the HTML form that was mentioned above in the research. First, the user will go to the main website of HVAC system and enter their account and password so that the server can search their information from their database and retrieve the related information from the server, like temperature and humidity. Then, the user accesses and enter their specified HVAC system controller.

Main Menu Design

First thing the user will see is the Main menu screen like in Figure 8. On the top of menu, we will have to add 4 buttons which link to different pages (Main, Scheduler, User Settings, and Administrator). Below is the design for our menu. A user can click on any on them, like main or scheduler in the figure below and the user will be directed to the respective web pages.

On the right panel are various options to be selected - Fresh Air, Air Quality, and Mood Scents. This can be done using the same techniques as the menu option design to create different buttons so the design will match with Figure 5.8. Each and every button is linked to different web pages for the different functions. On the bottom of right panel is the selection of "Fan (Auto)" or "Fan (On)". Because these two options can only exist by themselves, the Radio Buttons are the only option to make the user choose one and only one option, the figure below is part of our original design. One thing to keep in mind is that if a user clicks on one of the two radio buttons, it becomes checked and all other radio buttons becomes unchecked corresponding with that name. If one has two different names for these two buttons, a user can choose more than one option without any difficulty. However, it is not our intention to use radio buttons. It resembles more like the checkboxes function. A diagram of the design is shown in Figure 5.8.

On the middle panel of our main menu, the page needed to use HTML scripts to display the inside and outside temperature and humidity. It could also be done without using HTML scripts. On the bottom of the middle panel is the display for the selection of modes by showing different bars which will display on the right panel. It needed to display the date, city, and state where the HVAC system was installed. All of that information could be display using the same technique. On the right panel of the main menu is the mode of the HVAC system that contains 5

bars from Maximal Savings to Maximal Comfort. We used JavaScript in this portion.



Figure 5.8: HTML design of web pages

Scheduler Design

When the user clicks the scheduler tab, first thing the user will see is the Week Planner and will need to choose the day of the week from Monday to Sunday. We made changes to the appearance on the website compare to LCD screen for efficiency purpose and convenience of design. Because each time, a user can only make one schedule, so it best to use radio buttons to implement this as the figure above shows.

When a user clicks the day of the week, the user will be directed to other pages that contains some drop-down lists for starting time and ending time of the new schedule, and also need the temperature and humidity control buttons for them to set up the desired range for the temperature and humidity. The figure above shows the possible design. After that, the user will be required to click the “save change” buttons in order to save the changes and the schedule will be sent to the server and save in the server’s database for future use or changes. After you set up the schedule, the user will be directed to the pages that show all the future schedules for reference and allows user to change it if they want to.

User Settings

When user clicks the user settings, they are led to the page that allow users to change some of the settings of the system. For example, a user can decide whether to turn on the machine or turn it off. Here, we only allow user to choose only one option between two so the “Radio Buttons” is the HTML form to implement it. It will also need the user to click save changes before the change is made. The tolerance level setting is displayed to user by using HTML scripts but the user is not allowed to make any change of this portion. Only a technician who had a valid password has the right to make any changes. On the bottom, we made a drop-down list the state, and a box for the zip code that a user can modify. Above is the possible design for our project. When the user chooses their state from the drop-down list then the system will show respective cites for that particular state. This means we needed to set this relationship in the server’s database first.

Administrator

Frankly speaking, a user cannot change many things in HVAC system, because most of the parameters can only be decided by the company for safety and efficiency reasons. However, an administrator has been given much more power on changing the system. But we needed to set up the user account and password for protection, which will only allow the technicians to access it. In order to do that, we used “Password Field” in HTML form to do it. Above is the possible design for password settings. In Figure 48, a username will use “Text field” to do it; however, the password part need to use “Password field” for safety reasons. Both of them can let a user to enter one-line input, but they are totally have different functions. In “Text Field”, a user can type anything they want for one line and it can be displayed on the screen explicitly. For example, when someone types “type username” in the blank after username, it will display exactly what they may type in the blank. However, when someone types something in the password field on the web pages, they will not be able to see them directly; otherwise, it is really easy to be glanced by others. After you successfully enter the password, they will need to click the submit buttons to send this information to the server for matching. If one enters the right username and password, they are directed to the administrator pages.

After that technicians enters, they should see a web page with many options. In our web pages, we designed two buttons for “Kill Application” and “Log out”. The “Kill Application” button will close the web browser to end all the HVAC control and “Log out” option will only log out from the administrator mode to user mode so the user can control the HVAC system again. After these two options, we made checkboxes for “installed system components”. And these checkboxes will be various options like AC1, AC2, FAN, NA, and DH. An technician can check the components for this portion and the HVAC system will automatically turn it on or off. Then we made 3 buttons for selection options- Fresh Air, Air Quality, and Mood Scents. If one chooses Fresh Air, the page will make a drop-down list letting an administrator choose the timer for when the system will allow the fresh air to come into the house. The options will be 5 to 40 seconds with 5 seconds intervals in between. We also made a done button for saving the changes to pass the changes to the server. If one choose “Air Quality” option, we will make 3 checkboxes letting an administrator to choose which filter is installed (filter 1, filter 2, and filter 3). We will also make a drop-down list letting an administrator to choose a timer to allow the fresh air to come into the house and once again, the options will be 5 to 40 seconds with 5 seconds intervals in between. We needed to make a done button to save the changes. The last option will be “Mood Scents”. For this part, we needed to have various checkboxes to choose which one to activate and also need a timer to control the spread frequency. The “done” button will save any changes that was made.

For the last part of the administrator page, we created 5 different tolerance levels ranging from maximal saving to maximal comfort. An technician will be allowed to change each level’s temperature or humidity difference and running time. Both of them are implemented using a drop-down list. The default setting for the temperature difference is from +/- 0 to +/-5, the default setting for the humidity difference is from +/- 0 to +/- 10, and the default setting for the time is 5 seconds to 60 seconds. All of these are accommodated according to the sponsor’s need. One thing that an administrator cannot change is the Maximal comfort because in our design, it was automatically assigned to +/- 0 for this part.

5.4 Design of the Mobile Phone Application

The final main section of the project to be discussed is the mobile phone app. The app was designed to work on both Android and iPhone. The software to be used will be coded in Java or C# primarily. The following section will describe the design in more detail.

The main software used for developing the mobile website of the project will be Dreamweaver, a software that can develop for a website in several devices at once. These devices to be designed for included the iPhone and Android configurations. This creates a site for the two most prominent mobile systems used in the U.S. Once most of the main website was functional, work was begun

to configure it for the mobile applications. The process for designing the mobile website was similar to the main website, the main difference being the languages used. Also, the website was designed to remain as close as possible to the main website while keeping it viewable for a mobile audience.

5.4.1 Common Features of All Pages

While all the pages of the mobile website are distinct and different from each other, all of them share common traits that will be designed for the pages. These are listed below.

1. Scrolling Bars

All of the pages have scrolling bars available to them. This is important so users will be able to see everything in the screen. Despite the variety of phones in the market, every phone is able to display the website perfectly. Scrolling bars allow the user to access the rest of the page in the case it isn't formatted properly.

2. Buttons

All of the pages use the basic button to choose from various settings, initiated from a click or the touch from a touch screen. Each button has differences in their function, such as extending and displaying a drop down menu, inputting a character into a text box, or simply opening that particular pop-up menu. The button is a simple feature, but an important one to the function of the system as a whole.

3. Pop-Up Boxes

Many of the pages have used pop-up boxes to display more options within the page. Pop-up boxes allow a user to access more options within that page without having to display all the information on the page itself, which saves space and data. This is especially true if the user doesn't access the data every single time he or she views the page. If the user needs to open a menu box, the page can obtain the data needed for it when the access call is made. Otherwise, the page has no need to load the data until necessary.

4. Text Boxes

Two pages of the mobile website use text boxes to display information. The Scheduler tab takes a schedule created by the user for a particular day and displays it in the day display button on the page. While the user himself won't be typing anything, the information inputted, such as the time to start, and duration, will be automatically type into the display of the day schedule table. The Administrator Settings page uses a text box for the password to enter the admin main page. The password can be entered by directly typing it in from the phone, or using the keypad provided on the page.

5. Display Images/Buttons

The site as a whole is reliant on presenting data on display boxes or images. Information such as temperature, humidity, and which savings/comfort level the system is set, and need to be appropriately displayed so the user can see it quickly and easily. These images are dynamic, as they will change depending on the data received by the server database. It should be clear to the user which button settings is currently pressed and should change accordingly when a user changes the setting. Basic features like these may seem frivolous, but are important to providing an excellent and efficient website for users.

6. Connection to the Server

One crucial part of the project is that the mobile website is capable of obtaining information from the online server database. After all, a website that can't display the correct data for the user is essentially useless. The mobile website has to constantly monitor the server for changes being made to the data. Or the server contacts the website when a change in data occurs. Regardless, when the website sees a change in the data, it must update the data accordingly on its own page. Also it needs to send any new data made by the website itself to the server. Then the server can send the new info to the other systems connected to it so all of them have the same correct data.

5.4.2 Main Page

Here is the design detail for the main page of the HVAC system in Figure 5.9. As such the mobile application is designed to be as similar as possible to the original system. The main page gives users the basic settings and controls for the HVAC system. This includes the various setting options, and the maximum savings and comfort setting bars.

Selection Menu

The selection menu will display buttons for all the basic settings for the HVAC system. The settings are Fresh Air, Air Quality, and Mood Scents. Each setting will likewise open up another box which will provide all the options for that particular setting. The Fresh Air setting controls the amount of fresh air in the system. Air Quality adjusts the quality of the air for the system to control. Finally Mood Scents allows users to add a mood scent to the HVAC system if they so desired. All three menu options will be similar in structure, design, and appearance for the most part.

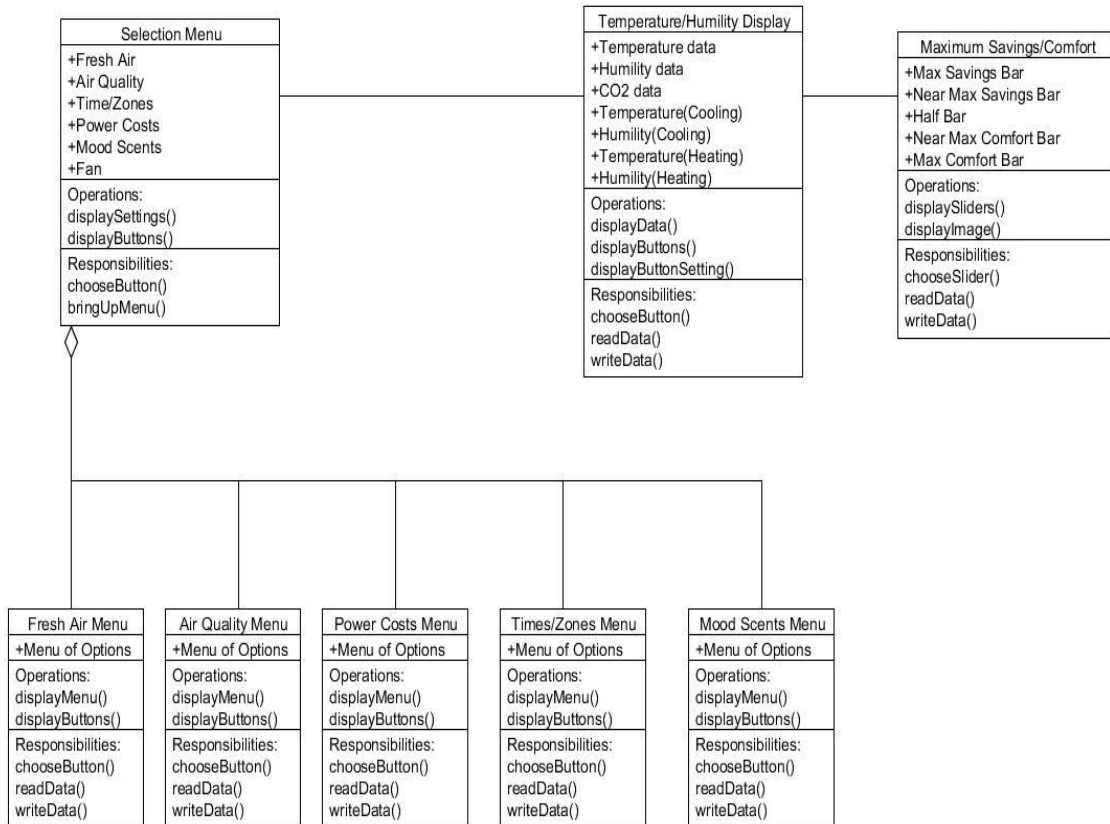


Figure 5.9: Main Page (Mobile) Class Diagram

Temperature/Humidity Display

The second part of the main page displays the current temperature, humidity, and CO2 rating for the house the system is monitoring. Read and write functions obtain this data from the main server to be displayed here using the display function. The menu here includes up and down buttons for both the temperature and humidity. Users here will be able to adjust the temperature and humidity ranges using the up and down buttons which sends the data to the server and update the main system accordingly to their specifications. This will be the main operating menu unit most users will interact and access.

Maximum Savings and Comfort Menu

The final part of the main page is the Maximum Savings and Comfort adjustment bars. The options here allow the user to change how much of the system is devoted to running the air-conditioning unit. The higher the Max savings bar is selected, the more the HVAC system will refrain from running power to the unit. At Max comfort, the unit should behave like any other air conditioning and heating system. The differences between the slider bars are determined using

the time/zones menus from the selection menu area of the main page. This section makes up the final third of the main page a user will be able to access, which is also the most unique part of the HVAC system.

5.4.3 Scheduler Page

The next page designed is the scheduler page. This page allows the user to create a schedule for the HVAC system to follow automatically, without receiving further input from the user. The scheduler page contains a separate schedule for each day of the week. A user will choose a week and then choose settings, the time to start, and the duration for those settings. The layout of the page is shown below in Figure 5.10.

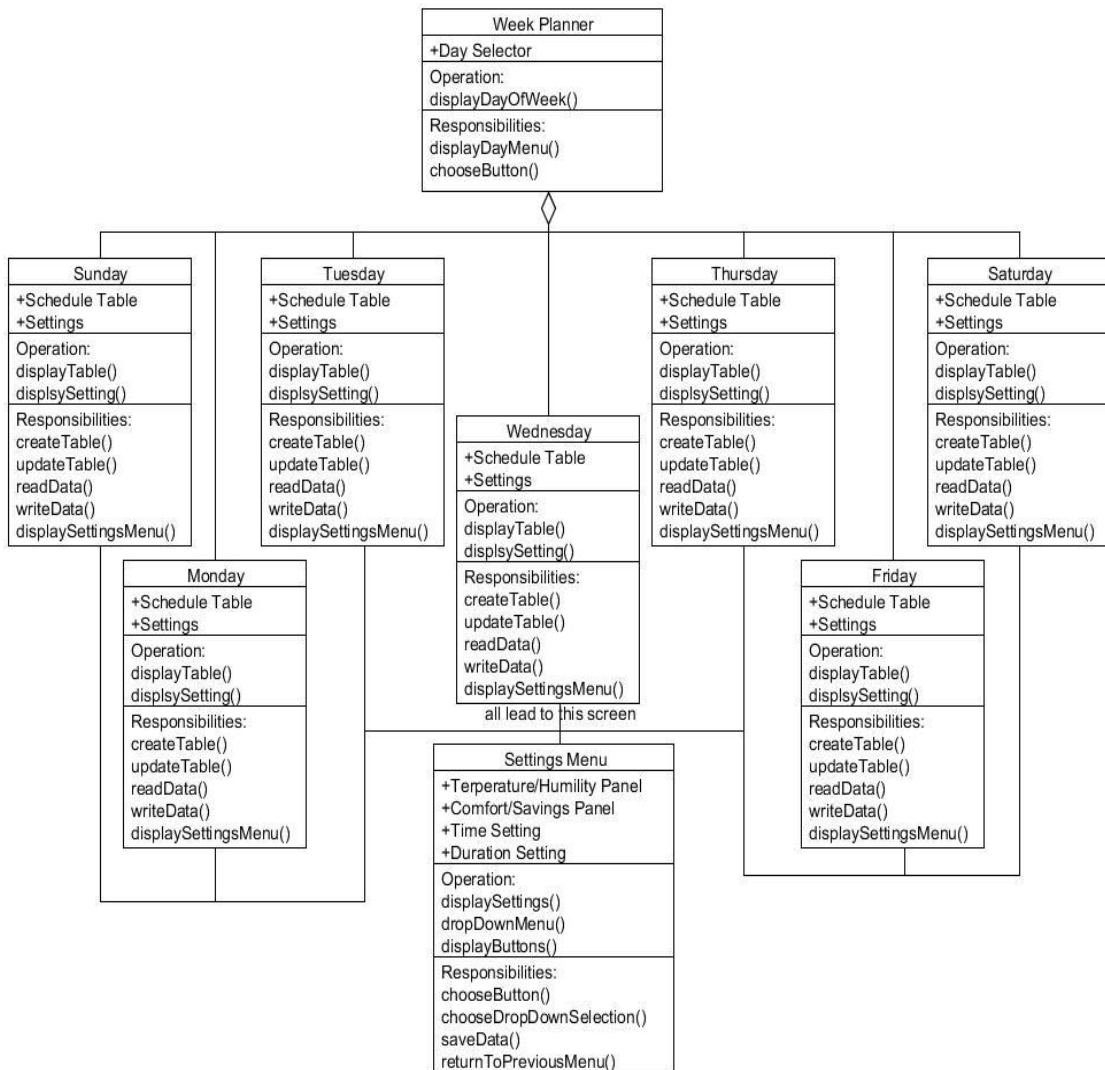


Figure 5.10: Scheduler Class Diagram

Main Schedule Page

The first page shows the buttons for each day of the week, as well as the current schedule for that day, if there is one. Here the user can check if his or her current schedule is acceptable for their system. If not they can make adjustments by clicking on the specific day they wish to modify. Choosing that day will change the page to the next page, the specific day's schedule menus for that week.

Day Schedule Page

The next page gives an detailed look of the day's current schedule display on a table. The table details the time the system is on, the duration the system will be on, and any additional descriptions about the schedule. There are two buttons that lead to additional settings pages. These settings pages will be what the user will modify to create the schedule for that particular day. All changes made by the user will show up on this table for the user to look over to ensure it is satisfactory. Otherwise they can continue to make changes as needed.

Day Settings Page

This page is the settings for the user to modify to create their specific schedule for a particular day. While the user only makes changes for that day, all the days use the same settings screen. The page includes most of the settings found in the main page, such as the temperature and humidity settings, as well as the maximum saving and comfort settings. There are also additional settings specific for this page. The time setting is a drop down menu displaying various times. A user can choose at what time the system will take the current settings into effect. The duration setting is also a drop down menu. It determines how long the HVAC system will use the current settings shown in the day's settings page. There are additional pages that link to the settings to adjust the max savings and comfort bars as well as the fresh air venting settings. Everything the user inputs here is finally saved and updated into the table in the schedule page as a new entry in the schedule. The user can then enter a new entry if he or she chooses to.

5.4.4 User Settings Page

The next page is the User Settings page. This page is intended to be much less used by the average user than the other pages. Here the user will find settings that won't require to be changed as often. Also the reset option is here, in the unfortunate event the user might need to use it. We hope the user will never need to use this option. The basic layout for this page is given in Figure 5.11 below.

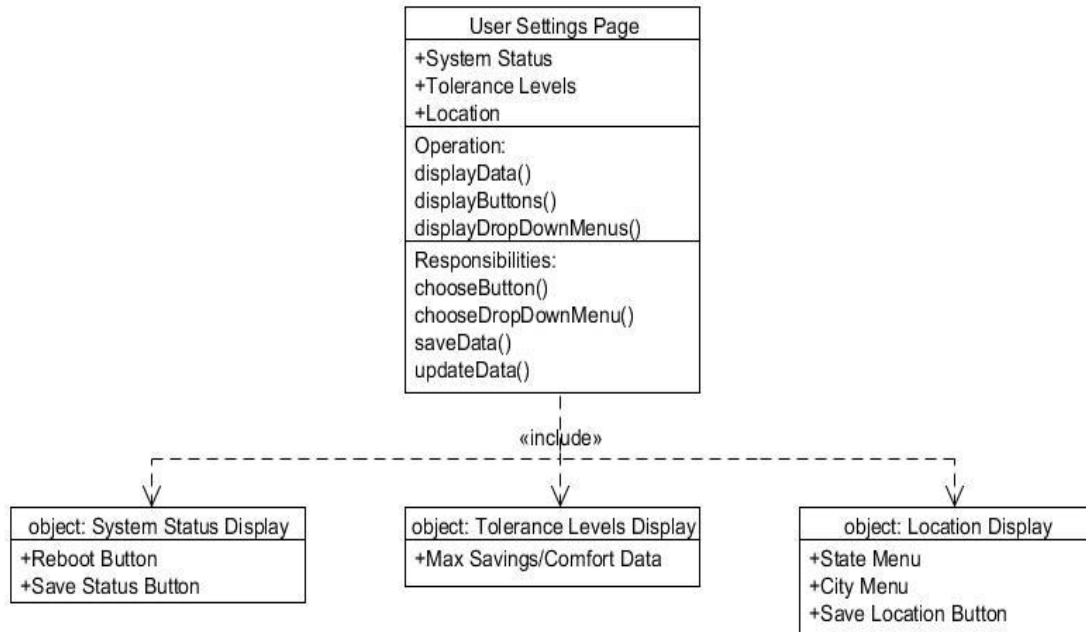


Figure 5.11: User Settings Class Diagram Basic Layout

The User Settings page has three main sections; the System Status display, the Tolerance Levels display, and the Location display. Except for the Tolerance display, the other two have settings the user can choose to modify. The Tolerance cannot be altered as only the administrator have access to modify it. Otherwise, the settings here will update when the user saves all the changes in this page.

System Status Menu

This screen is on the top of the main user settings page. Here the user can choose to make the very important decision to reboot the system. Sometimes the system will refuse to respond or communicate correctly with the server in which case resetting the system can hopefully fix the connection between the devices. Also the user may just merely want to shut down the HVAC system and the option is also present in this screen. Thusly, the user will have the option to reset or shut down the system if the situation is necessary.

Tolerance Display:

The Tolerance display, as mentioned above, only displays the data at present. The user will not be able to modify the data here, but at the main screen, under the power costs menu. The Tolerance Levels show the difference between each level, in terms of how much variation the system will allow in power and temperature, time differences. The higher the level, the less variation allowed for

the system, and the more power that will be used to stay within that range. This unique data is difficult to be able to display in the main page, thus its appearance in this page instead. Once the user understands these settings, he or she can easily switch between the levels from the main page. Nevertheless, these settings chosen will be shown in detail here if the user should ever need it.

Location Menu:

The third part of the User Settings page, the Locations screen, is at the bottom of the page. Here the user can input the location of his or her HVAC system. There is a drop down menu displaying all the states in the U.S. A text box is provided for the user to enter their zip code. After the user enters this information, there is a save location button to save the user's input. Relatively simple, the user shouldn't need to adjust this setting all that often, only if the system happens to change locations, such as if the user decides to move, taking the HVAC system to control his or her new home air-conditioning and heating system.

5.4.5 Administrator Settings Page

The final main page of the HVAC system mobile website is the only one users probably will never visit, the Administrator Settings page. This page, only accessed by a technician of the HVAC system, allows the admin to play with the settings of the system. The admin gains access to this page by entering a password in the log-in screen. From there, the admin or technician will have access to the rest of the administrator page's contents. A layout is shown in Figure 5.12.

Log-In Screen

When this page is first accessed, the log-in screen will appear. Here a user must input a password that will allow access to the rest of the page. The password will be a series of numbers that will be entered from the keypad provided. A correct password will allow that person into the main Administrator Settings page. If the person is unable to give a correct password, nothing will happen, and that person will be stuck there. In the main Administrator page, the user will see three areas that can be accessed and modified. These are the Installed Components screen, the Tolerance Levels screen, and the Selection Menu screen.

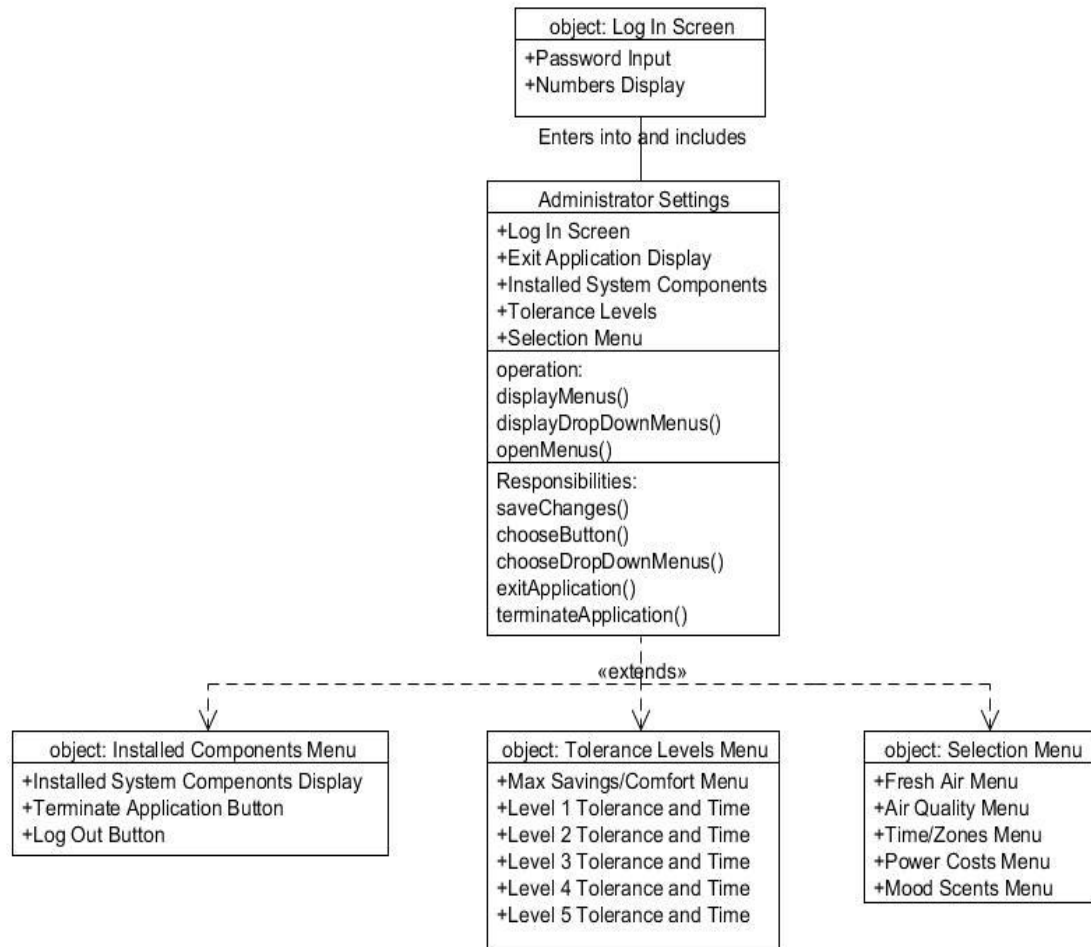


Figure 5.12: Administrator Settings Class Diagram

Installed Components Menu

In this screen, all the installed components that are connected to the HVAC system are displayed. The user will have the option to choose which device will stay connected and which devices to disconnect. If there is a particular problem with one of the devices, it can be isolated here, to ensure none of the other devices are affected by the problematic device. The other option here is to terminate the device immediately. If the user deems it necessary, he or she can choose to disconnect the system from every device, perhaps if the system itself is causing the problem. Finally, the log out button is here. When the user is done examining the page and wishes to exit, they can click on this button to return to the original log-in screen. Leaving the page by choosing another tab will also automatically cause the user to log out.

Tolerance Levels Menu

The next section is similar to the display with the same name in the User Settings page. Anyone seeing this page will see the data for the tolerance levels which is the same as the previous page's data. However, the user this time is able to modify the information regarding each level. The temperature variance and the time duration are both fully modifiable using their respective drop down menus for the user's choosing.

Selection Menu

The selection menu is for the most part, the same as the selection menu screen in the main page of the website. All the same options for the HVAC system are available to view and change for the user's convenience. This way, the user can stay in the same page and not have to log out of the Administrator page in order to change the settings of the system.

5.4.6 Final Layout of Mobile Site

The final section will describe the layout of the entire website, as shown in Figure 5.13. While there is no central hub, every page can access every other page as needed. This is performed by having links displayed as tabs for each page. These are displayed on the top of each page allowing easy access from one page to another.

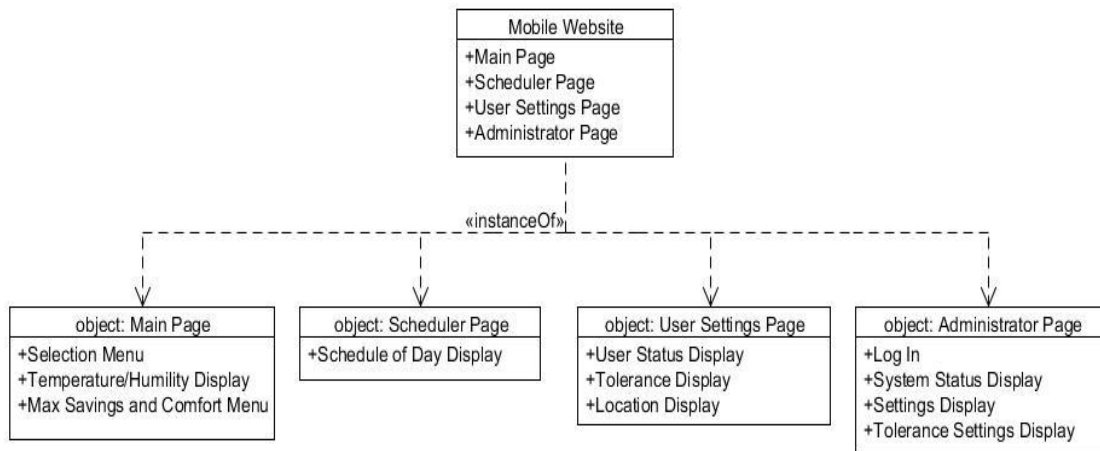


Figure 5.13: Layout of Entire Mobile Website

As the figure above shows, all the pages are essentially tabs connected together. Upon entering the web site, users will begin in the main page tab. From there the other pages are easily accessed by using the tabs to hop from one page to another. All of the information and data here is taken from the main server

database on the host website. The mobile website will connect to the server and update all of its information from the server. This way, the user will be sure that the information regarding his or her system is correct.

5.5 Simulated HVAC Hardware

Senior Design Team 16, Proton Design, and the previous Senior Design Team 2 can all agree that the free TCP/IP Stack is an incredibly useful component to creating the necessary hardware capabilities to simulate the HVAC system, so that the other components can communicate with it. The original plan was to create a simulated HVAC system by arbitrarily programming one with C or C# as Proton Design Inc. suggested. In fact, redesigning the software for the HVAC system was not included in their contract. While that is one alternative, there is a great opportunity to fully simulate the HVAC system by programming the exact same processor that is being used for the next edition of the HVAC system's hardware, the PIC32 microprocessor.

Section 6: Testing

6.0 Testing Requirements Summary

The entire project can be described as several small projects all tied together by the Internet. Due to the nature of this project, testing was a constant, ongoing challenge throughout the project. The four major components of the project are the web database, the browser based HVAC controller, the smart phone application, and the simulated HVAC system. As they all being developed, they were all rigorously tested to understand the conditions in which they work and do not work. The order in which they were tested is the same as the order in which they were created. First the MySQL database was created, and initially, it can only be evaluated by inspection. The second component was the website for observing and manipulating the database, and only when was created would the database be truly tested. In turn, the website was tested for its ability to both manipulate the database, and accurately represent an HTML based version of the existing HVAC control system.

As the smart phone application and simulated HVAC system are developed, only when they are were completed can the entire project be truly tested, because the true test was making sure all four components are successfully tied together through the Internet. Without the simulated HVAC system, the smart phone application couldn't access and make changes to the MySQL database. Without the browser and smart phone HVAC controllers, the simulated HVAC system couldn't receive commands from the database, as well as upload it's data to the database. When all four elements were ready, they were fine tuned to all work at the same time without interfering with each other.

6.1 Hardware Wireless Testing

Late in the fall semester, Team 16 learned of Proton Design's selections in making the hardware cheaper and more suitable for mass production. Originally, Natural Air specified that the ultimate goal of the project was to create the web application. The addition of hardware allows us to take the project further by testing the PIC32 microprocessor's ability to interact with the high level database and software we created. The original prototype that the previous senior design team made is in the care of Natural Air, and we didn't have access to it, because Natural Air uses it to show to shareholders and other business partners to this day. It did not change the fact that we could use the new hardware to test its ability to download and update variables from the MySQL database we create. The hardware was tested by performing simple POST and GET commands and tested its ability to interact with the database. The PIC32 needed to be able to successfully perform these commands and the database needed to be able to make the necessary changes. A sample change would be changing the requested temperature and humidity settings, and having the database, web browser, and smart phone application picking up those changes.

6.2 MySQL Database Testing

Early in the development of the system, the MySQL Database was one of the earliest components produced in the project. It is one of the easiest parts, but it was also the most vital, and so it was tested rigorously throughout the spring semester. It was created almost immediately when production began in the spring, and initially, any testing performed upon it will be done via inspection. A single HVAC system from our client requires a moderately large amount of data. Initial evaluation of the database will be making sure every possible piece of data is represented within the database. What goes into the database has already been designed by Kevin as of the fall semester, but once it has been created, it must be further inspected to make sure that not only every variable is represented, that it is represented by the correct variable type and that there are no redundancies contained within a single user object. Only when the other components of the database are completed can the database be truly tested. The other components need to be able to read and write from the database. Furthermore, each component must be able to understand changes made from the other devices. For example, early into the project, the web based HVAC controller must have been able to manipulate the database, but by the end of the project, the smart phone application must have been able to manipulate the database and the web browser controller must have been able to pick up those changes.

6.3 Web Browser HVAC Control System Testing

The web based HVAC Control System will be the second component of the project to be tested, and like the MySQL database, it was tested frequently throughout the course of the project. First, the web site must have been inspected visually to ensure that it matches the original touch screen HVAC system as closely as possible in function. Initially, it would look dramatically different from the original system, because aesthetics are a very low priority for the site. This did not change the fact that the web site needed to be easily understood by the user. At minimum, all the website's pages needed to be equivalent to all the screens of the touch screen HVAC system. Every possible screen contained within the original touch screen needed to be represented in the browser, so that users have the same amount of control from their computer that they do from the touch screen. (Making sure the browser and touch screen work together came later.) As the database needed to contain every necessary variable for a user, the web HVAC controller needed to be able to manipulate every item in the database the way the touch screen would. Every variable in the database needs to be accurately represented on the web site's various forms, making the appropriate posts and queries depending on the data being manipulated. Browser compatibility was tested with a variety of browsers, including Mozilla Firefox, Microsoft Internet Explorer, Google Chrome, and Apple Safari. Many students and developers agree that Firefox and Google Chrome are the "best" browsers for using the Internet, but if Natural Air intends to sell this to homeowners of every age and community, Team 16 could not jump to the conclusion that these browsers have conquered the favor of every consumer across America. Although the web browser was developed with Firefox in mind for ease, adjustments were made for Internet Explorer, Safari, and Google Chrome.

Due to time constraints, nearly every web page needed a Submit and/or Update button in its initial design. Each page was static and needed these buttons to submit changes created by the user, and to detect changes in the actual or simulated HVAC system. Giving the website the ability to dynamically detect changes in data such as temperature and humidity rising and falling requires AJAX architecture within each page. This is highly ideal for the project, and if we had the time to implement it, it would be implemented in every page to detect changes as they occur.

6.4 Smart Phone Application Testing

The Smart Phone HVAC controller was one of the later parts of the project to be tested, at a point in the project in which the database and web browser are already tested (keeping in mind that all testing was ongoing throughout the

project.) The testing of the smart phone application is particularly extensive because it is the ultimate goal of the project, the key product that the client wants, and everything the other elements of the project were building up to. First, unlike testing for functionality first like the browser application, the application must be inspected aesthetically. At this point in the project, the database and browser controller was tested already, so the first priority for the phone application is making sure it looks like a professional application for a smart phone. This was evaluated for both Android and iPhone. Daniel owns an Android phone, and Kevin owned an iPhone so both phones could be inspected visually. The application needed the right amount of screens to accurately represent all the options that the user has on the browser controller, and the touch screen itself. On the phone, space will be limited, so the smart phone's menus and screens will not be symmetrical with the touch screen or browser, yet the application needed all the same options and capabilities. After visual inspections on the smart phone application were complete, its functionality was tested. All its options and statistics must have successfully post and query the database, and ultimately control the HVAC system no matter where the user's phone is located. Like the web application, the phone app needed to be inspected for compatibility. The same applied to the smart phone application, but the browser priority changes to the Android Browser and iPhone browser, and then Firefox and Google Chrome was inspected afterward.

6.5 Web Security Testing

The client specifically asked for the product to be secure. As the client takes over our work and either builds upon it or has a third party utilize it, the web security measures will have to be change based on business models and architectures that are developed as the system is expanded. That does not change the fact that our team can make sure what we created is secure. Security is not exclusive to worrying about malicious attackers. It must be secure in the sense that one user cannot interfere with the other accidentally. In a typical household, the product will have three different ways of being used. The first is the touch screen, the second is the web browser, and the last is the phone application. One user must not be able to interfere with the work of another. If one user is interacting with the screen directly, the other users need to know that. The browser application should warn users that the touch screen is being used. The phone application should be able to detect that the web browser or touch screen are in use. The future edition of the HVAC touch screen software, in turn, needs to be able to detect when the web application or smart phone application are in use.

Ensuring the product is secured for the users is the first step in making the system secure against malicious attackers. Passwords need to stay censored in both the web browser controller and the phone application. PHP errors must be prevented, and when they cannot be, the errors must be inspected to ensure that no sensitive information about the database is revealed. Different errors was intentionally created and its respective errors inspected to ensure database

access is not exposed. A not-so-obvious test was to deactivate JavaScript and observe the behavior of the web page.

6.6 Testing Environment

Due to the Internet being a core component of the project, the testing environment of the project is not only flexible, but expansive. Nearly all the work of the project is stored in online servers. The web pages the team builds started off located on each group member's respective laptop, but testing them there is unacceptable. The most up to date files was put on the servers, and then tested from browsers of every type for quality and compatibility issues. The smart phone application, like the web app, was tested on smart phones. Daniel and Marvin each had their own Android smart phone, and downloaded the application onto their phones to test the behaviors on a real world setting. The only physically binding component to the project is the hardware simulating the theoretical third edition of the touch screen HVAC controller, which was used simply to check the hardware's ability to communicate with the web database.

Section 7: Conclusion

7.0 Future Additions and Considerations

The results of this project should be an exciting addition to existing work from the previous design team, but due to constraints in time and the scope of the project, the results are a far cry from a marketable product. Natural Air, at minimum, will have an excellent working prototype for controlling the next edition of the HVAC system hardware, but therein lays the first problem: the hardware. If there was time, Team 16 could attempt to transfer the touch screen HVAC software from the first team over to the new hardware. The new hardware Team 16 had was not intended to fully replace the work of our predecessors. In fact, this was not even in Proton Design Inc's contract. It is meant to simply simulate the environment that must be able to communicate with the web infrastructure produced. A future Senior Design team, Proton Design, or some other subcontractor can create solid hardware that can be installed onto a customer's home or office building and can connect to the web infrastructure. The second problem is much vaster and requires Natural Air working together with a web software oriented team on the merging of internet technology with Natural Air's entire business structure. To begin, Natural Air must work together with a developer and consider how to sell HVAC control units to consumers and how to provide customer account services.

One possible scenario is as follows: a customer buys a service package from Natural Air Inc. This service package includes the purchase and installation of an HVAC control system unit in their building. It is connected and attached to an air conditioning system. That control system is connected to a network for Natural

Air, and an account is set up linking the customer's HVAC system to the internet so it can be reached from a web browser or smart phone. Several things need to happen to make this possible. First, each hardware unit needs a unique serial number, and possibly account number, assigned to it. When a unit is installed in a home, a new customer account is created within a database, unique from other accounts based on this serial number. This permanently establishes a customer's account for ease of use and control. Next, Natural Air could potentially have an account page added to the browser controller to make payments and check upon the status of their unit, depending on the way Natural Air wants to charge customers for their service.

Upon the complete redesign of the hardware and web browser, aesthetic redesigns should be considered upon everything, including the HVAC touch screen interface, the web browser, the phone application, and even the current Natural Air website (www.naturalaireenergy.com.) The aesthetics of the current HVAC system's touch screen interface uses a lot of black, white and gold, as well as the University of Central Florida logo. Natural Air's website, on the other hand, uses white and green associated with Natural Air's commitment to helping consumers conserve energy and save money. Early in the fall semester, Team 16, and Daniel in particular, offered to design the web browser controller around Natural Air's current website, rather than the university. William Carson declined, arguing he greatly enjoyed the current aesthetic design of the HVAC touch screen interface. It certainly works, but Natural Air may want to consider a united branding of all hardware and software based on the company's future direction, and that includes something as basic as the color scheme of all software.

7.1 Future Additions

While this project main goal was to provide a cross-platforming interface throughout the Internet and mobile devices, there are other ideas that can warrant future development. These additions are not in the final design; due to the extra time needed to develop them were likely outside the scope of these two semesters, on top of the rest of the project. Nevertheless, these ideas are something to look into in future versions.

7.1.1 Self-Updating Firmware

One unique feature that could be included into the current project is to have the interfaces allow self-updating firmware. Self-updated firmware is a technology that continuously monitors changes in the data. If there is a change, a signal will be sent to update the device with the new data. This technology will be applied mostly on the max/min power energy and savings sliders. If a change is detected

within the data connected with the power sliders, it is up to the firmware technology to update this change to the entire system.

7.1.2 Weather Monitoring Service

Another feature that could have potential in the future is to include a weather monitoring service to the interface of the HVAC system. This way, consumers can have access to the weather near their homes, and be able to adjust their heating and cooling settings accordingly using the system. It can also be possible to set the system so that it will always change settings to a certain position according to a specific weather pattern. The system will obtain the information about the weather either from outdoor sensors or from the Internet using the server database. These features will obviously extend to the web site page on the Internet as well as the mobile phone application.

7.1.3 Distributing the Phone App

An important step in the mobile phone development phase for the future is to determine an effective way to distribute the application for users. There are several ways to go about this. While the usual approach for mobile phone applications is to directly sell them in a phone network's app store, it would seem somewhat unreasonable to ask users to pay extra for a mobile version of the same website they can already access. Many users would simply access the main website on their phones instead, which in turn defeats the purpose of having a mobile phone version of the system website. One potential alternative to this problem is to distribute the phone application for free. The only issue with this is how to make it available for users easily. One way is to make it possible on the main website to download the mobile phone app, by way of a link that allows users to receive the app for their phone. Since only users who have the HVAC system installed in their homes would be accessing the site, there is little reason for users to have to obtain the app from an outside source. People who don't have a HVAC system have no reason to have the application available from an app store. Additional research will be needed to discover the particulars of this method, such as whether a user must be on the phone to install the app. Regardless; this idea will be very beneficial to the future of this project. The total future additions can be summarized below.

- Self-Updating Firmware
- Weather Monitoring Service
- Web based account creation
- Web based billing and customer services
- Software art style and aesthetics based on business direction

- Serialization of all hardware units and accounts

7.2 Concluding Comments

The software we produced for Natural Air's Efficient HVAC System allows the future edition of the HVAC hardware to communicate to the Internet. This software has an entire system of infrastructure needed so the next edition of the HVAC system will have everything needed to provide three new ways to control the HVAC system. This entire system was composed of four parts which Team 16 worked on from December 2011 to April 2012.

The first part of the project was the core of making the entire project possible. The MySQL database was created on an online server and stores all the variables that the current HVAC system produces. This database initially stores one "account" containing all the necessary values that the previous HVAC touch screen design had. It also contains some initially arbitrary variables with "customer account information," which will later serve Natural Air when the database could theoretically store many more customer accounts. The database serves as the center from which all the other parts of the project will communicate through.

The website is both the second component of the project and a stepping stone to the final part. The website is able to communicate with the database from any computer with an Internet connection. It visibly mimics the Graphical User Interface of the original HVAC touch screen controller, as requested by Natural Air. Like the original, it has all the same functions, allowing a user to make changes to temperature and humidity as well as other settings. It does not communicate to the HVAC controller directly; it must use the database. It also has Administration capabilities, because Natural Air would like the ability to provide technical support to customers via their headquarters or a service truck in the event a customer needs assistance.

The third part of the project is the smart phone application, and the ultimate goal of the project. Natural Air wants this feature the most, in which a user would be able to access control to their HVAC system from any smart phone device. By producing the website first, technically a smart phone could be able to use it. Instead however, a second website optimized for smart phones was produced. A customer will be able to access their account the same way they would on the website produced, so they will be able to manipulate their HVAC controller with their phone. We tied all these components together with a simulated HVAC system programmed in C and developed based on the PIC32 microprocessor. This program was able to read and write to the database based on user inputs from the program, the browser controller, and the smart phone controller.

The deliverables to Natural Air are a unique and complex combination of software and continuity. The two website based HVAC controllers (one for

desktop browsers and the other for the mobile phone application) are given to Natural Air on a CD. This is important because Team 16 will not permanently be maintaining the websites indefinitely, nor will the web hosting service providers that reserve web space for the project. The project needs to be able to be moved from service to service, or between working environments, depending on the direction Natural Air wishes to take with the Team's work. All passwords and user accounts to such services, should they be intact after the project is delivered, will also be given to Natural Air, so that work can be continued if Natural Air wishes to keep the services that Team 16 would have purchased. The hardware that Team 16 performs experiments and development on will also be turned in directly to either Natural Air or Proton Design Inc (in the event Tim VanderMey will continue development on the project.) The simulated HVAC system was turned in to Natural Air on a separate CD from that of the websites, due to the nature of the hardware and software being so much different. On top of all these deliverables is a continuity report, intended to provide instructions to all these components. These deliverables can be summarized below.

- One CD containing web based software and access to all services purchased and remaining available for use
- One CD containing software coding related to the HVAC hardware
- One continuity report for future development.

One part of the project that cannot go to Natural Air is access to Team 16's Adobe Dreamweaver account. Team 16 has made arrangements with Dr. Richie and Workforce Central Florida to turn in the access code to this software. The account that Team 16 purchases will be intended for educational purposes, and therefore not only can't Natural Air not use them, but the software focused nature of the project means that rather than turn in hardware to the College of Engineering and Computer Science, we will turn in access to the software purchased, so that future Senior Design groups may take advantage of them to their fullest ability. This project is to assist Natural Air in producing a powerful and lucrative solution to the problem of energy consumption within America. Team 16 would like to thank Dr. Samuel Richie, William Carson and William Carson Jr. for their support and the opportunity to expand our engineering learning experience with this exciting project. We would also like to give a special thanks to Tim VanderMey of Proton Design Inc for his involvement and support in the initial design of the project, and helping to define our project objectives.

Appendix

Bibliography

- Kathy Sierra & Bert Bates, Head First Java, O'Reilly Media, Inc. Sebastopol, CA, 2003
- (Authors), Head First PHP and MySQL, O'Reilly Media, Inc. Sebastopol, CA, 2003
- (Authors), Head First jQuery, O'Reilly Media, Inc. Sebastopol, CA, 2003
- Christian Darie, Brinzarea Bogdan, Filip Chereches-Tosa, & Mihai Bucica, AJAX & PHP – Building Responsive Web Applications, Packt Publishing, Birmingham, UK, 2006
- Scott Duffy, How to Do Everything with JavaScript, The McGraw-Hill Companies, Berkley, CA, 2003
- Michael Cross, Developer's Guide to Web Application Security, Syngress Publishing, Rockland, MA, 2007
- Erik Schetina, Ken Green & Jacob Carlson, Internet Site Security, Pearson Education, Indianapolis, IN, 2002
- Merritt Maxim & David Pollino, Wireless Security, The McGraw-Hill Companies, Berkeley, CA, 2002
- Eric Meyer, Eric Meyer on CSS – Mastering the Language of Web Design, New Riders Publishing, Berkeley, CA, 2002
- Andrew S. Tanenbaum & David J. Wetherall, Computer Networks, Pearson Education, Boston MA, 2011
- Abraham Silberschatz, Peter B. Galvin & Greg Gagne, Operating System Concepts, John Wiley & Sons, 2011
- <http://yourhome.honeywell.com/home/Products/Thermostats/7-Day-Programmable/Prestige+HD+7-Day+Programmable+Comfort+System.htm>
- http://pandaboard.org/sites/default/files/board_reference/A1/Panda_Board_Spec_DOC-21010_REV0_6.pdf
- http://www.microchip.com/en_US/family/32bit/architecture.html
- <http://www.xenarc.com/product/706tsa.html>
- http://www.newhavendisplay.com/app_notes/SSD1963.pdf

Project Operation

Hardware Project Operation

Step 1: Connect the debugging USB to the corresponding place on the PIC32 starter kit and computer. You should see the light on the kit is on which means the PIC32 was power on properly.

Step 2: Open the source code for our design using MPLAB latest version.

Note: We use MPLAB IDE v8.83

Step 3: Run the program. You should see the successful building and start to run

Step 4: All the sensor value will be show on the computer screen

Step 5: Press Button 1 (Middle Button) on the PIC32 Microprocessor

Step 6: Refresh either Desktop or Mobile browser for up to date sensor values

Web Application Operation

These instructions apply to both the standard web browser and mobile browser applications.

Logging in and using the main screen.

1. If you wish to use a standard web browser to use the HVAC web application, go to <http://www.naturalaireenergy.com/hvac/> using a web browser. If you want to use the mobile version, go to <http://www.naturalaireenergy.com/m/> instead.
2. Click on the login field and enter “suser” (short for Sample User)
3. Enter the password “hvac” and click submit.
4. A correct entry take you to the main control screen. Sample temperature and humidity values will be in place regardless of the hardware being on or off (to make the values of indoor and outdoor temperatures work, see instructions on hardware operation.)
5. Enter the mode of the HVAC system you want. (Cool, Heat, or Off.)
6. Enter the Temperature and Humidity values desired in their respective fields.

7. Select one of five Tolerance levels from the Emax drop down list.
8. Select the amount of ventilation (Fresh Air Exchange) desired from the list.
9. Select the duration of Air Quality Duration from the Air Quality List.
10. Select the Mood Scenets to be turned on or off.
11. Clicking Submit sends the user selections to the database. (If using the Mobile version, the page will not update. Instead, wait for a message saying the submission was completed.)

Using the scheduler

1. Click on the Scheduler tab at the top of the menu selection for the web browser or mobile browser application.
2. If using the Mobile version, the previously entered changes to the scheduler will be seen first. If using the standard browser, these entries will appear after the selection options.
3. To add a change, select one of the seven days of the week from the selection list, as well as a time in the Start time selection menu.
4. Choose the mode you want to activate at the selected time (you can choose from Cool, Heat and Off, similar to the main screen.)
5. Choose the temperature and humidity desired at the start time.
6. Choose the duration you want this change to last.
7. Choose the duration you want for Fresh Air ventilation and Air Quality filtration that you want.
8. Click the submit button for either browser and wait for the page to refresh. The table of previous entries will update.
9. To delete changes, select the radio button under the Delete column under the list and click the submit button. Wait for the page to refresh on either browser.

Using the Settings Page

1. Click the Settings page on either browser

2. The Degrees of changes under the five tolerance levels **cannot** be changed by a standard user.
3. Enter a Zip Code and State and click Submit to update your location.

Using the Admin Page

1. Click on the Admin page on either browser.
2. There is only one sample administrator. Enter the username as “dgalzarza” and the password as “airpower.”
3. The Standard selection radio buttons indicate hardware the sample user has installed. AC1 and AC2 are potential independent air conditioning pumps. “Fan” represents a ventilation fan. “NA” represents Natural Air. “DH” represents a dehumidifier. Make changes to these settings as desired.
4. Under “Natural Air” is the option for a Hepa filter. Turn this component on or off as desired.
5. Under “Mood Scents” are the options for Mood Scents devices. Turn the three values on or off as desired.
6. Under Heat/Cool are the individual adjustments for the behavior of the five tolerance levels on the Main Page. Each level can be adjusted by five degrees, as well as the amount of time the system will react to crossed temperature thresholds. Make changes to these levels as desired.
7. Under Humidity are the individual adjustments for the same five tolerance levels, except these entries make changes to the behavior of dehumidifier controls. Each level can be adjusted by ten degrees, with the same time values as temperature control. Make changes to these levels as desired.
8. Click Submit to store these changes to the database. (On the mobile browser, instead of waiting for the page to refresh, wait for a popup message saying the submission was completed.)

Copy Permissions

#1: References on system security, courtesy of Dr. Mark Llewellyn

Hi Daniel!

You certainly have my permission to do so. The textbook we used for the course was: Operating System Concepts, Essentials, Silberschatz, Galvin, Gagne, 2011, John Wiley & Sons, ISBN: 978-0-470-88920-6.

Regards,

Mark

----- Original Message -----

From: "Daniel Galarza" <danielgalarza767@msn.com>

To: "Mark Llewellyn" <markl@cs.ucf.edu>

Sent: Saturday, November 26, 2011 11:32:13 AM

Subject: Permission to reference notes

My senior design project requires a little bit of web security, and since I'm going into the field of National Security when I graduate, I would like to refer to your notes from Operating Systems, which I kept. In particular, the notes on Security would be used.

May I have permission to use them, as well as the name of the textbook we used from that class? It would be a tremendous help. Thank you.

V/R

//SIGNED//

DANIEL GALARZA, C/Capt, AFROTC

Professional Education Officer

#2: Images from Microchip Technologies, permission from Public Relations

I am Daniel Galarza, military engineering student in the University of Central Florida, and for my team's senior design project, we are working with a PIC32 microprocessor. May my team have permission to reprint specific images as part of our research paper? All appropriate accreditation and thanks will go to Microchip in writing on the paper of course.

Such pictures are:

The TCP/IP stack diagram:

The following images in the user guide for the PIC32 USB

- * The processor diagram on page 12/13
- * The block diagram on page 39 (the appendix)
- * The pin arrangement on page 42

If you have any questions you can email me at this email, or if necessary, call me at [407-435-9175](tel:407-435-9175).

--

Hi Daniel,

Per the following Copyright Usage page on Microchip's Web site, I don't believe you need to gain our authorization for this educational usage:

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=487¶m=en023282

I'm referring specifically to the following section of the page:

Educational and Non-Profit Use of Copyrighted Material: If you use Microchip copyrighted material solely for educational (non-profit) purposes falling under the "fair use" exception of the U.S. Copyright Act of 1976 then you do not need Microchip's written permission. For example, Microchip's permission is not required when using copyrighted material in: (1) an academic report, thesis, or dissertation; (2) classroom handouts or textbook; or (3) a presentation or article that is solely educational in nature (e.g., technical article published in a magazine). Please note that offering Microchip copyrighted material at a trade show or industry conference for the purpose of promoting product sales does require Microchip's permission.

Regards,

Eric Lawson, Public Relations Manager

Microchip Technology Inc.

#3: Permissions from Honeycutt

To whom it may concern,

I am an undergraduate electrical engineering senior at University of Central Florida. I am in the Senior Design Course and our group is developing HVAC system. I would like to ask for permission to use the information and diagrams from the datasheet in our design report.

Thanks for your assistance.

-Chung Chien Hsia
University of Central Florida

Chung,

Thank you for contacting Honeywell.

We would request you to contact Honeywell Corporate Office at :-

Morristown Corporate

101 Columbia Road

Morristown, NJ, 07962-1057

Phone No.: [973-455-2000](tel:973-455-2000)

Honeywell ECC Customer Care

Suggestions and recommendations made by the Honeywell Customer Care are provided, without charge, as a complimentary service and are therefore given on the express understanding and agreement of the recipient that they are given without warranty (express or implied) or liability of any kind. It is the responsibility of the recipient of this message to verify that a Honeywell Customer Care Center recommendation or replacement suggestion is suitable for installation or implementation. This may require referring to product installation instructions, contacting the original equipment manufacturer (OEM), or contacting a qualified service person or taking other appropriate steps. Final responsibility for the selection and application of a control or procedure rests solely with the installer.