CSCE 4925: Project Aero
Final Notebook

By: Alyssa Thurston, Breuna Riggins, James Sabetti, Travis Goral
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 5 May 2018

Table of Contents

What Happened?

As you might have noticed, this project did not reach full fruition. There are multiple reasons for this and this document is meant to outline all of these reasons.

First, we lacked the knowledge needed for a web based front-end, database design, and the middle-man between the two. A large portion of the code that we wrote required us to do extensive research. We had to learn what to code and what each line of code did. Put simply, we had to learn how to code from the ground up. Everything we did we learned on the fly.

We spent a large amount of time working on a Ubuntu based server and kept experiencing issue after issue. It was only after experiencing a large number of major roadblocks that we finally decided to jump ship and move on to a Windows based server. After doing that, the sailing was much smoother.

Aside from the setup of the infrastructure and general lack of coding knowledge, we also experienced many issues in the coding process. We would find many tutorials on how to do something and none of them ever worked the way they should. One such issue was setting up the Google Charts. Between three group members, well over forty hours were poured into trying to link the data in our database to the charts. Unfortunately, examples we found online didn't work either. That was a big piece of what our project was missing. Due to the sheer number of issues we experienced, lack of knowledge, and time spent trying to fix these issues, we weren't left with enough time to even get our hands dirty on much else.

Our initial plan was to divide and conquer, and the two team members in charge of doing the front end had no idea how to build one. One member had a decent amount of experience and threw one together eventually, and it was only then that things on the front end started to get done. The reason that this didn't happen sooner was because this particular member was preoccupied working on other things for the project, specifically the node.

Because we had to take up so much time learning and researching coding aspects of our project, we were only minimally successful in getting this project implemented. Had we started the formal development in the Fall Semester, and been where we are now at the end of the Fall Semester, this project might have stood a better chance of being brought to fruition. However, we were told not to start any coding until Winter Break, which was advice that we unfortunately followed.

As the project manager, I do have to say that I am proud of where I and my teammates were able to get to with this project. Everything we did we had to learn on the fly, we went in with zero knowledge of how to use any aspect of any of the items that used in this project, and were able to create a cohesive looking front end, a database, a PHP script to run queries and output results into a JSON file, and a functioning air quality sensor. We also didn't have a lot of time to

work with to get the server setup and to actually implement the plan into code. In the end, the only things missing from this project, albeit big things, included an API to collect data from the sensors, an onboarding process for new sensors, a login portal for users to manage their contributed sensors, and the link between the charts and the database.

CSCE 4925: Project Aero
Arduino Guide

By: Alyssa Thurston
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 5 May 2018
Status: Final Draft

Table of Contents

Revisions

Below is a list of any revisions made to this document.

Date	Description of Change Made	Person Making Change
5/5/2018	Draft Created	Alyssa
5/5/2018	Draft Edited	Alyssa

1 Materials and Software Needs

This section will provide links on the things that we purchases, and the software needs to get an Arduino Sensor set-up.

1.1 Materials

Before we get down to the shopping list, there are a few essentials that you'll want to have. These include:

- A soldering station, including a soldering iron, cleaning materials, proper ventilation, and a pin-point solder iron tip
- A multimeter in the event you're unsure something is working the way it ought to.

The following items are items that we purchased to make our own sensor. This sensor measured only particulate matter, but you could purchase other sensors and follow similar instructions to achieve the same sort of results.

- Arduino Mega: https://www.amazon.com/gp/product/B0046AMGW0/ref=crt_ewc_title_gw_1?ie=UTF8&psc=1&smid=AM0JQO74J587C
- WiFi Breakout: https://www.adafruit.com/product/2999#tutorials
- PM Sensor: https://www.adafruit.com/product/3686#tutorials
- GPS Shield: https://www.adafruit.com/product/1272#tutorials
- GPS Antenna: https://www.adafruit.com/product/960#tutorials
- A printer USB cable (called a USB A to USB B cable): https://www.adafruit.com/product/62
- SMA to uFL Adapter to attach the GPS Antenna: https://www.adafruit.com/product/851#tutorials
- A hefty amount of jumper cables: https://www.adafruit.com/product/153
- Micro SD Card (not essential, but we did include one on our sensor)
- Breadboard: https://www.adafruit.com/product/64#tutorials
- Plastic Mounting Board to hold the Arduino and Breadboard: https://www.adafruit.com/product/275#tutorials
- Shield Stacking Headers: https://www.adafruit.com/product/85#tutorials

1.2 Software Needs

First and foremost, we recommend that you get a good text editor. There are many out there, we recommend Atom or Notepad++. You can download these programs from online.

Development for the Arduino is done on the Arduino's IDE. We used the most current version, version 1.8.5. For Windows, you can download that here: https://www.arduino.cc/download_handler.php?f=/arduino-1.8.5-windows.exe For Mac, you can download that here: https://www.arduino.cc/download_handler.php?f=/arduino-1.8.5-macosx.zip

Please note that the installation instructions that follow are designed for a Windows machine. If you have a Mac, they may differ slightly.

Run the file that you downloaded, there may be a security message that comes up and you can click "yes" on that. Next click "I agree." On this screen, make sure that all boxes are checked, and click "next." Lastly, click "install." When complete, simply click "close."

When the installation completes, you'll need to do some basic setup. First, go to "Tools" then "Board," and select Arduino Mega ADK. Another thing you'll need to take note of is the "Port." When you plug the Arduino in, you'll need to go to the port, and select the proper option. Usually, it will show something that says "COM# (Arduino/Genuino Mega or Mega 260)." If you get errors when uploading your code, verify that the proper port is selected.

Next we'll need to install a few libraries. To do this, go to "Sketch," then "Include Library," Then "Manage Libraries." From this window, type "wifi101" then hit enter. Click on the first one that comes up (the title should say "WiFi101" by "Arduino," we used version 0.15.2), and click "Install." Before you close this window, you'll need to install one more library. Search for "Adafruit_GPS" and install this library. We used version 1.0.3 for this.

Once you are done with this, it's time to begin building your node.

2 Building Your Node

The first thing you'll need to do is solder a header strip to the WiFi board. Never done this before? Don't fret, we had minimal soldering experience and it went well on our first try. Follow these links to get this step done:

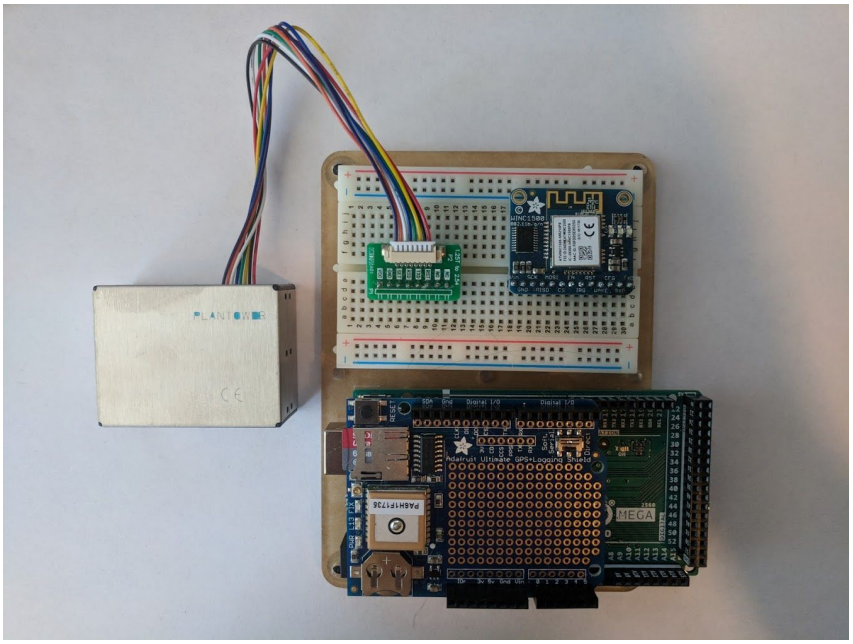
- Assembling the WiFi Breakout Board: https://learn.adafruit.com/adafruit-atwinc1500-wifi-module-breakout/assembly
- How to Solder: https://learn.adafruit.com/adafruit-guide-excellent-soldering

Next, you'll need to solder some parts of the GPS board. Follow this link to accomplish this: https://learn.adafruit.com/adafruit-ultimate-gps-logger-shield/shield-headers

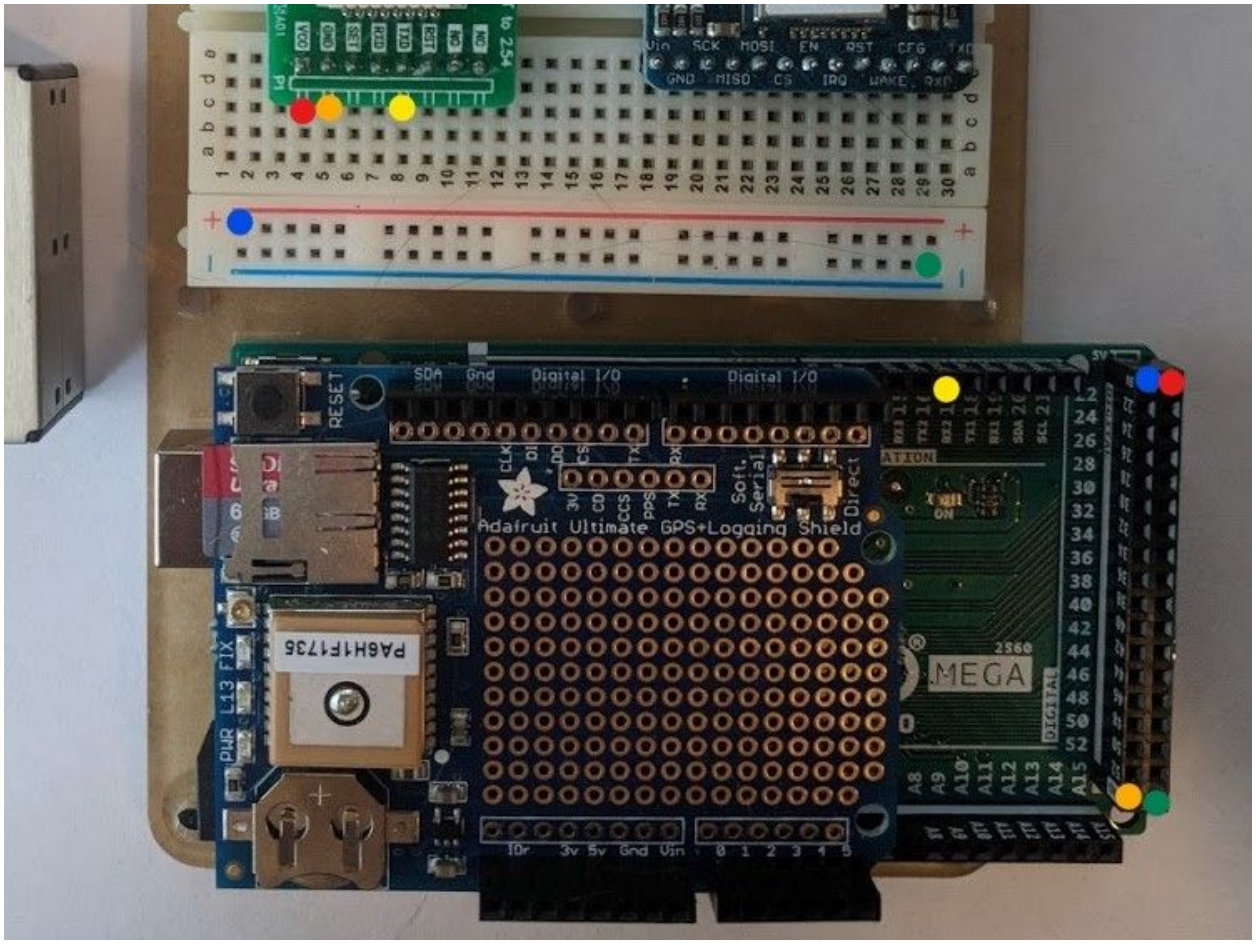
Lastly, you'll need to plug in the Particulate Matter Sensor (the box) to the breakout board using the included ribbon-wire.

With the soldering complete and wires connected, we can now begin setting up our Arduino. The first thing we recommend doing is assembling your mounting plate, if you chose to purchase one. It keeps the arduino in place and prevents it from sliding around.

First, you'll need to place the Particulate Matter Breakout and the Wifi Breakout boards to the breadboard. The Particulate Matter Board should go on the breadboard row titled "E," and the first pin (titled VCC) should go on column 4, the last pin (NC) should be on column 11. The WiFi board should also go along row "E," with the first pin on column 18 and the last on column 30. It should look something like this:



Next, we need to wire everything together, we'll take this one sensor at a time. First, let's start with the particulate matter sensor. Use the picture below as a guide to guide you on where to place the wires. Plug one end of the jumper cable to one color dot, and find the same color on the picture, and plug it into the appropriate place. A list is provided to guide you through any issues you might have following the picture.

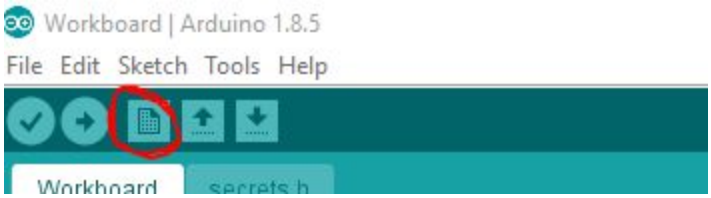


Wiring for the PM Sensor, 5 wires:

- Upper most Positive breadboard hole -> Arduino 5V (right above port 22)
- Bottom most Negative breadboard hole -> Arduino Ground (right below port 52)
- PM VCC (on the breadboard that would correlate to C, 4) -> Arduino 5V (right above port 23)
- PM GND (on the breadboard, that would correlate to C, 5) -> Arduino Ground (right below port 53)
- PM TXD (on the breadboard, that would correlate to C, 8) -> Arduino port number 17 (RX2).

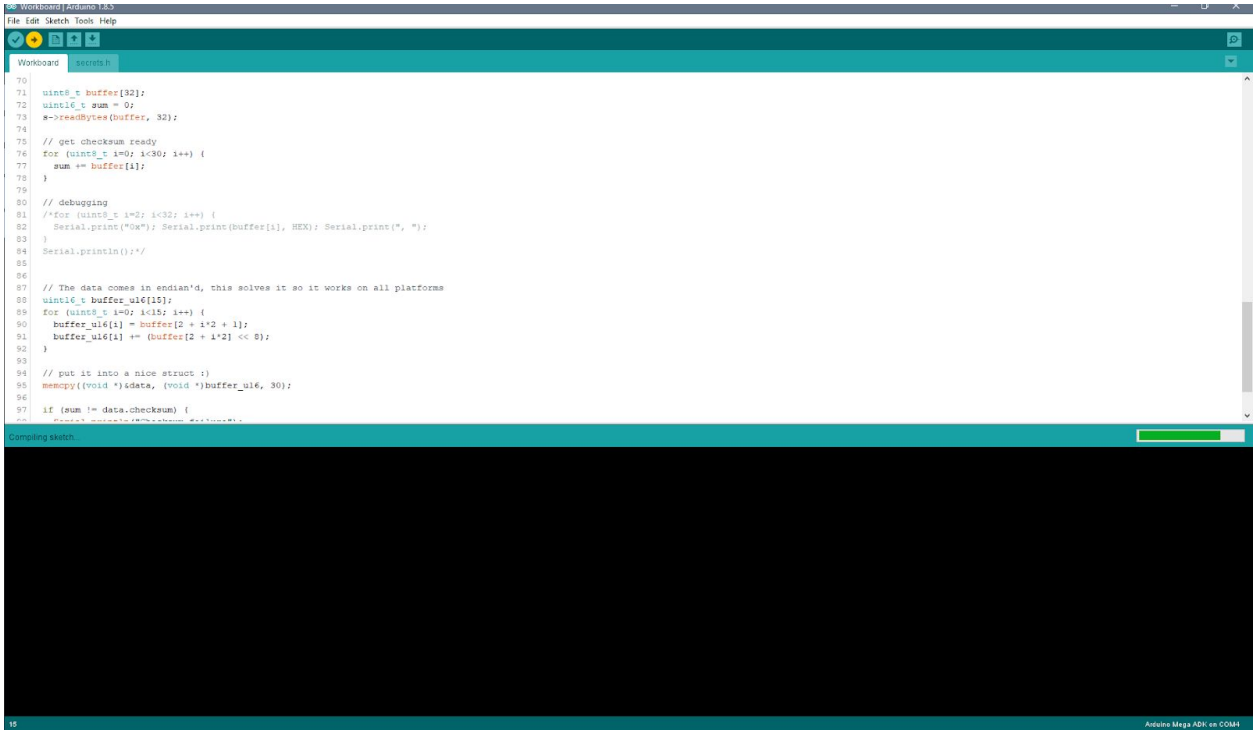
This should be all you need to get the PM Sensor running. To confirm, we need to run some code on the Arduino.

Plug in your Arduino to your computer using the USB A to USB B cable, and open up the Arduino IDE, and click the icon shown below to create a new file:

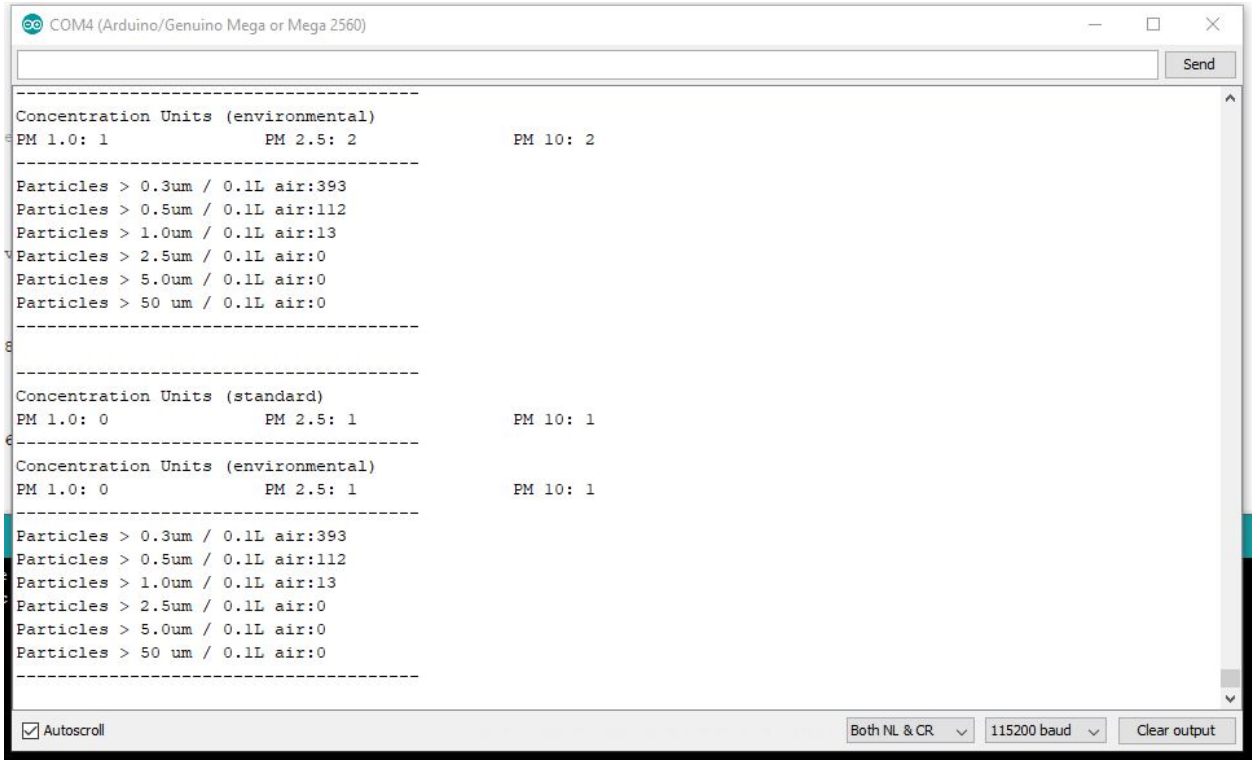


Once you create your new file, click “File” then “Save As” and name the file “Workboard” or something similar.

Copy and paste the code here to your Workboard file (https://github.com/Project-DAMN/Arduino-Code/blob/master/PM%20Sensor%20Test), then click the arrow to the left of the “new file” icon to upload the file to the Arduino. This may take a few moments. You can monitor the progress from the blue bar on the lower part of the screen. Here, it indicates that the sketch is compiling, the green bar shows the progress.

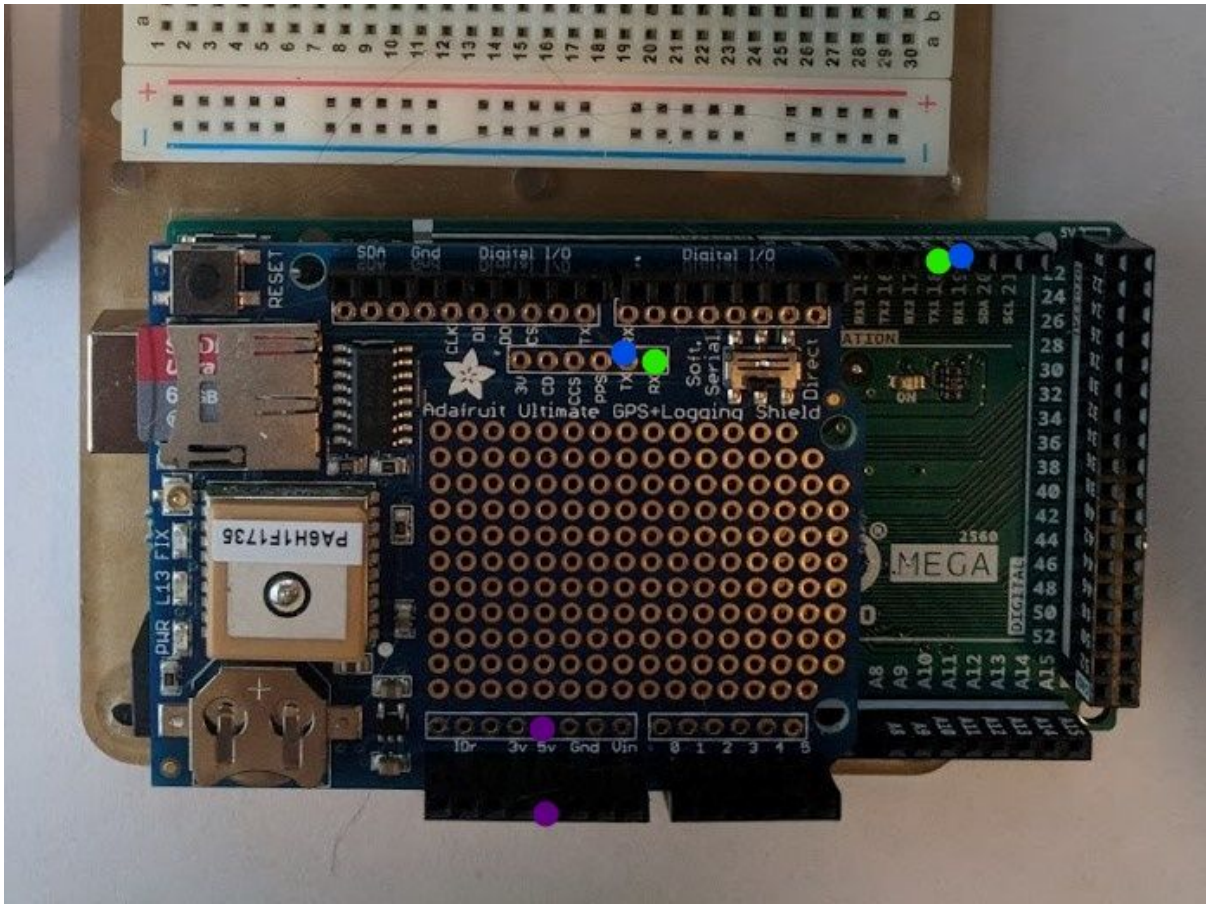


When the uploading process is complete, go to tools, then select “Serial Monitor.” Near the bottom, you’ll see something that says “9600 baud,” change this to “115200 baud.” You should see output similar to the following image, which indicates that this sensor is working as expected.



If you do not see numbers with the output, or if you see nothing at all, reread the instructions and make sure nothing was missed. For further help, see this website: https://learn.adafruit.com/pm25-air-quality-sensor/arduino-code

Continuing with our wiring, the next thing we will test is our GPS sensor. First unplug the Arduino from USB, then wire up the following, in addition to what was wired in the previous step, do not unplug anything:

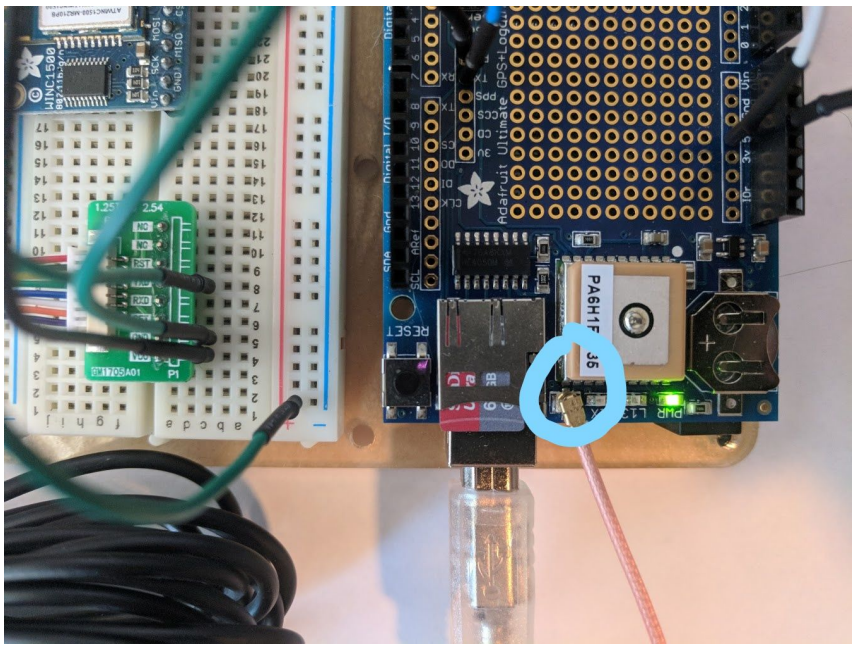


Wiring for the GPS Sensor, 3 Wires:

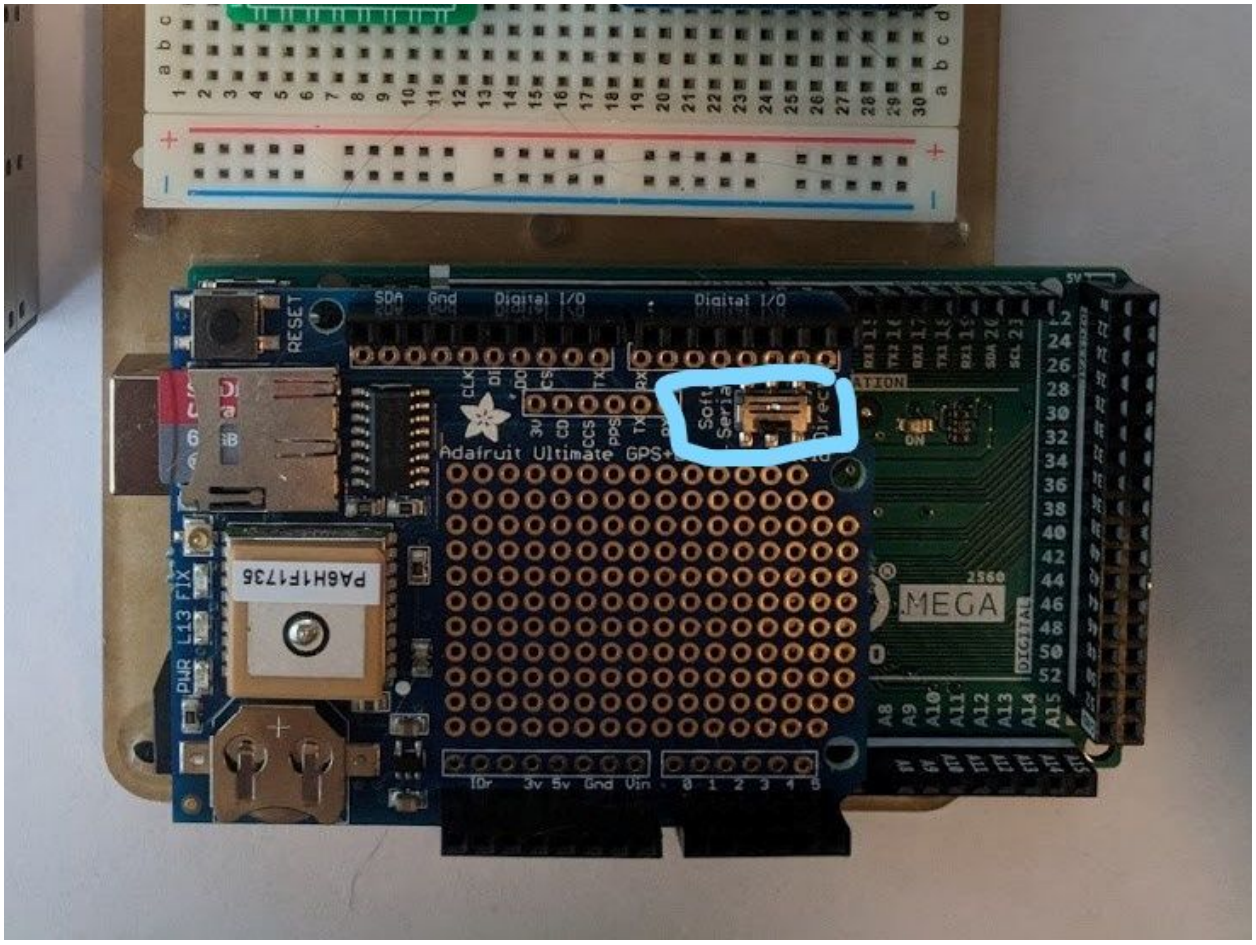
- 5V on the GPS Hat -> 5V on the Arduino. If you look at the side, underneath the GPS hat, you should be able to follow the pins to find the right hole. Otherwise, it should be the 5th hole from the left, if the unit is positioned the same way as shown on the above image.
- TX on the GPS Hat -> Arduino Port 19 (RX1). This can be confusing, but it is the lower most TX Port if you're looking at the arduino as it is oriented in the above image.
- RX on the GPS Hat -> Arduino Port 18 (TX1). This can be confusing, but it is the lower most TX Port if you're looking at the arduino as it is oriented in the above image.

Next, we should test the GPS Sensor. To do this, first attach the SMA to uFL Adapter to the GPS Antenna. It will look something like this.

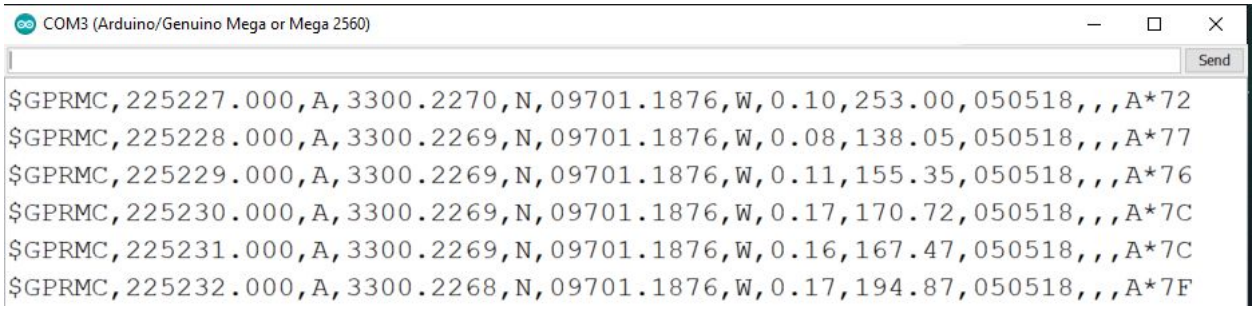
Then, attach the GPS Antenna to the GPS Shield. To do this, simply snap the end of the adapter onto the shield, it will look something like this:



Next, copy the code from here (https://github.com/Project-DAMN/Arduino-Code/blob/master/GPS%20Sensor%20Test1) and replace with the code that we sent to the Arduino in the previous step. There is a switch on the GPS Sensor where it says soft serial, and direct. Flip the switch to direct.



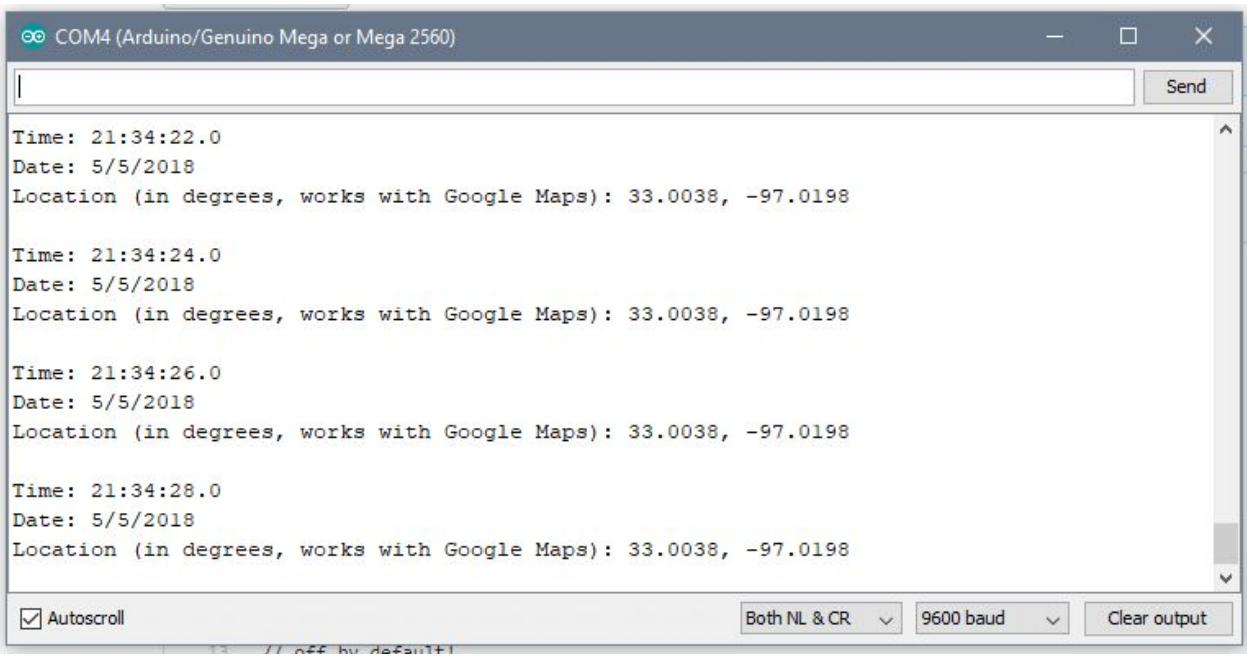
Upload the code. If you don't see anything at first, make sure that you switch the baud from 115200 to 9600. The code should look something like this:



As our next test, we're going to run some more code on the GPS Sensor. Flip the switch from direct back to soft serial, and copy the following code, replacing what was previously in the

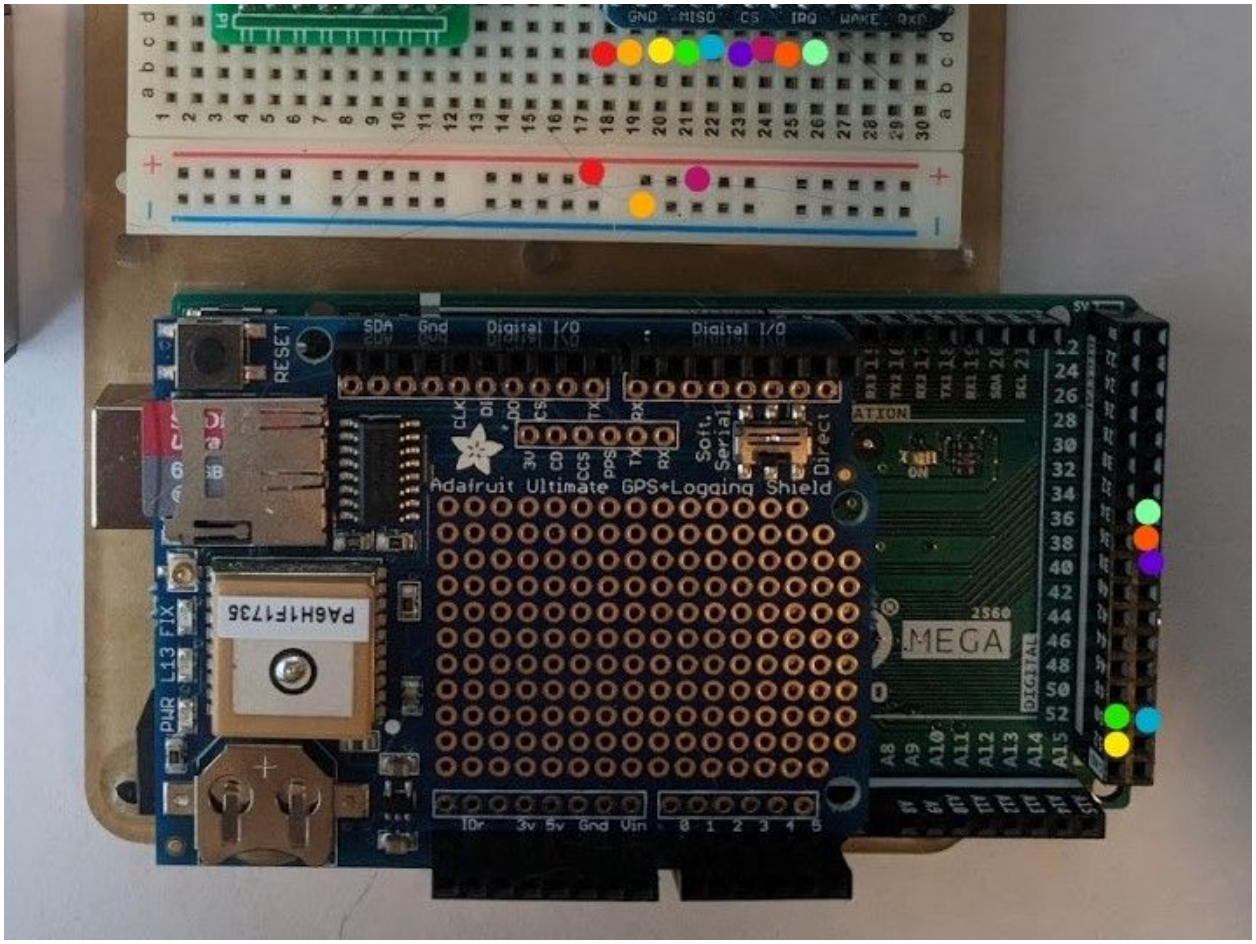
arduino IDE, and upload this to your board:
https://github.com/Project-DAMN/Arduino-Code/blob/master/GPS%20Sensor%20Test2

This should produce output similar to the following:



If you do not see numbers with the output, or if you see nothing at all, reread the instructions and make sure nothing was missed. For further help, see this website:
https://learn.adafruit.com/adafruit-ultimate-gps-logger-shield/overview

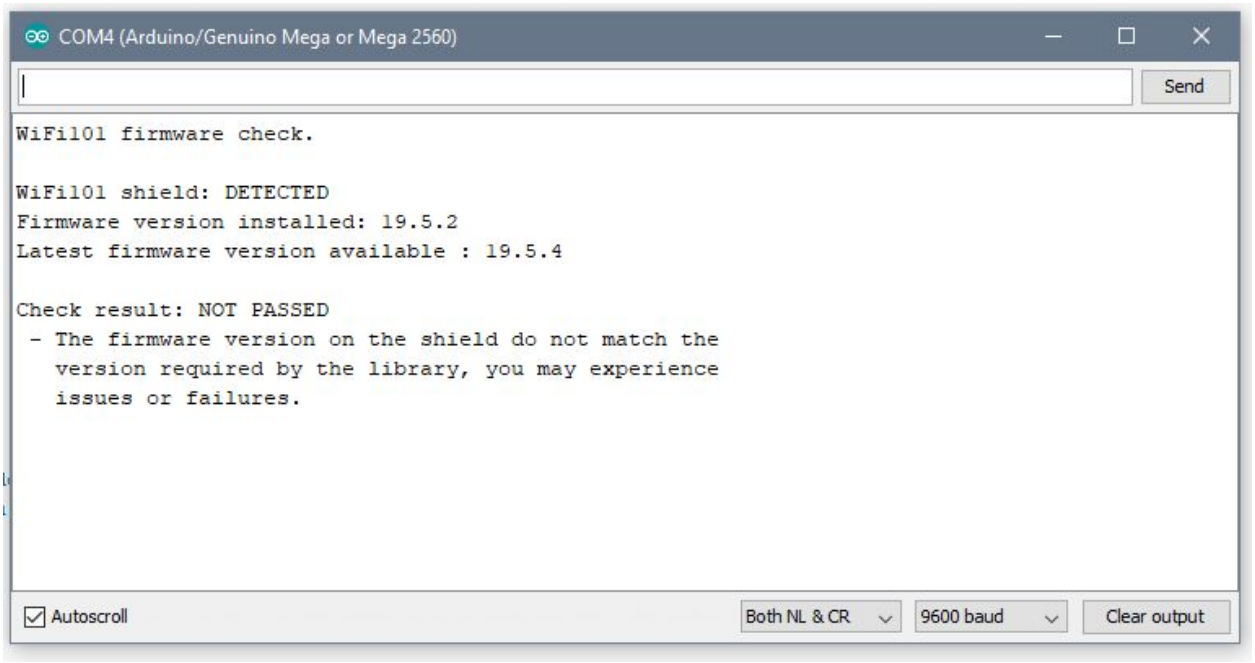
The last sensor that we need to connect is the WiFi Sensor. Same as before, don't disconnect anything, I'll show a picture of everything, and list out the connections following that.



Wiring for the WiFi Breakout Board, 9 wires:

- Vin (C, 18 on the Breadboard) -> Positive area on the breadboard
- GND (C, 19 on the Breadboard) -> Negative area on the breadboard
- SCK (C, 20 on the Breadboard) -> Port 52 on the Arduino
- MISO (C, 21 on the Breadboard) -> Port 50 on the Arduino
- MOSI (C, 22 on the Breadboard) -> Port 51 on the Arduino
- CS (C, 23 on the Breadboard) -> Port 39 on the Arduino
- EN (C, 24 on the Breadboard) -> Positive area on the breadboard
- IRQ (C, 25 on the Breadboard) -> Port 37 on the Arduino
- RST (C, 26 on the Breadboard) -> Port 35 on the Arduino

Our test code for this shield can be copied from here:
https://github.com/Project-DAMN/Arduino-Code/blob/master/WiFi%20Test
Copy the code from there, plug your Arduino in, replace the code that was in the IDE with the code you copied, upload it, and verify that the output in your Serial Monitor looks something similar to the following, if it does, keep moving along. Ignore the message about “NOT PASSING”:



If you do not see this output, or if you see nothing at all, reread the instructions and make sure nothing was missed. For further help, see this website, noting that the type of Arduino and breadboard used are different:
https://learn.adafruit.com/adafruit-atwinc1500-wifi-module-breakout/wiring-and-test

Now that you have your Arduino assembled and you have tested each sensor to verify functionality, it is time to upload the final code to the Arduino and run it. First, copy and replace the code in the IDE with the code found here, then save your file. Don't run it just yet, or you'll get some scary looking errors. Just save the file, then close out of the Arduino IDE for now.
https://github.com/Project-DAMN/Arduino-Code/blob/master/Final%20Code

Before we upload the file, there is one last thing we need to do. Near the top of the Arduino IDE, you'll see a little triangle pointing down. Click that, and then click "new tab." Type the name "secrets.h" and click OK. Lastly, add the following text to the file:

```
1 #define SSID ""
2 #define PASS ""
```

Enter in the name of your WiFi network between the quotes on line one, and the password to your network on line two.

First open "My Computer" (or more formally known as "File Explorer"). You can also do this by hitting the Windows Key and Letter E at the same time on your keyboard. Then type C:\Program Files (x86)\Arduino\hardware\arduino\avr\libraries\SoftwareSerial\srcv into the Address bar.

Right click “SoftwareSerial.cpp, and click edit, either with Notepad or Notepad++. For those on a Mac, you’ll need to navigate to this same file.

Next, you’ll need to change a few lines of code. Add “/*” to line number 227, and “*/” to like 244, like so:

```
206  #endif
207  }
208
209  uint8_t SoftwareSerial::rx_pin_read()
210  {
211    return *_receivePortRegister & _receiveBitMask;
212  }
213
214  //
215  // Interrupt handling
216  //
217
218  /* static */
219  inline void SoftwareSerial::handle_interrupt()
220  {
221    if (active_object)
222    {
223      active_object->recv();
224    }
225  }
226
227  /*#if defined(PCINT0_vect)
228  ISR(PCINT0_vect)
229  {
230    SoftwareSerial::handle_interrupt();
231  }
232  #endif
233
234  #if defined(PCINT1_vect)
235  ISR(PCINT1_vect, ISR_ALIASOF(PCINT0_vect));
236  #endif
237
238  #if defined(PCINT2_vect)
239  ISR(PCINT2_vect, ISR_ALIASOF(PCINT0_vect));
240  #endif
241
242  #if defined(PCINT3_vect)
243  ISR(PCINT3_vect, ISR_ALIASOF(PCINT0_vect));
244  #endif*/
245
246  //
247  // Constructor
248  //
249  SoftwareSerial::SoftwareSerial(uint8_t receivePin, uint8_t transmitPin, bool inverse_logic /* = false */) :
250    _rx_delay_centering(0),
251    _rx_delay_intrabit(0),
252    _rx_delay_stopbit(0),
253    _tx_delay(0),
254    _buffer_overflow(false),
255    _inverse_logic(inverse_logic)
256  {
257    setTX(transmitPin);
258    setRX(receivePin);
259  }
260
261  //
262  // Destructor
263  //
264  SoftwareSerial::~SoftwareSerial()
265  {
266    end();
267  }
268
269  void SoftwareSerial::setTX(uint8_t tx)
270  {
271    // First write, then set output. If we do this the other way around,
272    // the pin would be output low for a short while before switching to
273    // output high. Now, it is input with pullup for a short while, which
274    // is fine. With inverse logic, either order is fine.
```

Save the file when you are done adding the appropriate information.

Now, click back on the tab that says “Workboard,” or whatever you named the file, and upload it to your Arduino and see if the output looks similar to this:

```
Connected.
Starting server connection.
Connected.

Concentration Units (environmental)
PM 1.0: 1               PM 2.5: 1               PM 10: 1

Concentration Units (environmental)
PM 1.0: 1               PM 2.5: 1               PM 10: 2

Concentration Units (environmental)
PM 1.0: 1               PM 2.5: 1               PM 10: 2

Concentration Units (environmental)
PM 1.0: 1               PM 2.5: 1               PM 10: 2

Concentration Units (environmental)
PM 1.0: 1               PM 2.5: 1               PM 10: 2

Concentration Units (environmental)
PM 1.0: 1               PM 2.5: 1               PM 10: 1

Time: 22:13:44.0
Date: 5/5/18
Location (in degrees, works with Google Maps): 33.0039, -97.0198
```

If you’re not seeing anything pull up, first be patient. In order for anything to show up, the node has to make a connection with your WiFi, if after about a minute or so, you still don’t see anything, verify that your WiFi connection information in Secrets.h is correct.

If it does, congrats! You have a working node.

3 Troubleshooting

This section will go through some basic troubleshooting that may be experienced with the nodes.

3.1.0 Code Won’t Upload

First, go to “Tools” then “Board,” and select Arduino Mega ADK. Another thing you’ll need to take note of is the “Port.” When you plug the Arduino in, you’ll need to go to the port, and select the proper option. Usually, it will show something that says “COM# (Arduino/Genuino Mega or Mega 260).” If you get errors when uploading your code, verify that the proper port is selected.

If the code still isn’t uploading, press upload again. In this instance, it’s okay to be a little impatient.

3.1.1 Nothing Shows Up on the Serial Monitor

This is usually caused by wiring issues or the baud rate not being set right. Verify both of these items, and also try unplugging the Arduino, and plugging it back in.

CSCE 4925: Project Aero
Server Guide

By: Travis Goral
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 5 May 2018
Status: Final Draft

Table of Contents

Revisions

Below is a list of any revisions made to this document.

Date	Description of Change Made	Person Making Change
5/5/2018	Draft Created	Travis
5/5/2018	Draft Edited	Travis

Installing Windows Server 2016

Installing Windows Server 2016 is a fairly straightforward process that is virtually identical to a regular Windows 10 installation. To start, insert your bootable media containing the Windows Server 2016 install files on it and boot from your media.

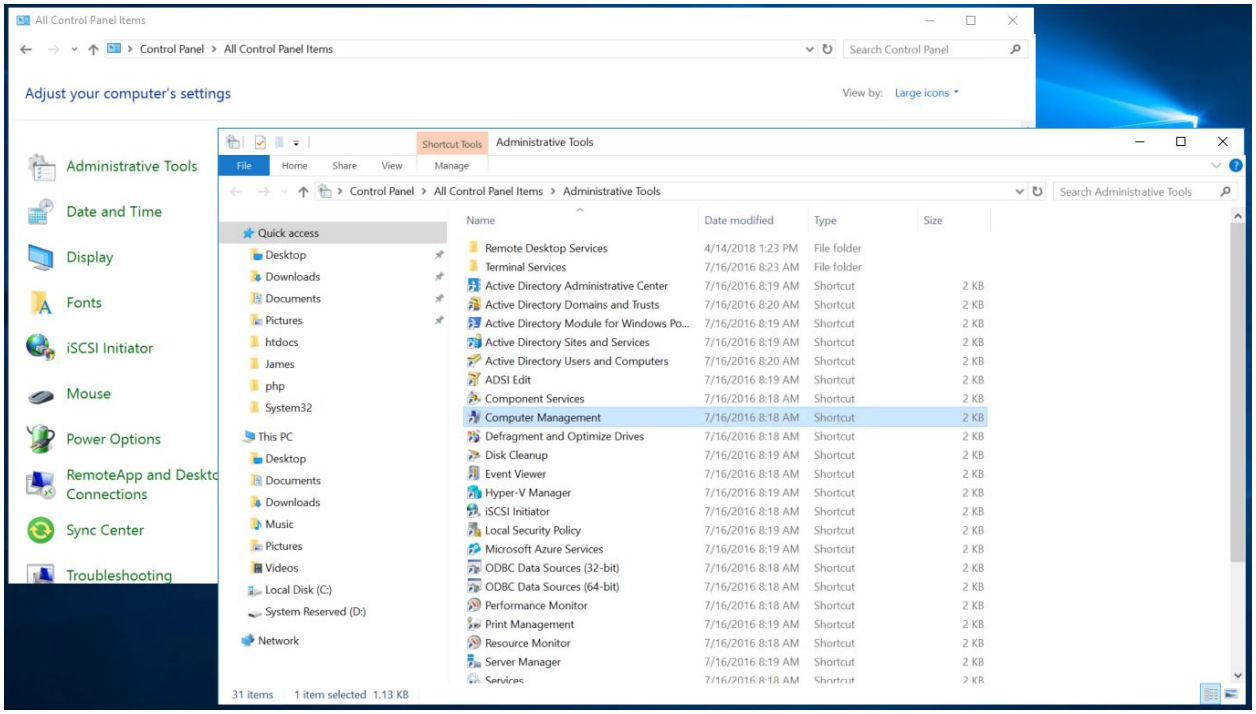
- The first screen you will see is titled “Windows Setup”. Select your language and region format, and keyboard layout and click Next.
- Click Install Now
- You will be prompted to enter your license key on the next screen
- Click Next and choose the default install setting
- Choose the drive that you wish to install on and format / partition it as desired
- Windows should now install and restart when finished
- When the restart is complete, you will be prompted to enter an Administrator password. Once you do this, installation will finish and you can login with the Administrator account.

Installing drivers

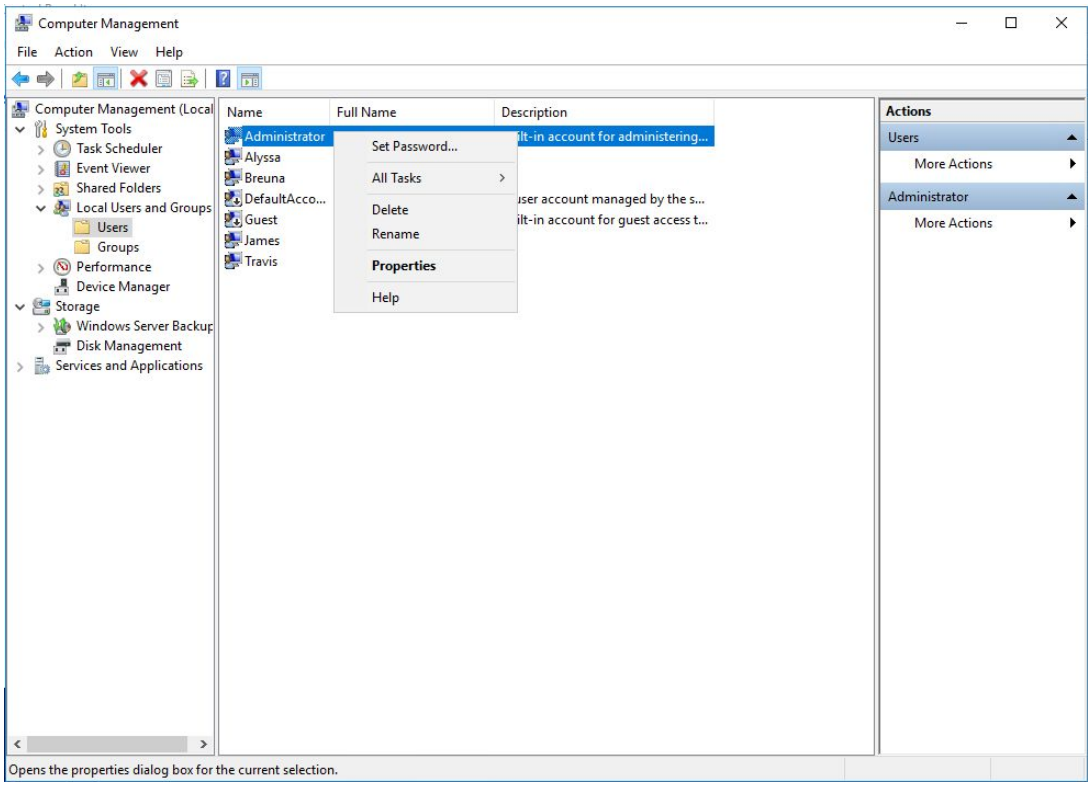
Some drivers such as those for ethernet adapters or your CPU may not come installed with Windows Server 2016. If this is the case, please refer to your chip manufacturer’s or device support page for driver downloads. For example, Intel has a list of supported drivers on Windows Server 2016 that can be found at: https://www.intel.com/content/www/us/en/support/articles/000022599/network-and-i-o/ethernet-products.html

Setting up the Administrator account

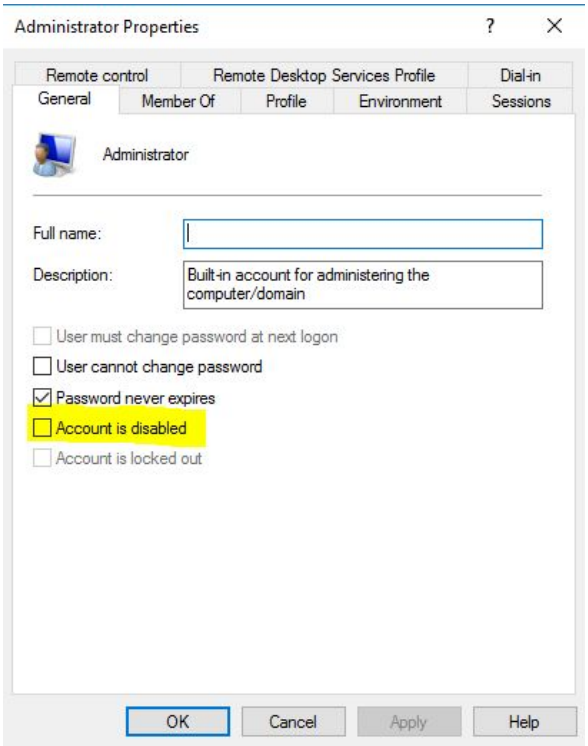
To setup an administrator account, open the Control Panel and select Administrative Tools, then Computer Management.



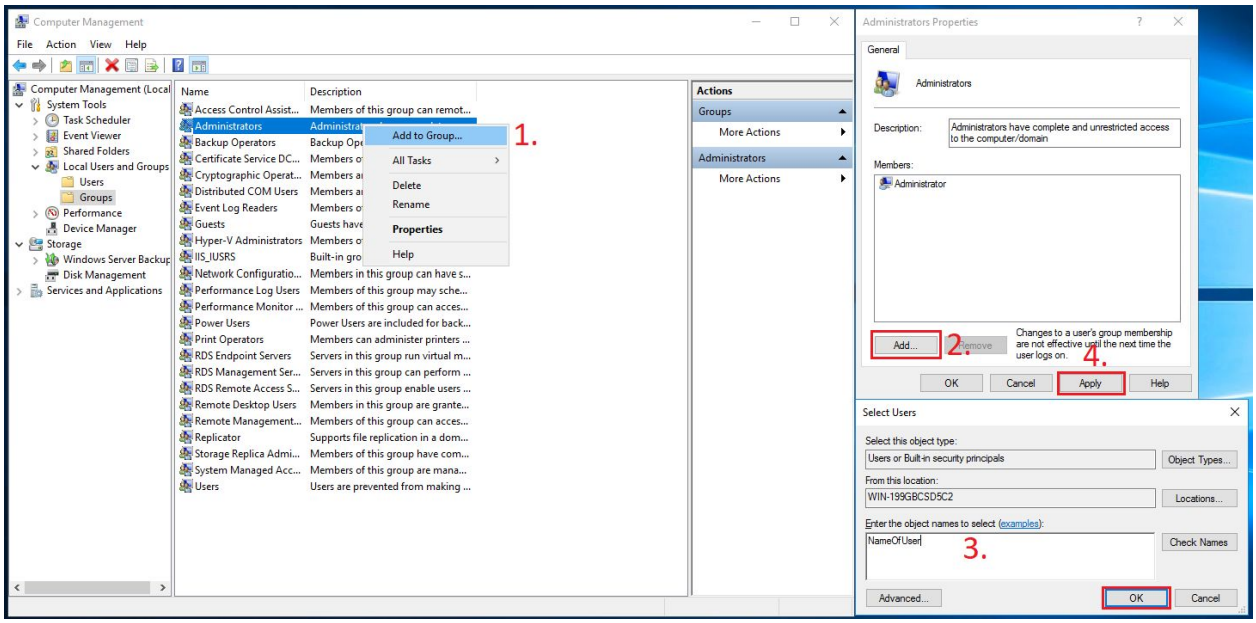
From Computer Management, you can modify users, set passwords, and manage groups among other things. To enable the Administrator account, go to Local Users and Groups > Users, and right-click on the Administrator profile and select Properties. You may also select Set Password… if you wish to change the Administrator password.



By default, the Administrator account is disabled. Make sure to uncheck this.

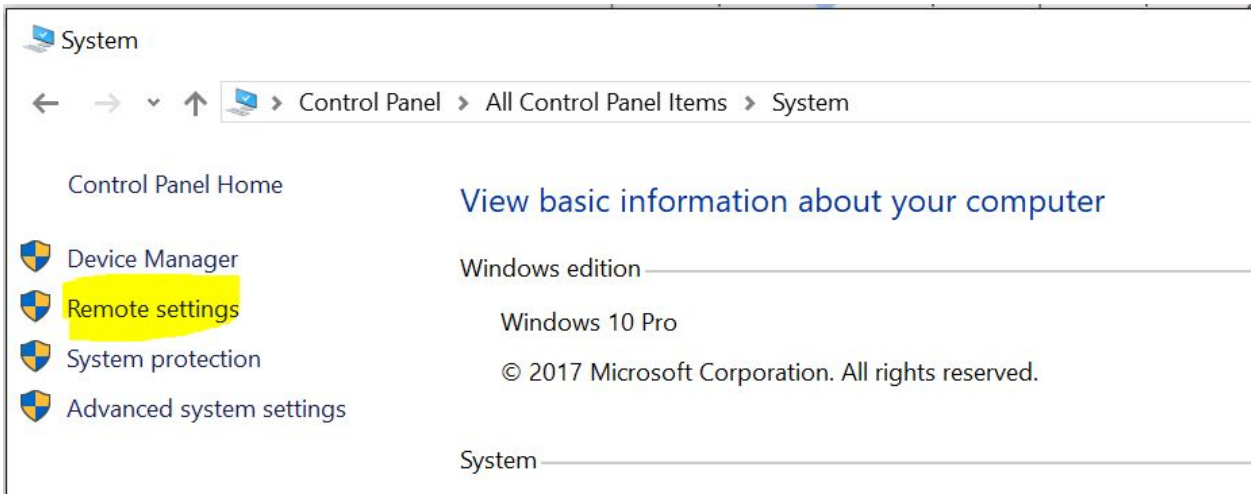


You can also give other users Administrator privileges. To do this, go to the Groups list and right-click on the Administrators group and click Add to Group. Next, click Add and enter the name of the user you wish to give admin rights into the box at the bottom. Finally, click Okay and Apply to apply these settings.

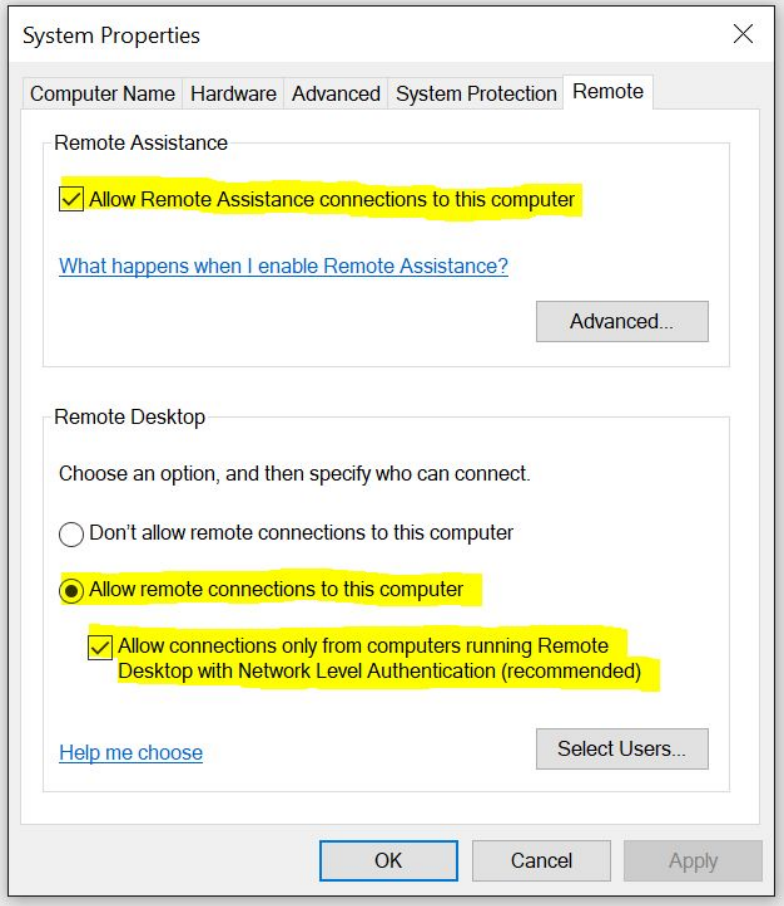


Setting up Remote Desktop

If Remote Desktop is not already configured on both the Server and Client computers, it will need to be enabled in order to connect to the server from a remote client. To enable Remote Desktop on Windows, go to the Control Panel, select Large icons under the View by: menu, then select System. From here, select Remote settings from the top-left.



Make sure all the following options are selected on both the client and server computers so that you can use Remote Desktop connections:



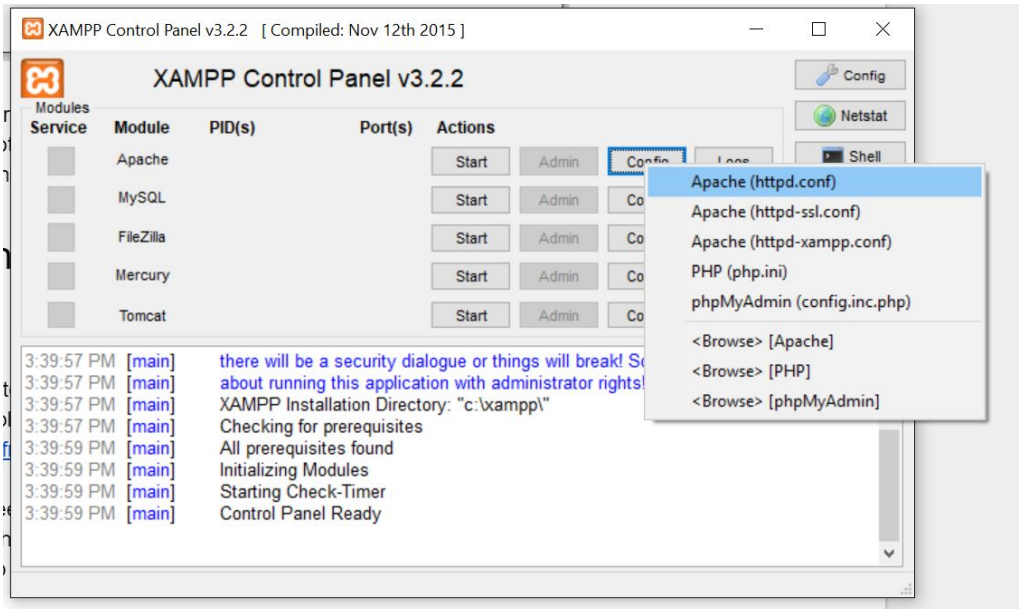
Depending on your router settings, you may also need to allow the default RDP port 3389 in order to allow Remote Desktop connections. If problems still persist, check your firewall settings for any deny rules that would forbid connections on port 3389.

Software Installation

XAMPP

XAMPP is required to host the website and process queries from the database. To install XAMPP, open the following link and select the Windows installation: https://www.apachefriends.org/index.html

Once the file has been downloaded, simply follow the standard installation instructions. If necessary, edit the httpd.conf file and change the root directory for XAMPP to the folder containing your web documents like index.html



You may also need to uncomment (remove the semicolon at the beginning of the line) or add the following lines in the php.ini file in the “Windows Extensions” section to enable Apache to work with PostgreSQL:

extension=php_pdo_pgsql.dll
extension=php_pgsql.dll
extension=php_pdo_sqlite.dll
extension=php_pgsql.dll

PostgreSQL

To install PostgreSQL, visit their download page for Windows:
https://www.postgresql.org/download/windows/

Once downloaded, follow the default setup. You will be asked to install a specific version of pgAdmin. I highly recommend version 3 instead of 4 since it has the most features and I feel is the easiest to use.

Firefox

It is highly recommended that an additional browser such as Firefox or Chrome be installed for ease of use and compatibility. https://www.mozilla.org/en-US/firefox/new/

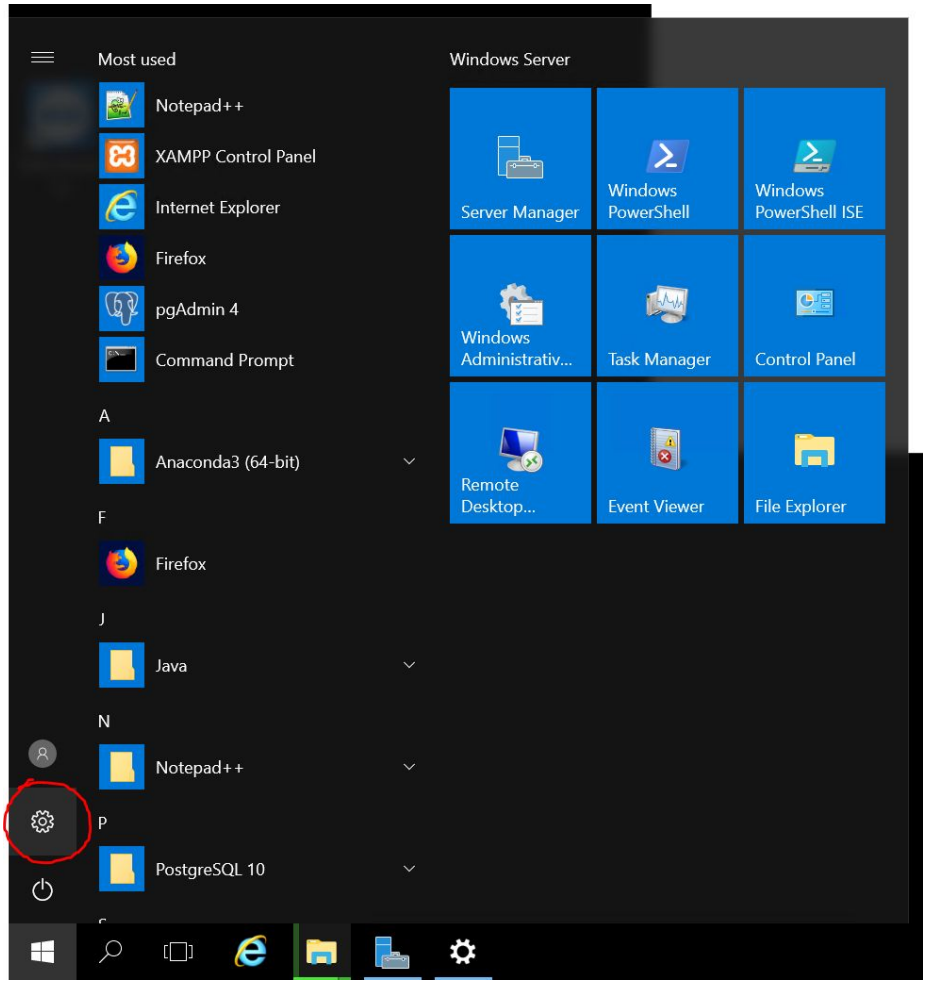
Notepad++

Optionally, you may find it helpful to install some kind of source code editor if you plan to edit any code within the server. Notepad++ is a good choice for simple code editing.
https://notepad-plus-plus.org/download/v7.5.6.html

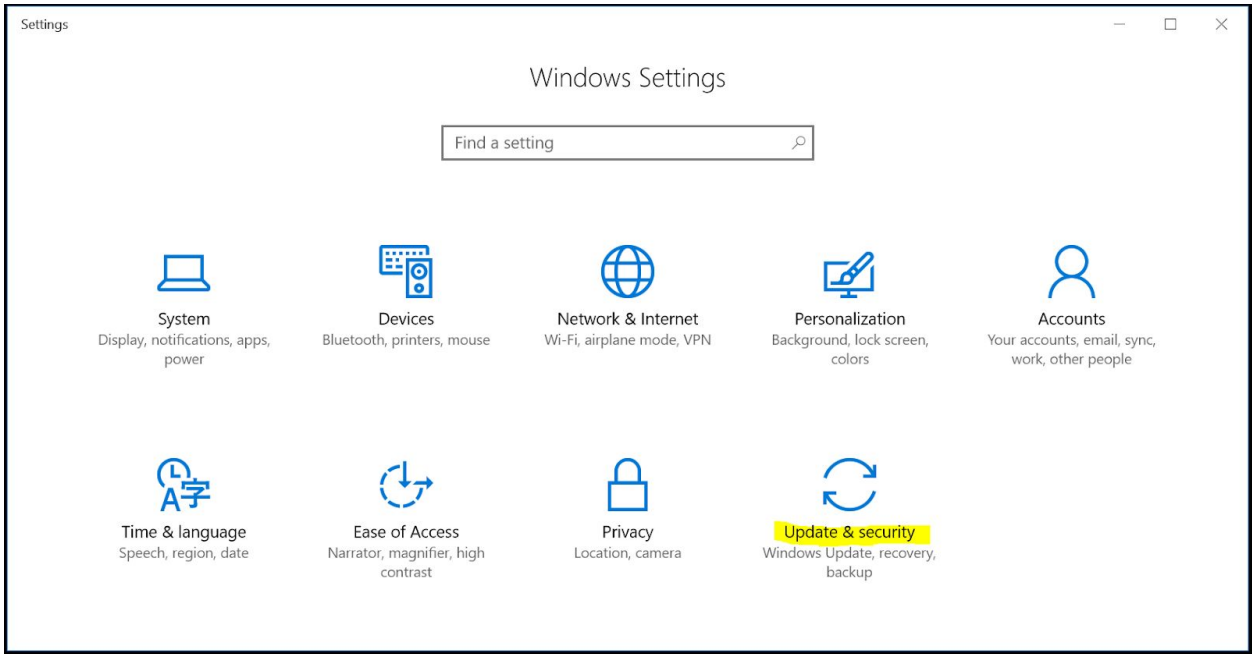
Routine Maintenance

Installing updates

To install updates, click the Windows (Start) icon and go into Settings



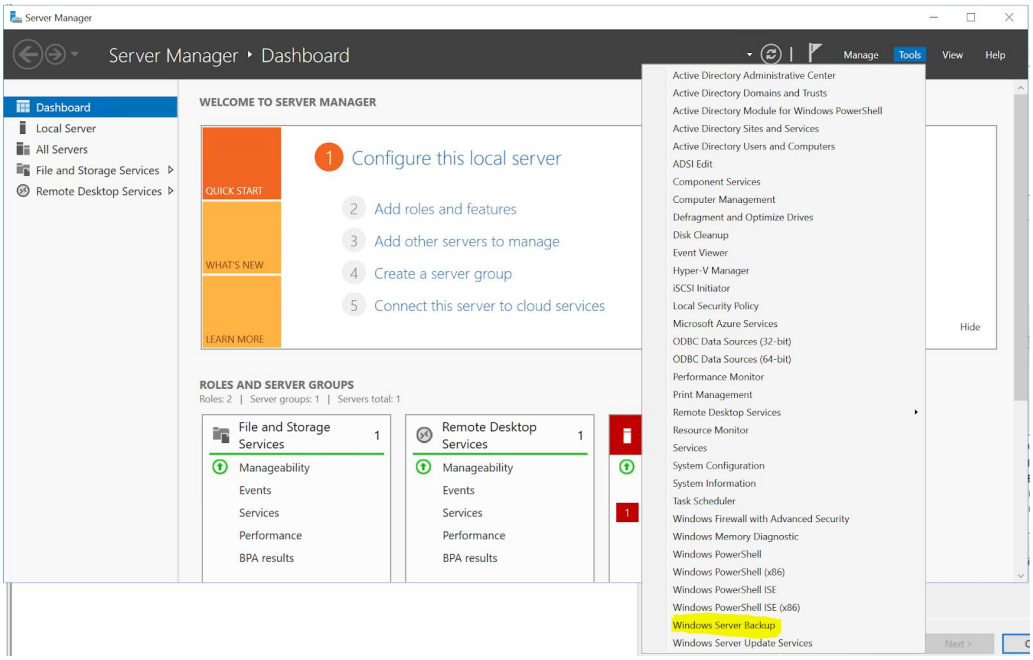
Next, select Update & Security



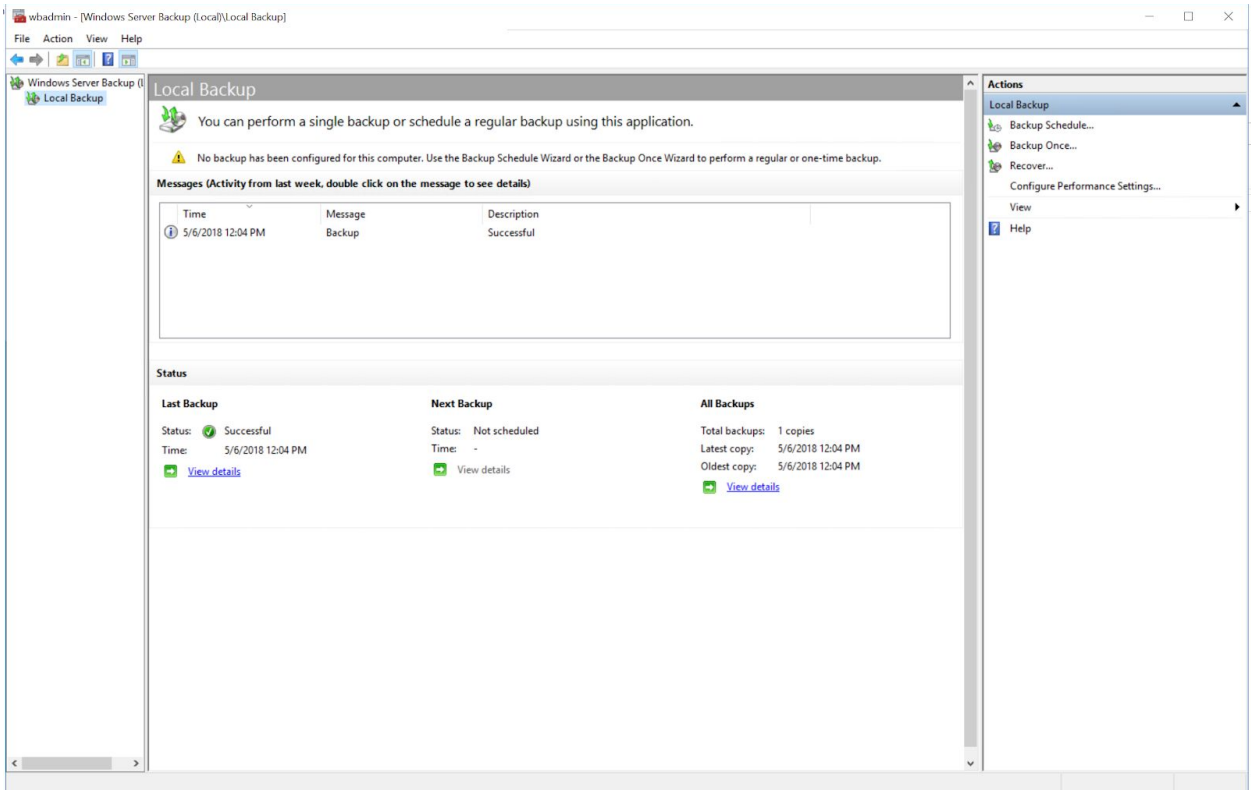
From this screen you can manually check for new updates or schedule times to automatically install updates

Backups

Backups are scheduled and configured via the Windows Server Manager too. As a prerequisite, Windows Server Backup needs to be installed in Server Manager > Add roles and features > Next > Role-based or feature-based installation > Next > Next > Next > Select “Windows Server Backup” and Install.



To create a new backup, go to the Server manager and select Tools > Windows Server Backup From here, you can schedule backups, create a system image, and restore from a system image. To create a system image click Backup Once under the Actions menu and follow the setup. A Full Metal backup will create a system image with all of the current programs and settings preserved. To restore from this image, choose the Recover option under Actions and select your system image file.


wbadmin - [Windows Server Backup (Local)\Local Backup]
File Action View Help
Windows Server Backup (l
Local Backup
Local Backup
You can perform a single backup or schedule a regular backup using this application.
No backup has been configured for this computer. Use the Backup Schedule Wizard or the Backup Once Wizard to perform a regular or one-time backup.
Messages (Activity from last week, double click on the message to see details)
Time
Message
Description
5/6/2018 12:04 PM
Backup
Successful
Status
Last Backup
Status: Successful
Time: 5/6/2018 12:04 PM
View details
Next Backup
Status: Not scheduled
Time: -
View details
All Backups
Total backups: 1 copies
Latest copy: 5/6/2018 12:04 PM
Oldest copy: 5/6/2018 12:04 PM
View details
Actions
Local Backup
Backup Schedule...
Backup Once...
Recover...
Configure Performance Settings...
View
Help

CSCE 4925: Project Aero
Project Plan

By: Alyssa Thurston, Breuna Riggins, James Sabetti, Travis Goral
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 29 September 2017
Status: Final Draft

Table of Contents

Revisions

Below is a list of any revisions made to this document.

Date	Description of Change Made	Person Making Change
9/18/2017	Draft Created	Breuna, Alyssa, James. Travis
9/18/2017	Draft Compiled	Breuna
9/18/2017	Document Edited	James
9/18/2017	Final Draft Complete	Breuna, Alyssa, James, Travis
1/26/2018	Final Draft Revisions- Added Spring Semester	Alyssa
5/4/2018	Final Draft Revisions	Alyssa

1 Project Summary

The air quality in the City of Denton is notoriously bad. It's not a surprising fact given that there are two universities and multiple major highways and roadways that run through the city's limits. Worse still, there is only a single air quality sensor within city limits, and the next closest one is almost 15 miles away from the city.

The goal of this project is to create a network of air quality sensors to measure the air quality in the city of Denton so that citizens and city personnel alike can monitor air quality and take appropriate actions. The data obtained from the sensors is to be made public, so that air quality trends can be tracked all around the city. With these requirements met, the network would provide secure, real-time, and reliable data on the air quality in the city of Denton.

With the aim of creating a robust and open source of air quality data for the city of Denton, the Open Denton group has set out to establish a network of air quality sensors to gather and compile that data. In order to help achieve this goal, our group has been tasked with the following:

- Create a web client to display data in a highly readable manner.
- Create an API that records and displays air quality data in a readable manner
- Implement a scalable database to hold and model recorded data
- Integrate sensors and controllers that will read and relay data to the database over the internet
- Create documentation that allows community members to add sensors to the network

This document will establish tasks needed to be completed this semester, a schedule, and an overall plan.

1 General Information

This section includes some general information about the team and the project.

1.1 Points of Contact

Team Members

Position	Name	Phone	E-Mail
Manager	Alyssa Thurston	(801)866-5851	alyssathurston@my.unt.edu
Editor	James Sabetti	(972)757-6437	jamessabetti@my.unt.edu
Compiler	Breuna Riggins	(214)298-7008	breunariggins@my.unt.edu
Analyst	Travis Goral	(940)465-9422	travisgoral@my.unt.edu

Client Information

Name	E-Mail
Kyle Taylor	kyletaylored@gmail.com
Dan Minshew	danminshew@gmail.com

1.2 Project Charter

1.2.1 Problem

The city of Denton lacks robust and available air quality data. Denton is the most polluted city in the state of Texas and there is a strong need for a community-oriented air quality monitoring network.

1.2.2 Statement of Work

Document requirements for a full working system by October 26th, have a final design report for the project ready by the end of the semester, and build a working prototype system during the Spring 2018 semester.

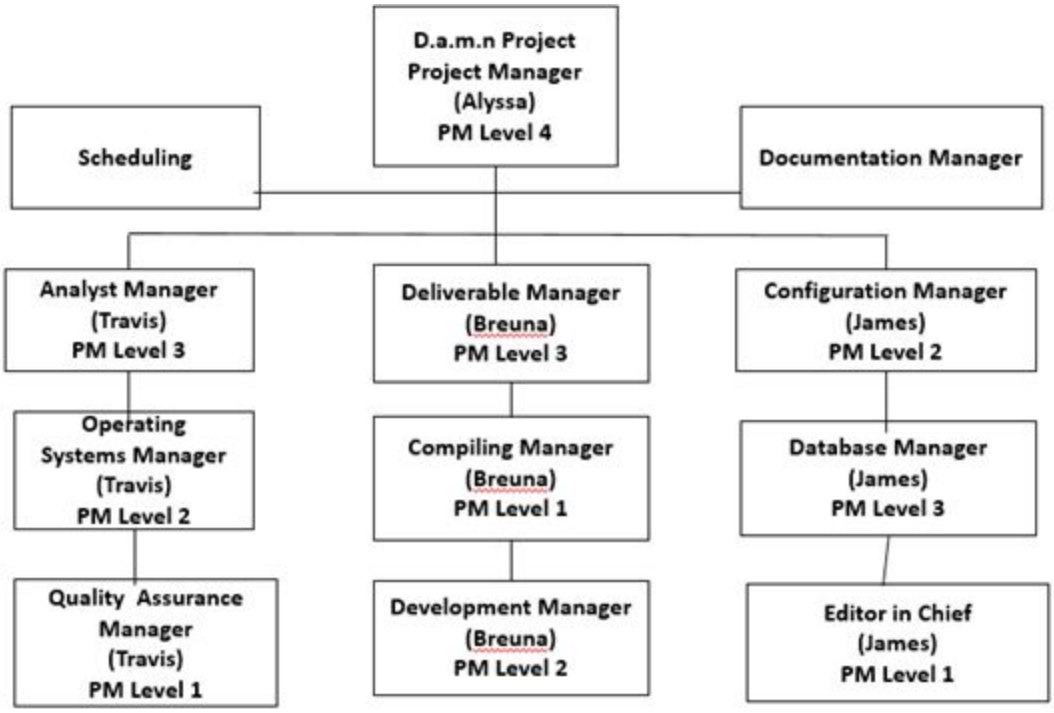
1.2.3 Objectives

Objectives may change over time as more project details are available. In general, our objectives are:

- Meet with the client regularly to discuss specific project details

- Produce a living status report and detailed plan
- Document requirements and specifications
- Test requirements with sensor controller
- Draw up design plan
- Submit final design report along with presentation
- Complete final documentation
- Prepare for delivery of the product
- Create a working prototype

1.3 Organization



1.4 Configuration Management

CCM Responsibility
Manager: J. Sabetti
Additional Staff for CM: To be recruited as needed.

Configuration Items: ensure that CM is implemented throughout the project’s life cycle.

No.	Item	Comments
1.	Database	
2.	API	
3.	Dashboard	

2 Project Details

This section contains more specific information in regards to tasks that need to be completed for this project to come to fruition.

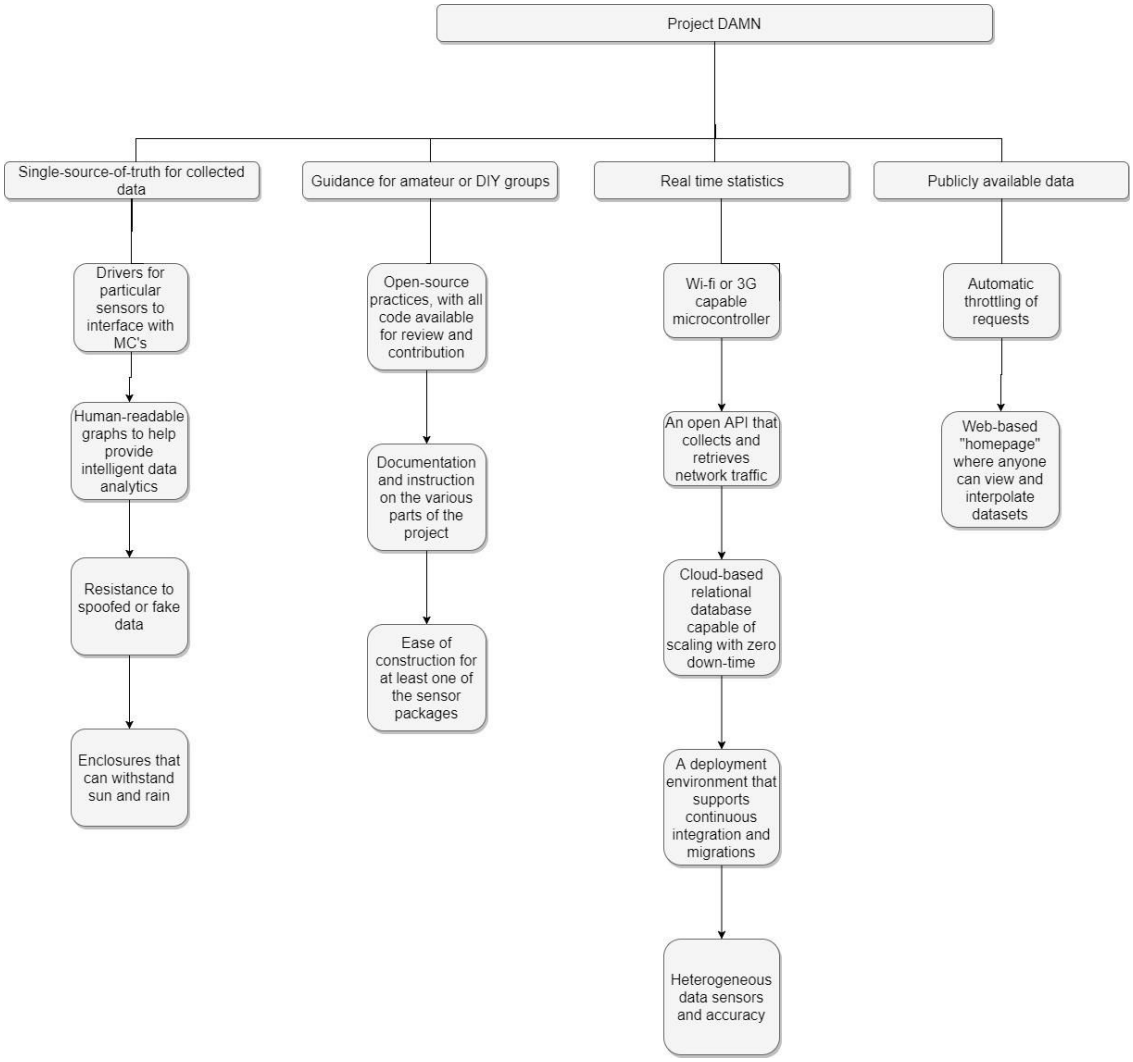
2.1 Activity List

The following is a list of activities pertinent to this project. Each draft and final form of a document has an internal due date of 24 hours before the final deadline, to accommodate any unforeseen obstacles related to the completion of this project.

Activity Number	Activity Description	Internal Due Date	Due Date
1.0	Status Report (One)	Jan 18th	Jan 19
2.0	Project Plan Revision	Jan 25th	Jan 26th
2.1	Detailed Design Revision	Jan 25th	Jan 26th
3.0	Status Report (Two)	Feb 1st	Feb 2nd
4.0	Deliverable Plan for Text Plan Draft	Feb 5th	Feb 5th
4.1	Deliverable Plan for Text Plan	Feb 8th	Feb 9th
5.0	Status Report (Three)	Feb 15th	Feb 16th
5.1	Test Plan and Procedure	Feb 15th	Feb 16th
6.0	Deliverable Report for Test Plan	Feb 18th	Feb 19th
7.0	Deliverable Plan for Maintenance Manual Draft	Feb 26th	Feb 26th
7.1	Deliverable Plan for Maintenance Manual	Mar 1st	Mar 2nd
8.0	Status Report (Four)	Mar 8th	Mar 9th
9.0	Deliverable Report for Maintenance Manual	Mar 11th	Mar 12th
10.0	Status Report (Five)	Mar 18th	Mar 19th

11.0	Midterm Peer Reviews	Mar 19th	Mar 20th
12.0	Test Readiness Review Presentations	Mar 21st	Mar 22nd
12.0	Deliverable Plan for Maintenance Manual Draft	Mar 19th	Mar 19th
12.1	Deliverable Plan for Users Manual	Mar 22nd	Mar 23nd
13.0	Status Report(six)	Mar 29th	Mar 30th
13.1	User Manual Draft	Mar 26th	Mar 26th
13.2	User Manual	Mar 29th	Mar 30th
14.0	Deliverable Report for Users Manual	Apr 1st	Apr 2
15.0	Status Report 7	Apr 12th	Apr 13th
16.0	Design Day Poster	Apr 22nd	Apr 23rd
17.0	Final presentation	Apr 26th	Apr 27th
17.1	Report Status (8)	Apr 26th	Apr 27th
18.0	Final Acceptance	Apr 30th	May 1st
18.1	Final Report	Apr 30th	May 1st
19.0	Final Peer Reviews	May 2nd	May 3rd
19.1	Final Project Notebook	May 2nd	May 3rd
20.0	Client Peer Review	May 9th	May 10th
20.1	Instructor Assessment	May 9th	May 10th

2.2 WBS



2.3 Work Products

Deliverable Name	Due Date	Date Delivered	Point of Contact
Deliverable Project Plan	Sept 20th	Sept 19th	Breuna Riggins
Status Report (One)	Sept 28th		
Project Plan	Sept 29th		
Deliverable Report for Project Plan	Oct 2nd		

Deliverable Plan for Requirements	Oct 3rd		
Status Report (Two)	Oct 12th		
Requirements Spec	Oct 17th		
Deliverable Report for Requirement Spec	Oct 18th		
Deliverable Plan for Prelim Design	Oct 23rd		
Status Report (Three)	Oct 26th		
Prelim Design	Nov 3rd		
Deliverable Report for Prelim Design	Nov 6th		
Status Report (Four)	Nov 9th		
Deliverable Plan for Detailed Design	Nov 13th		
Status Report (Five)	Nov 30th		
Detailed Design	Dec 1st		
Deliverable Report for Detailed Design	Dec 4th		
Status Report (Six)	Dec 7th		
Status Report (One)	Jan 19		
Project Plan Revision	Jan 26th		
Detailed Design Revision	Jan 26th		
Status Report (Two)	Feb 2nd		
Deliverable Plan for Test Plan	Feb 9th		
Status Report (Three)	Feb 16th		
Test Plan and Procedure	Feb 16th		

Deliverable Report for Test Plan	Feb 19th		
Deliverable Plan for Maintenance Manual	Mar 2nd		
Status Report (Four)	Mar 9th		
Deliverable Report for Maintenance Manual	Mar 12th		
Status Report (Five)	Mar 19th		
Midterm Peer Reviews	Mar 20th		
Test Readiness Review Presentations	Mar 22nd		
Deliverable Plan for Users Manual	Mar 23nd		
Status Report(six)	Mar 30th		
User Manual	Mar 30th		
Deliverable Report for Users Manual	Apr 2		
Status Report 7	Apr 13th		
Design Day Poster	Apr 23rd		
Final presentation	Apr 27th		
Report Status (8)	Apr 27th		
Final Acceptance	May 1st		
Final Report	May 1st		
Final Peer Reviews	May 3rd		
Final Project Notebook	May 3rd		
Client Peer Review	May 10th		
Instructor Assessment	May 10th		

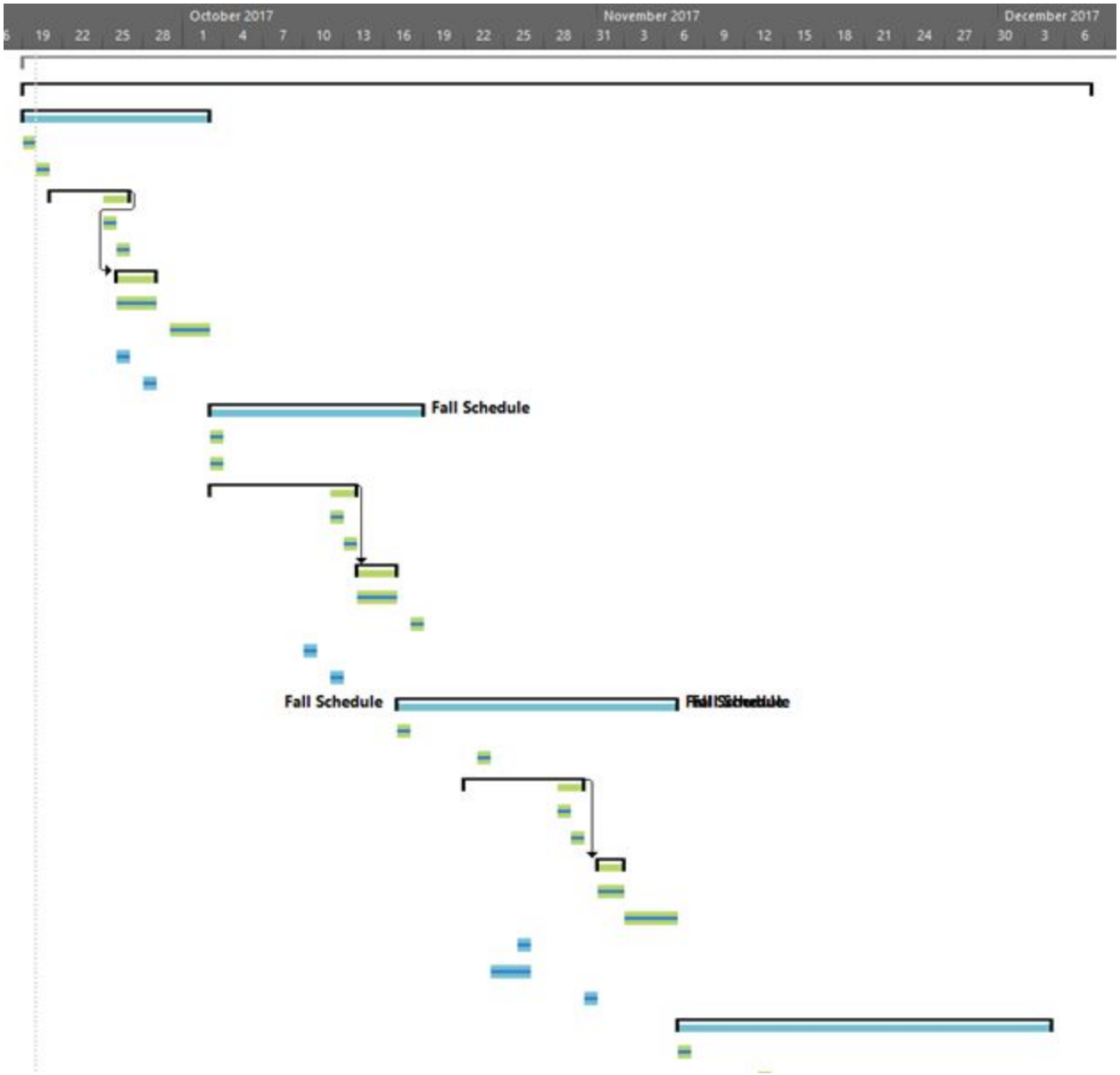
2.4 Schedule

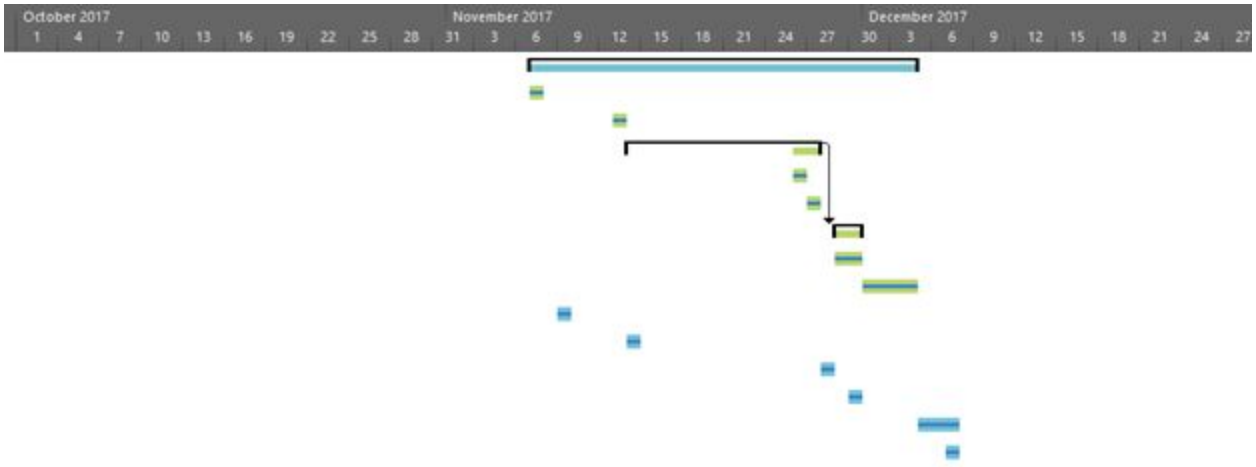
For each assignment, those that are turned in to either the client or the professor, the assignment is due (internally) 48 hours before the draft is due. This will allow 24 hours to compile the document together, and 24 hours to edit the document and turn it in by the due date. On this note, the Gantt Chart is provided below.

ID	Task Mode	Task Name	Duration	Start	Finish
0		**Project Plan**	**164 days**	**Tue 9/19/17**	**Thu 5/3/18**
1		**Fall Schedule**	**59 days**	**Tue 9/19/17**	**Thu 12/7/17**
2		**Project Plan**	**11 days**	**Tue 9/19/17**	**Mon 10/2/17**
3		Project Plan Lecture	1 day	Tue 9/19/17	Tue 9/19/17
4		Project Plan Plan	1 day	Wed 9/20/17	Wed 9/20/17
5		**Project Plan Draft**	**5 days**	**Thu 9/21/17**	**Tue 9/26/17**
6		Document Compilation	1 day	Mon 9/25/17	Mon 9/25/17
7		Document Editing	1 day	Tue 9/26/17	Tue 9/26/17
8		**Project Plan- Final**	**3 days**	**Tue 9/26/17**	**Thu 9/28/17**
9		Necessary Revisions	3 days	Tue 9/26/17	Thu 9/28/17
10		Project Plan Report	2 days	Sat 9/30/17	Mon 10/2/17
11		Team Meeting (During Class)	1 day	Tue 9/26/17	Tue 9/26/17
12		Status Report 1	1 day	Thu 9/28/17	Thu 9/28/17
13		**Requirements Specifications**	**12 days**	**Tue 10/3/17**	**Wed 10/18/17**
14		Requirements Lecture	1 day	Tue 10/3/17	Tue 10/3/17
15		Requirements Specifications Plan	1 day	Tue 10/3/17	Tue 10/3/17
16		**Requirements Specifications Draft**	**9 days**	**Tue 10/3/17**	**Fri 10/13/17**
17		Document Compilation	1 day	Thu 10/12/17	Thu 10/12/17
18		Document Editing	1 day	Fri 10/13/17	Fri 10/13/17
19		**Requirements Specifications- Final**	**2 days**	**Sat 10/14/17**	**Mon 10/16/17**
20		Necessary Revisisions	2 days	Sat 10/14/17	Mon 10/16/17
21		Requirements Specifications Report	1 day	Wed 10/18/17	Wed 10/18/17
22		Team Meeting (During Class)	1 day	Tue 10/10/17	Tue 10/10/17
23		Status Report 2	1 day	Thu 10/12/17	Thu 10/12/17
24		**Preliminary Design**	**15 days**	**Tue 10/17/17**	**Mon 11/6/17**
25		Preliminary Design Lecture	1 day	Tue 10/17/17	Tue 10/17/17
26		Preliminary Design Plan	1 day	Mon 10/23/17	Mon 10/23/17
27		**Preliminary Design Draft**	**7 days**	**Sun 10/22/17**	**Mon 10/30/17**
28		Document Compilation	1 day	Sun 10/29/17	Sun 10/29/17
29		Document Editing	1 day	Mon 10/30/17	Mon 10/30/17
30		**Preliminary Design- Final**	**2 days**	**Wed 11/1/17**	**Thu 11/2/17**

31	✓		Necessary Revisions	2 days	Wed 11/1/17	Thu 11/2/17
32	✓		Preliminary Design Report	2 days	Fri 11/3/17	Mon 11/6/17
33	✓		Status Report 3	1 day	Thu 10/26/17	Thu 10/26/17
34	✓		Requirements Review	3 days	Tue 10/24/17	Thu 10/26/17
35	✓		Team Meeting (During Class)	1 day	Tue 10/31/17	Tue 10/31/17
36	✓		**Detailed Design**	**20 days**	**Tue 11/7/17**	**Mon 12/4/17**
37	✓		Detailed Design Lecture	1 day	Tue 11/7/17	Tue 11/7/17
38	✓		Detatiled Design Plan	1 day	Mon 11/13/17	Mon 11/13/17
39	✓		**Detailed Design Draft**	**10 days**	**Tue 11/14/17**	**Mon 11/27/17**
40	✓		Document Compilation	1 day	Sun 11/26/17	Sun 11/26/17
41	✓		Document Editing	1 day	Mon 11/27/17	Mon 11/27/17
42	✓		**Detailed Design- Final**	**2 days**	**Wed 11/29/17**	**Thu 11/30/17**
43	✓		Necessary Revisions	2 days	Wed 11/29/17	Thu 11/30/17
44	✓		Detailed Design Report	2 days	Fri 12/1/17	Mon 12/4/17
45	✓		Status Report 4	1 day	Thu 11/9/17	Thu 11/9/17
46	✓		Team Meeting (During Class)	1 day	Tue 11/14/17	Tue 11/14/17
47	✓		Wrapup Lecture	1 day	Tue 11/28/17	Tue 11/28/17
48	✓		Status Report 5	1 day	Thu 11/30/17	Thu 11/30/17
49	✓		Design Review	3 days	Tue 12/5/17	Thu 12/7/17
50	✓		Status Report 6	1 day	Thu 12/7/17	Thu 12/7/17
51			**Spring Schedule**	**78 days**	**Tue 1/16/18**	**Thu 5/3/18**
52			Project Plan Updates	9 days	Tue 1/16/18	Fri 1/26/18
53			Detailed Design Updates	9 days	Tue 1/16/18	Fri 1/26/18
54			Status Report 1	1 day	Fri 1/19/18	Fri 1/19/18
55			Status Report 2	1 day	Fri 2/2/18	Fri 2/2/18
56			**Test Plan and Procedure**	**20 days**	**Tue 1/23/18**	**Mon 2/19/18**
57			Lecture	3 days	Tue 1/23/18	Thu 1/25/18
58			Test Plan and Procedure Plan	1 day	Tue 1/30/18	Tue 1/30/18
59			**Test Plan and Procedure Plan Draft**	**12 days**	**Tue 1/30/18**	**Wed 2/14/18**

ID	Task Mode	Task Name	Duration	Start	Finish
60		Document Compilation	1 day	Tue 2/13/18	Tue 2/13/18
61		Document Editing	1 day	Wed 2/14/18	Wed 2/14/18
62		**Test Plan and Procedure Final Draft**	**3 days**	**Wed 2/14/18**	**Fri 2/16/18**
63		Necessary Revisions	3 days	Wed 2/14/18	Fri 2/16/18
64		Test Plan and Procedure Report	1 day	Mon 2/19/18	Mon 2/19/18
65		**Maintenance Manual**	**35 days**	**Tue 2/13/18**	**Mon 4/2/18**
66		Lecture	1 day	Tue 2/13/18	Tue 2/13/18
67		Maintenance Manual Plan	1 day	Thu 2/15/18	Thu 2/15/18
68		**Maintenance Manual Draft**	**15 days**	**Thu 2/15/18**	**Wed 3/7/18**
69		Document Compilation	3 days	Sat 3/3/18	Tue 3/6/18
70		Document Editing	1 day	Wed 3/7/18	Wed 3/7/18
71		**Maintenance Manual Final Draft**	**3 days**	**Wed 3/7/18**	**Fri 3/9/18**
72		Necessary Revisions	3 days	Wed 3/7/18	Fri 3/9/18
73		Maintenance Manual Report	1 day	Mon 3/12/18	Mon 3/12/18
74		**User Manual**	**33 days**	**Thu 2/15/18**	**Mon 4/2/18**
75		Lecture	1 day	Thu 2/15/18	Thu 2/15/18
76		User Manual Plan	1 day	Fri 2/16/18	Fri 2/16/18
77		**User Manual Draft**	**29 days**	**Sat 2/17/18**	**Wed 3/28/18**
78		Document Compilation	3 days	Sat 3/24/18	Tue 3/27/18
79		Document Editing	1 day	Wed 3/28/18	Wed 3/28/18
80		**User Manual Final Draft**	**3 days**	**Wed 3/28/18**	**Fri 3/30/18**
81		Necessary Revisions	3 days	Wed 3/28/18	Fri 3/30/18
82		User Manual Report	1 day	Mon 4/2/18	Mon 4/2/18
83		Status Report 3	1 day	Fri 2/16/18	Fri 2/16/18
84		Application Prototype	24 days	Tue 1/30/18	Fri 3/2/18
85		Status Report 4	1 day	Fri 3/2/18	Fri 3/2/18
86		Status Report 5	1 day	Mon 3/19/18	Mon 3/19/18
87		Test Readiness Presentation	1 day	Thu 3/22/18	Thu 3/22/18
88		Status Report 6	1 day	Fri 3/30/18	Fri 3/30/18
89		Status Report 7	1 day	Fri 4/13/18	Fri 4/13/18
90		Application Revisions	31 days	Sat 3/3/18	Fri 4/13/18
91		Design Day Poster	15 days	Tue 4/3/18	Mon 4/23/18
92		Design Day	1 day	Fri 4/27/18	Fri 4/27/18
93		Status Report 8	1 day	Fri 4/27/18	Fri 4/27/18
94		Final Acceptance Test Procedure and Results	1 day	Tue 5/1/18	Tue 5/1/18
95		Final Report	1 day	Tue 5/1/18	Tue 5/1/18
96		Final Project Notebooks	1 day	Thu 5/3/18	Thu 5/3/18


October 2017
November 2017
December 2017
19 22 25 28 1 4 7 10 13 16 19 22 25 28 31 3 6 9 12 15 18 21 24 27 30 3 6
Fall Schedule
Fall Schedule



2.5 Risks

With a project of this size, there is going to be some level of risk. Risks are listed below by level of severity along with possible ways to mitigate those risks.

2.5.1 High

- Scope creep – The project scope may grow immensely, forcing the group to push back deadlines. In some cases, deadlines may have to be moved to earlier dates to allow a certain amount of time slack.
- Learning curve – Group members will need to quickly familiarize themselves with client frameworks and current databases. Significant time will need to be spent learning how to interact with their existing sensors and databases.

2.5.2 Medium

- Other classwork – Homework and projects from other classes may eat into time to work on Capstone. Not much can be done about this risk except to make sure that individual group members do their work and communicate with the group.
- Lack of Skills – Group members may not possess sufficient skills in coding, documentation or other areas needed to complete the project. Although this is not a huge risk for the first semester, it can become a big risk by the Spring if group members do not take the time to learn and hone their skills

2.5.3 Low

- Group member availability – A group member may be busy with other things or unable to meet or turn in work. Other group members should be able to pick up the slack. Google Hangouts and Drive will be used as an alternative to personal meetings as needed.

2.6 Issues and Action Items

For now, because the project is just starting out, there aren't many issues. There are a few action items that are being worked on. Should any issues arise down the road, this document will be updated to include the following items: Description of the issue, when the issue was identified, when the issue was resolved, and who is responsible for the resolution.

Listed below are the top five action items as the project currently stands.

Action Item Number	Action Item Description	Assigned Person(s)	Open Date	Closed Date	Status
1	Research database servers	All	9/26		In progress
2	Research and discuss possible API solutions	All	TBD		In progress
3	Research EPA requirements for what constitutes as clean air and dirty air	All	9/26		In progress
4	Learn how to best utilize Slack (our means of communication with our sponsors)	All	9/26		In progress
5	Research dashboard options	All	TBD		In progress

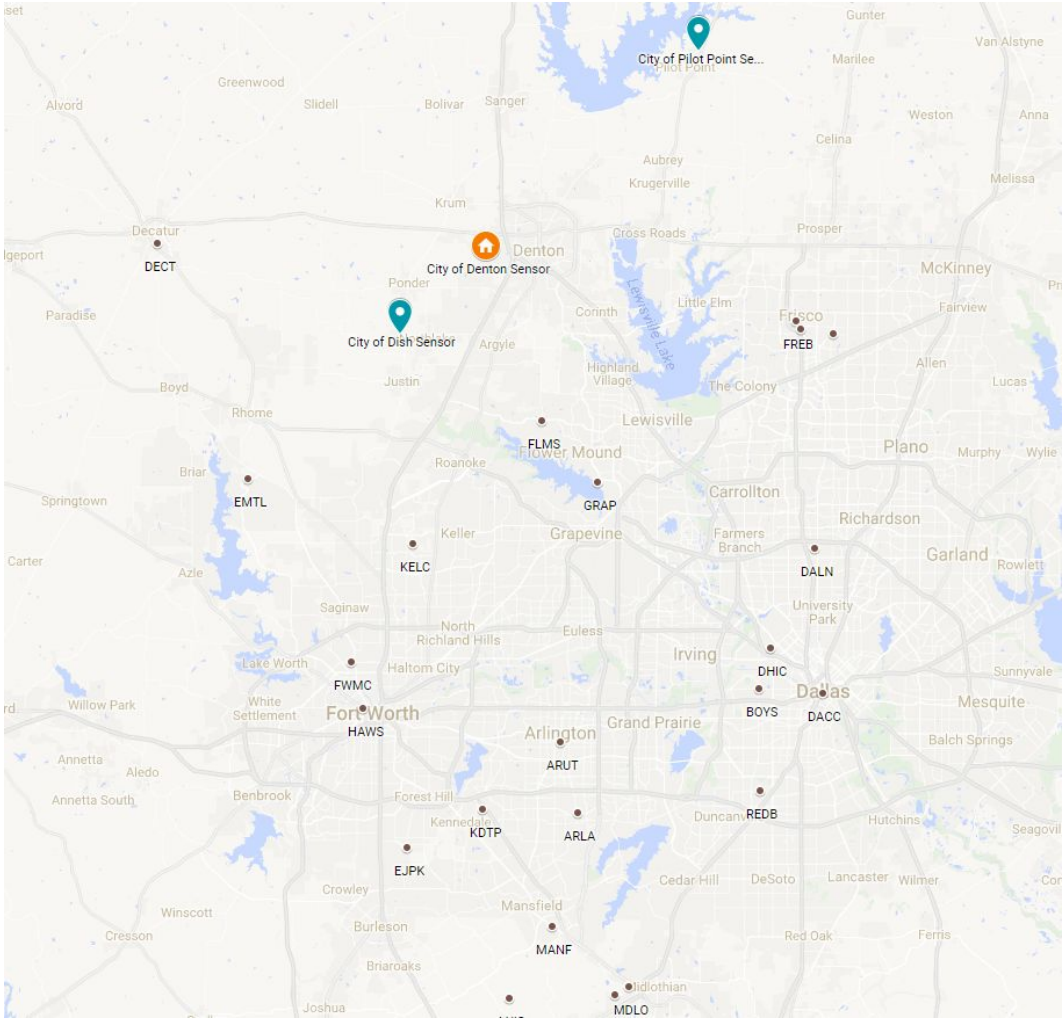
Project Aero

Denton Air Monitoring Network

“...the worst ozone levels were found in the northwest quadrant of the DFW area... caused by the predominant SE to NW winds that blow pollution from the coast up through the coal and gas patches of East and Central Texas, over the Midlothian Industrial Complex and North Texas central urban cores into Northwest Tarrant, Wise and Denton counties.”

The problem

- No sensors around the Denton area
- No sensors directly North of Denton
- Bottom-line: we have no idea how bad Denton's air really is



Introducing Project Aero

What is Project Aero?

- A network of Air Quality Sensors
- Allows citizens to create their own sensor and contribute their data
- Helps pinpoint where the worst air is in Denton

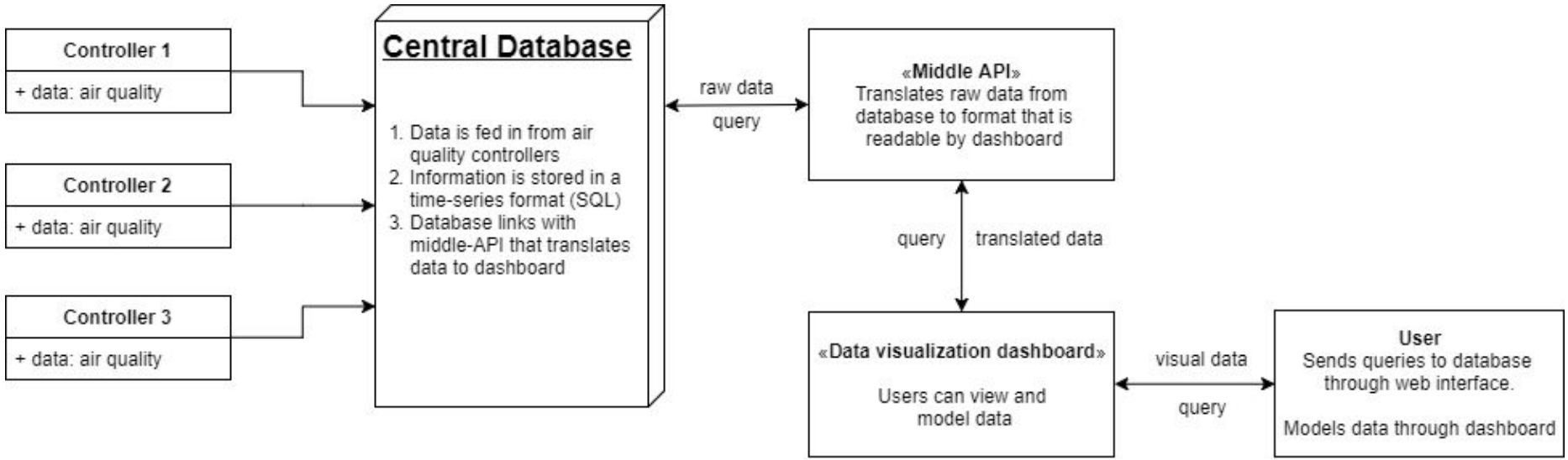
Main Goals

- System should be *scalable* or have the ability to increase or decrease in size on demand
- System should be be *modular* in the sense that sensors can be added or removed as needed
- System should be *useable*, specifically, easy to use, understand, and access
- System should Publicly Available and made open source
- System should be secure
- Sensors should hardy

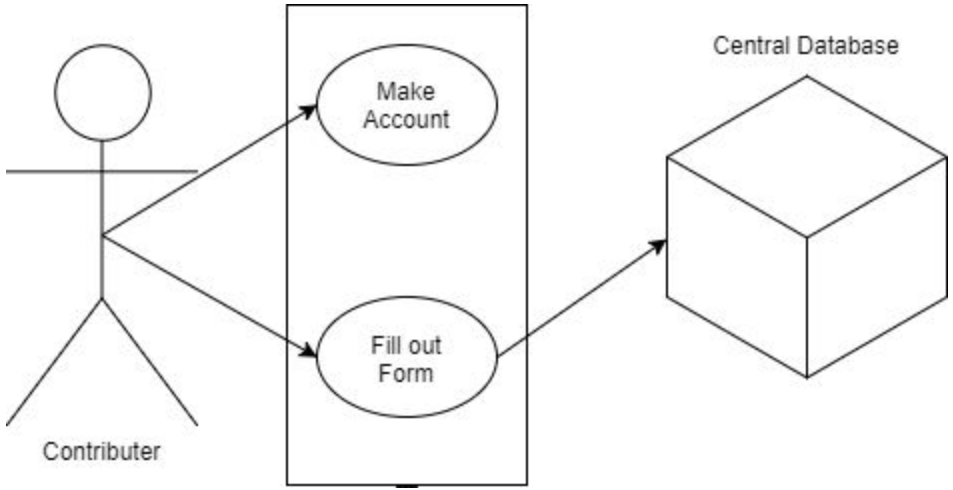
Denton's "average ozone level stands at about 80 ppb," the highest in Texas in 2016.

Functional Requirements

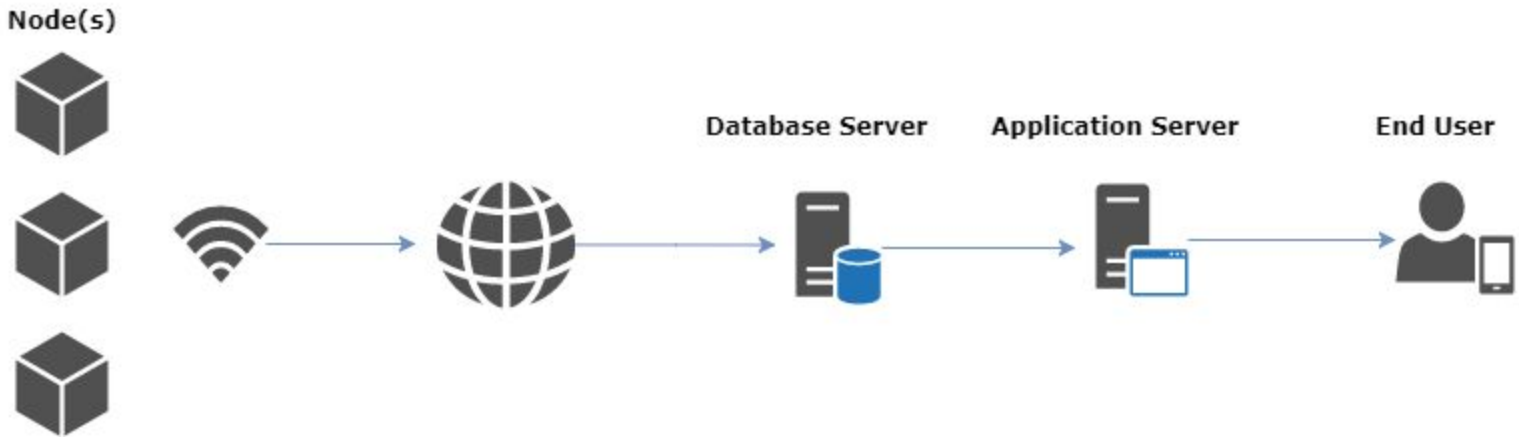
- Open Source
- Data in time-series format
- Open API to collect and retrieve data
- Web-based Front end
  - Should use graphs, meters, charts
- Documentation and instructions
- Secure
- API should be restful



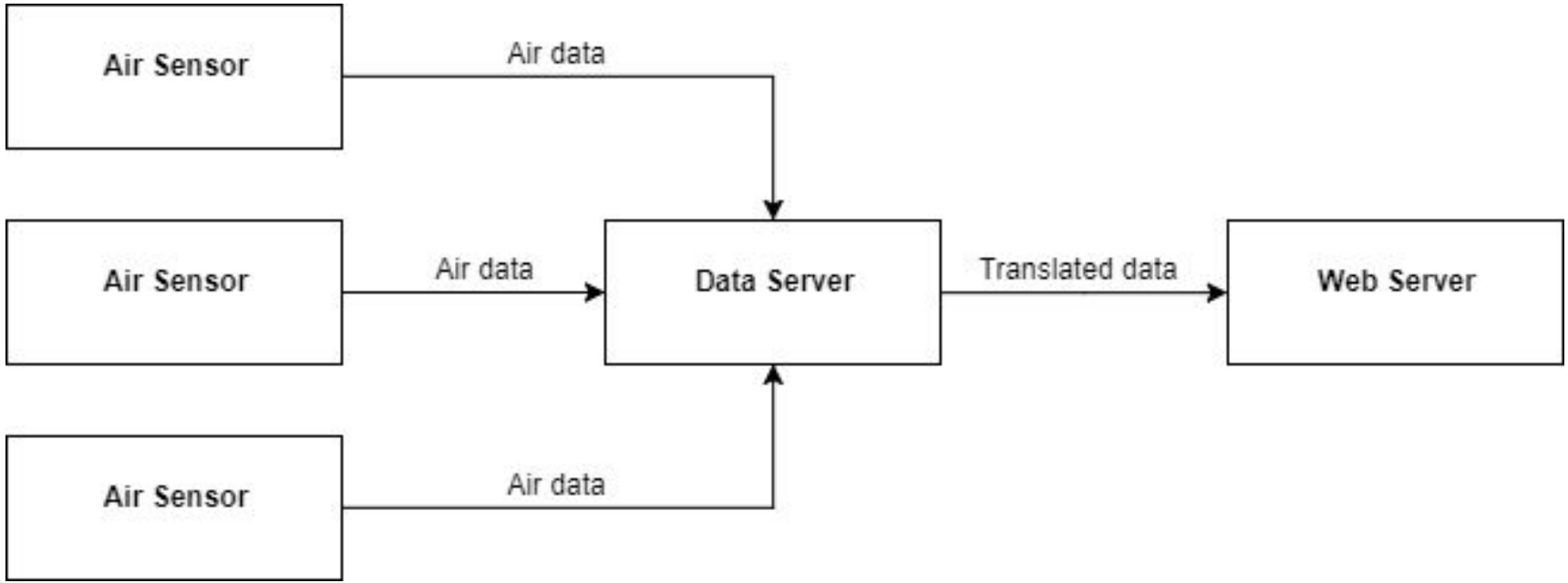
Software Requirements



User Requirements



Communications Requirements



Hardware Requirements

Non-Functional Requirements

- Minimal downtime
- Updated in real time
- Easy to maintain past implementation
- Hardy system, resistant to erroneous data
- Documentation must be made available to the client

Sources & Further Reading

- 1: Dallas Observer: Why Denton's Air Quality Is the Worst in Texas
  http://www.dallasobserver.com/news/why-dentons-air-quality-is-the-worst-in-texas-7108471
- 2: Denton Record Chronicle: Denton air ranks as state’s dirtiest
  http://www.dentonrc.com/news/news/2016/11/29/denton-air-ranks-as-states-dirtiest

CSCE 4925: Project Aero Requirements Specifications

By: Alyssa Thurston, Breuna Riggins, James Sabetti, Travis Goral
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 17 October 2017
Status: Final Draft

Table of Contents

Revisions

Below is a list of any revisions made to this document.

Date	Description of Change Made	Person Making Change
11/7/2018	Draft Created	Breuna, Alyssa, James. Travis
11/23/2018	Draft Compiled	Breuna
11/24/2018	Document Edited	James
12/1/2018	Final Draft Complete	Breuna, Alyssa, James, Travis
5/4/2018	Final Draft Revisions	Alyssa, Travis

1 Project Summary

The air quality in the City of Denton is notoriously bad. It's not a surprising fact given that there are two universities and multiple major highways and roadways that run through the city's limits. Worse still, there is only a single air quality sensor within city limits, and the next closest one is almost 15 miles away from the city.

The goal of this project is to create a network of air quality sensors to measure the air quality in the city of Denton so that citizens and city personnel alike can monitor air quality and take appropriate actions. The data obtained from the sensors is to be made public, so that air quality trends can be tracked all around the city. With these requirements met, the network would provide secure, real-time, and reliable data on the air quality in the city of Denton.

With the aim of creating a robust and open source of air quality data for the city of Denton, the Open Denton group has set out to establish a network of air quality sensors to gather and compile that data. In order to help achieve this goal, our group has been tasked with the following:

- Create a web client to display data in a highly readable manner.
- Create an API that records and displays air quality data in a readable manner
- Implement a scalable database to hold and model recorded data
- Integrate sensors and controllers that will read and relay data to the database over the internet
- Create documentation that allows community members to add sensors to the network

This document will serve as the guiding documentation for the implementation of this project.

2 Overall Description

For our product, our goal is to design a web based client, a database, and an API to get the data from one to the other. The web based client should be accessible, up-to-date and easy to use. The database should contain as few errors as possible, should be scalable, and presented in a time-series format. End users should be able to access the website at anytime from as many device types as possible. Equally important, if a user wants to contribute a node, this needs to be made as easily as possible as well.

2.1 Functions

The web client will perform the following functions:

- 2.1.0: Display air quality statistics in an easy to read, graphical format
- 2.1.1: Provide users with a map of all sensors within city limits
- 2.1.2: Provide a method of looking up historical data
- 2.1.3: Show what sensors are online and offline
- 2.1.4: If a sensor is offline, show when it was last online or last “seen”
- 2.1.5: Display data for a single sensor
- 2.1.6: Provide a help menu for users needing help with the system
- 2.1.7: Allow a user to login to an account
- 2.1.8: Allow a user to create an account
- 2.1.9: When logged in, show a map of where the user’s sensor’s are located
- 2.1.10: When logged in, show sensor information
- 2.1.11: When logged in, show statistics from the user’s sensor’s
- 2.1.12: Provide location data of a sensor, either via address or through coordinates.
- 2.1.13: Verify that users are able to create an account.
- 2.1.14: The map should use different icons for sensors that are online and offline
- 2.1.15: Users should be able to add sensors into the network
- 2.1.16: Users should be able to delete their sensors from the network

The database will perform the following functions and have the following qualities:

- 2.1.17: Store data from the sensor(s)
- 2.1.18: Accept user reported data
- 2.1.19: The system should accept data from any sensor, regardless of type.
- 2.1.20: The database should be presented in a time series format

Some other requirements of this project include:

- 2.1.21: It should be able to store any air quality value that a user is able to provide with their sensor
- 2.1.22: Verify that the data reported from the sensor is not a outlier, and somewhat matched the data reported from other sensors.
- 2.1.23: A large volume of users should be able to use this system without issue.
- 2.1.24: The system should be secure.
- 2.1.25: Raw data should also be made available for download

2.1.26: The system should refresh itself with new data pulled from the sensors every five minutes.
2.1.27: System should notify a user if the data that is being pulled is an outlier

2.2 Use Cases

This system is designed to be used to report the level of air pollution and the quality of the air dependent on human pollution, the environment, and weather circumstances, to be recorded and reported by both ourselves and other users.

2.3 Operating Environment

The operating environment for this project is presumably a server that houses all of the database information, that is then recorded and fed to a dashboard for users to read and interpret the statistics presented by the users.

2.4 User Documentation

Users will be primarily using this system in one of two ways: they will either be accessing the information via the dashboard (at which point, there would be no need for documentation), or they will be inputting the information recorded by their devices, at which point the system would need to document what the users have input into the system and verify that information to make sure that it is correct and accurate. All user inputs would need to be documented in order to keep the legitimacy of our statistics reported.

2.5 Assumptions and Dependencies

We are assuming, while building these requirements, than the information reported by the users would be reported by the device they have made and own, not information submitted by the user's hand. We are also assuming that our database can correctly store all of this information separate from the dashboard, and that the dashboard merely reads out the information to the user.

2.6 Project Constraints

We are constrained in terms of time and the resources given to us. We will have no control over the machine that will be given to us to record all of this information with, and we are constrained by a year time limit in order to accomplish everything we have set out to do. As a result, while this project should not push the year long time limit, it may lack the polish and finish that a fully realized system with a longer timeline would normally have.

3 External Interfaces

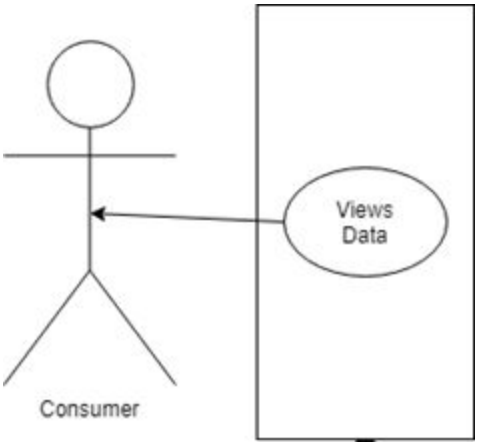
3.1 User

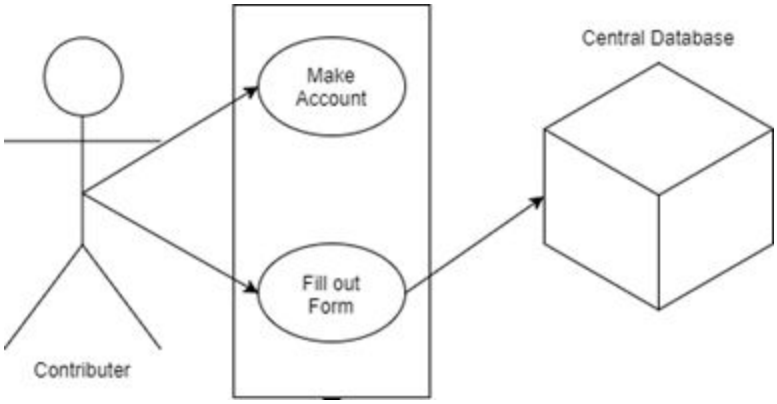
The user interface will be a responsive web based design, so that the project can be accessible from virtually any device with an internet connection. The readings should be shown graphically, using graphs, meters, and more, so users know what the levels are, if they're unhealthy or not, and can view trends over time.

As appropriate, charts from the Data Viz Project will be selected to display the data in a way that is friendly to users. Some of these may include an angular gauge, a pie chart, line graphs, grouped bar graphs, and more.

The user interface will be browser based, and should not be restricted to specific browsers or clients. It should adapt to fit both a desktop computer screen and a mobile screen. It should be available to use on all web browsers, such as Microsoft Edge, Mozilla Firefox, Google Chrome, and Mac's Safari.

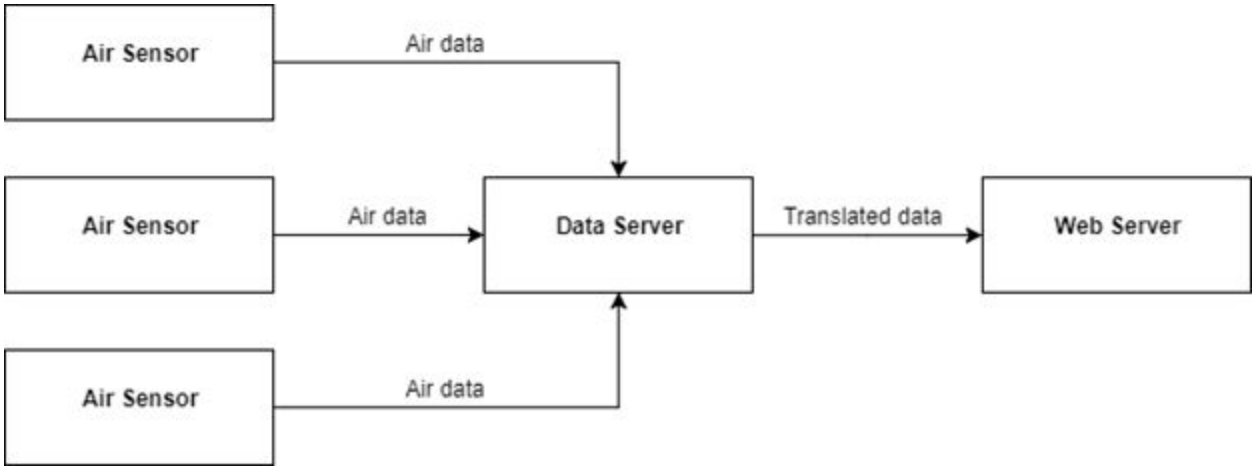
Two different user types were defined in meetings with the client- a consumer and contributor. The consumer simply views data on the UI, while the contributor will be able to create an account and fill out a form in order to contribute a sensor to the network. These functions are shown in the following diagrams.





3.2 Hardware

The project should be able to load data into a database from the air quality sensors, then take the data from the database and represent it in a human readable format. A diagram showing these details is shown on the next page.



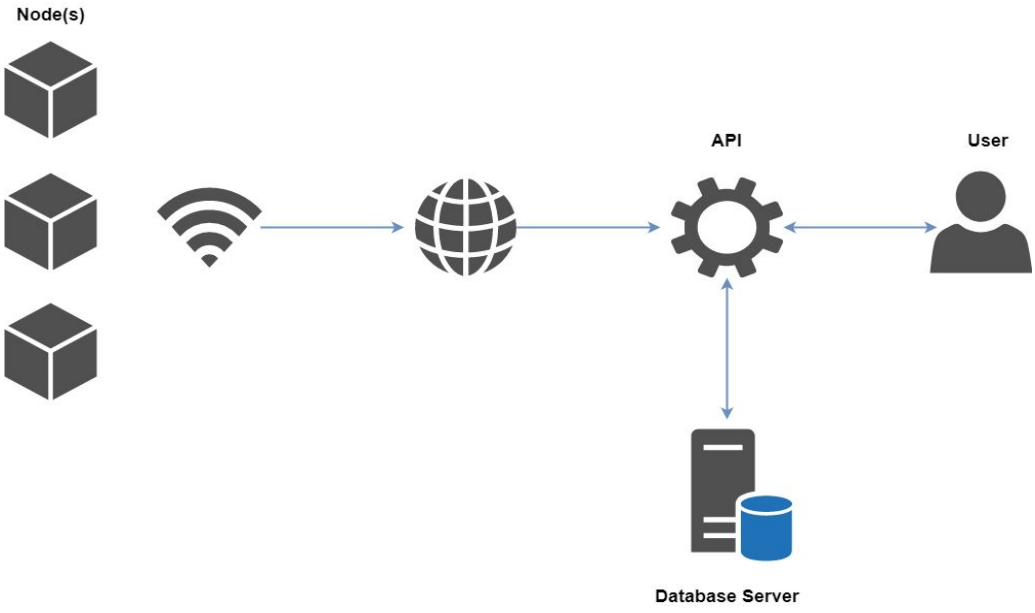
3.3 Software

This product will have several interactions between components. The sensors will send data, and the data will be populated into a database. A middle agent will then take the data and analyze it and make it more user friendly, and then the user interface will then take the prepared data and display it. A diagram is shown of the way this system will work in Section 6.0—System Features.

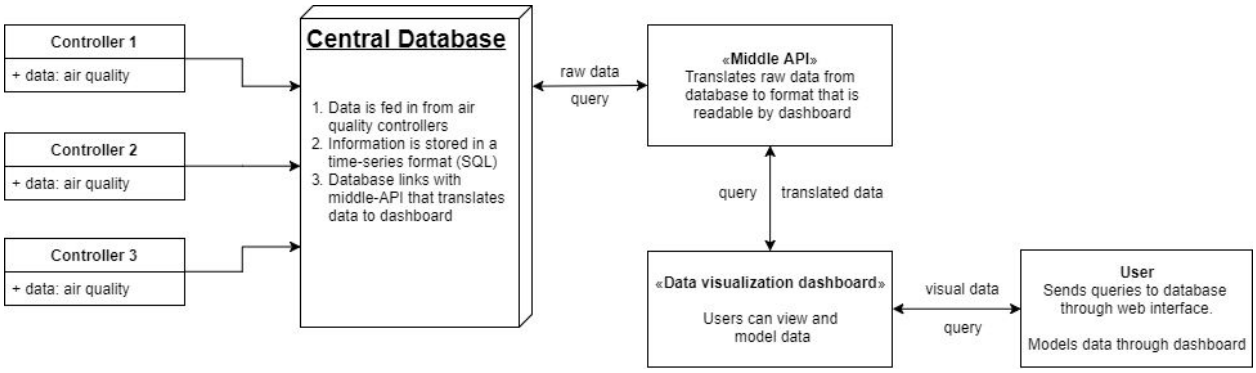
Messages will mainly be flowing out of the database into the UI, but some will be taken in the opposite direction, specifically when a citizen wants to connect a sensor to the network. This case is shown in section 2.1—User.

3.4 Communications

The main form of communication in this system is done over the internet. The sensors will be connected to the internet either via WiFi or a cell connection which will allow the data to be read into the database. A diagram of this model is shown below.



The system shall be comprised of several distinct functional requirements that outline the basic roles required for desired operation. They are detailed in a high-level orientation in the diagram below:



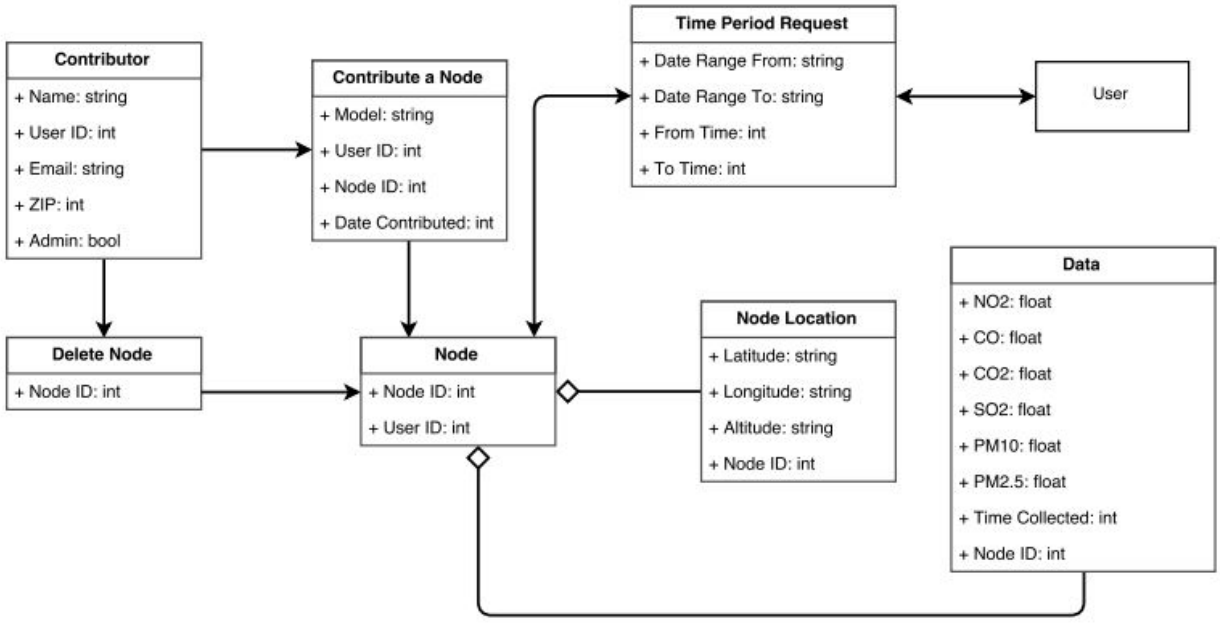
In order to keep the scope manageable at this stage, the overall vision of the system has been condensed into three primary functional requirements:

- A central time-series database
- A Middle API that translates data
- Data visualization dashboard

Later on, requirements will be broken down from these main three as needed. Additional requirements will be detailed as they become apparent.

4 Class Diagrams

The following depicts the different data that will be stored in the system and what the database might look like.



5 Non-Functional Requirements

An effort should be made to fulfill as many non-function requirements as possible. These requirements outline more of the general look and feel of the system as well as security, safety, and privacy to name a few. Below are listed the main categories of concern as well as the requirements of the system in each respective category.

5.1 Performance

- Statistics and data models shall be presented in real time and be publicly available.
- Requests should be automatically throttled to deal with traffic.
- Data queries shall be completed in the order of milliseconds to a few seconds.

5.2 Safety

- System shall have minimal downtime
- Data should not be affected by malfunctioning nodes

5.3 Security

- System shall comply with local, state and federal data security laws
- System shall be protected against normally expected attacks
- Integrity of data shall be maintained and must not be wrongfully altered by authorized or unauthorized parties.
- Users must be authenticated by the system before being granted access.
- Appropriate permissions must be enforced to protect data.
- Physical access to system should be restricted as needed.

5.4 Software Quality

- A process will be implemented for discovering and mitigating faults and bugs in the code
- QA testing should take place so that the functionality of the system can be verified
- Data gathered by sensors shall be publicly available.
- Code shall be open source.
- Web interface shall be usable by the general public.
- System database must be scalable and able to accept additional sensor controllers.
- System must be able to be easily maintained past implementation.

5.5 Legal / Social

- System and controllers hall comply with applicable local, state, and federal law

5.6 Environmental

- System shall comply with all applicable environmental regulations
- Proper care and diligence should be taken to maintain awareness of environmental impacts and mitigate these impacts where possible

5.7 Business Rules

- The general public may perform roles such as viewing data and contributing code changes.
- Administration and operation must be handled solely by the client and authorized parties.
- Proper system documentation must be available to client

8.0 Other Requirements

This area is reserved for additional requirements that do not follow within the previous categories. They will be added when and if they become apparent and necessary.

CSCE 4925: Project Aero
Preliminary Design

By: Alyssa Thurston, Breuna Riggins, James Sabetti, Travis Goral
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 3 November 2017
Status: Final Draft

Table of Contents

Revisions

Below is a list of any revisions made to this document.

Date	Description of Change Made	Person Making Change
11/7/2018	Draft Created	Breuna, Alyssa, James. Travis
11/23/2018	Draft Compiled	Breuna
11/24/2018	Document Edited	James
12/1/2018	Final Draft Complete	Breuna, Alyssa, James, Travis
5/1/2018	Final Draft Revisions	Alyssa, Travis

1 Project Summary

The air quality in the City of Denton is notoriously bad. It's not a surprising fact given that there are two universities and multiple major highways and roadways that run through the city's limits. Worse still, there is only a single air quality sensor within city limits, and the next closest one is almost 15 miles away from the city.

The goal of this project is to create a network of air quality sensors to measure the air quality in the city of Denton so that citizens and city personnel alike can monitor air quality and take appropriate actions. The data obtained from the sensors is to be made public, so that air quality trends can be tracked all around the city. With these requirements met, the network would provide secure, real-time, and reliable data on the air quality in the city of Denton.

With the aim of creating a robust and open source of air quality data for the city of Denton, the Open Denton group has set out to establish a network of air quality sensors to gather and compile that data. In order to help achieve this goal, our group has been tasked with the following:

- Create a web client to display data in a highly readable manner.
- Create an API that records and displays air quality data in a readable manner
- Implement a scalable database to hold and model recorded data
- Integrate sensors and controllers that will read and relay data to the database over the internet
- Create documentation that allows community members to add sensors to the network

This document will serve as the guiding documentation for the implementation of this project.

2 Overall Description

For our product, our goal is to design a web based client, a database, and an API to get the data from one to the other. The web based client should be accessible, up-to-date and easy to use. The database should contain as few errors as possible, should be scalable, and presented in a time-series format. End users should be able to access the website at anytime from as many device types as possible. Equally important, if a user wants to contribute a node, this needs to be made as easily as possible as well.

2.1 Functions

The web client will perform the following functions:

- 2.1.0: Display air quality statistics in an easy to read, graphical format
- 2.1.1: Provide users with a map of all sensors within city limits
- 2.1.2: Provide a method of looking up historical data
- 2.1.3: Show what sensors are online and offline
- 2.1.4: If a sensor is offline, show when it was last online or last “seen”
- 2.1.5: Display data for a single sensor
- 2.1.6: Provide a help menu for users needing help with the system
- 2.1.7: Allow a user to login to an account
- 2.1.8: Allow a user to create an account
- 2.1.9: When logged in, show a map of where the user’s sensor’s are located
- 2.1.10: When logged in, show sensor information
- 2.1.11: When logged in, show statistics from the user’s sensor’s
- 2.1.12: Provide location data of a sensor, either via address or through coordinates.
- 2.1.13: Verify that users are able to create an account.
- 2.1.14: The map should use different icons for sensors that are online and offline
- 2.1.15: Users should be able to add sensors into the network
- 2.1.16: Users should be able to delete their sensors from the network

The database will perform the following functions and have the following qualities:

- 2.1.17: Store data from the sensor(s)
- 2.1.18: Accept user reported data
- 2.1.19: The system should accept data from any sensor, regardless of type.
- 2.1.20: The database should be presented in a time series format

Some other requirements of this project include:

- 2.1.21: It should be able to store any air quality value that a user is able to provide with their sensor
- 2.1.22: Verify that the data reported from the sensor is not a outlier, and somewhat matched the data reported from other sensors.
- 2.1.23: A large volume of users should be able to use this system without issue.
- 2.1.24: The system should be secure.
- 2.1.25: Raw data should also be made available for download

2.1.26: The system should refresh itself with new data pulled from the sensors every five minutes.
2.1.27: System should notify a user if the data that is being pulled is an outlier

2.2 Use Cases

This system is designed to be used to report the level of air pollution and the quality of the air dependent on human pollution, the environment, and weather circumstances, to be recorded and reported by both ourselves and other users.

2.3 Operating Environment

The operating environment for this project is presumably a server that houses all of the database information, that is then recorded and fed to a dashboard for users to read and interpret the statistics presented by the users.

2.4 User Documentation

Users will be primarily using this system in one of two ways: they will either be accessing the information via the dashboard (at which point, there would be no need for documentation), or they will be inputting the information recorded by their devices, at which point the system would need to document what the users have input into the system and verify that information to make sure that it is correct and accurate. All user inputs would need to be documented in order to keep the legitimacy of our statistics reported.

2.5 Assumptions and Dependencies

We are assuming, while building these requirements, than the information reported by the users would be reported by the device they have made and own, not information submitted by the user's hand. We are also assuming that our database can correctly store all of this information separate from the dashboard, and that the dashboard merely reads out the information to the user.

2.6 Project Constraints

We are constrained in terms of time and the resources given to us. We will have no control over the machine that will be given to us to record all of this information with, and we are constrained by a year time limit in order to accomplish everything we have set out to do. As a result, while this project should not push the year long time limit, it may lack the polish and finish that a fully realized system with a longer timeline would normally have.

3 External Interfaces

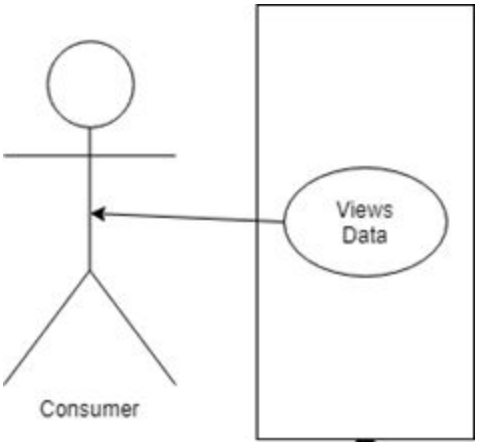
3.1 User

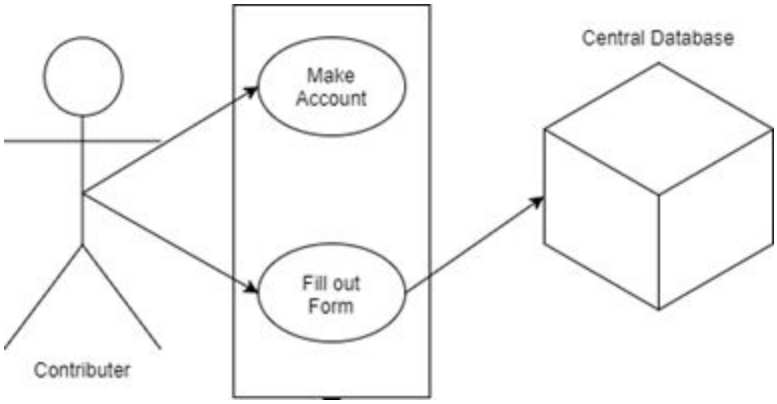
The user interface will be a responsive web based design, so that the project can be accessible from virtually any device with an internet connection. The readings should be shown graphically, using graphs, meters, and more, so users know what the levels are, if they're unhealthy or not, and can view trends over time.

As appropriate, charts from the Data Viz Project will be selected to display the data in a way that is friendly to users. Some of these may include an angular gauge, a pie chart, line graphs, grouped bar graphs, and more.

The user interface will be browser based, and should not be restricted to specific browsers or clients. It should adapt to fit both a desktop computer screen and a mobile screen. It should be available to use on all web browsers, such as Microsoft Edge, Mozilla Firefox, Google Chrome, and Mac's Safari.

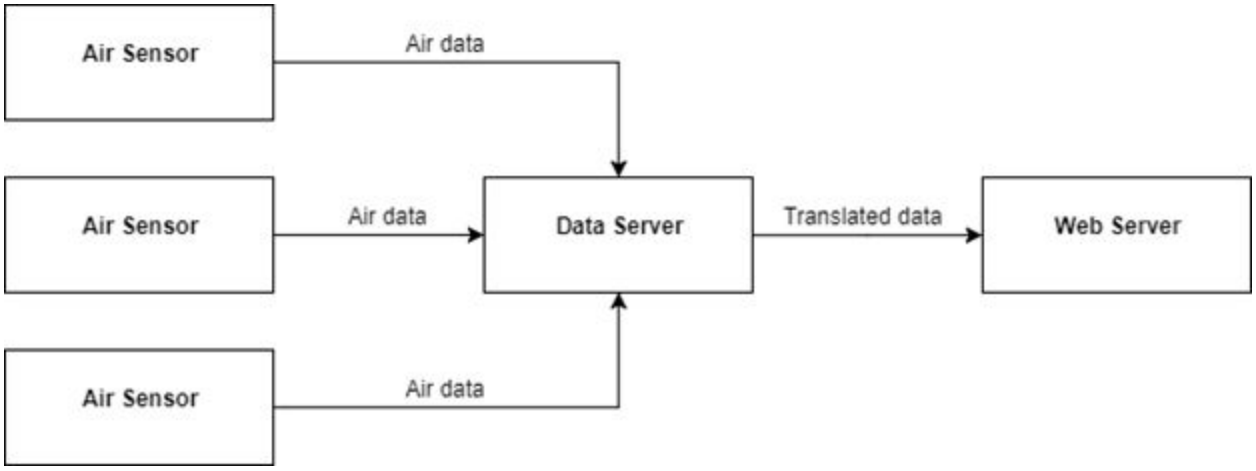
Two different user types were defined in meetings with the client- a consumer and contributor. The consumer simply views data on the UI, while the contributor will be able to create an account and fill out a form in order to contribute a sensor to the network. These functions are shown in the following diagrams.





3.2 Hardware

The project should be able to load data into a database from the air quality sensors, then take the data from the database and represent it in a human readable format. A diagram showing these details is shown on the next page.



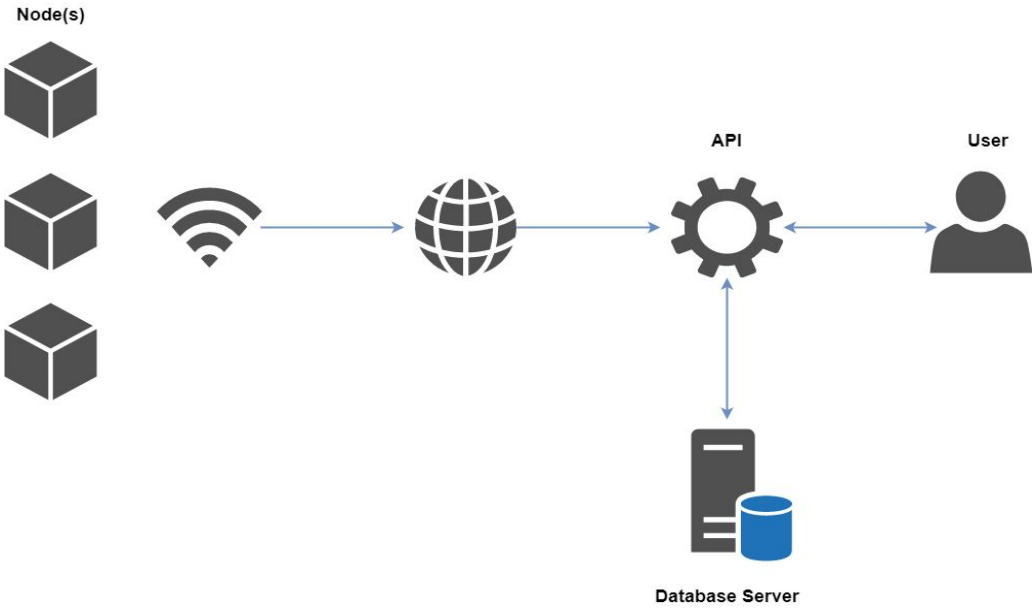
3.3 Software

This product will have several interactions between components. The sensors will send data, and the data will be populated into a database. A middle agent will then take the data and analyze it and make it more user friendly, and then the user interface will then take the prepared data and display it. A diagram is shown of the way this system will work in Section 6.0—System Features.

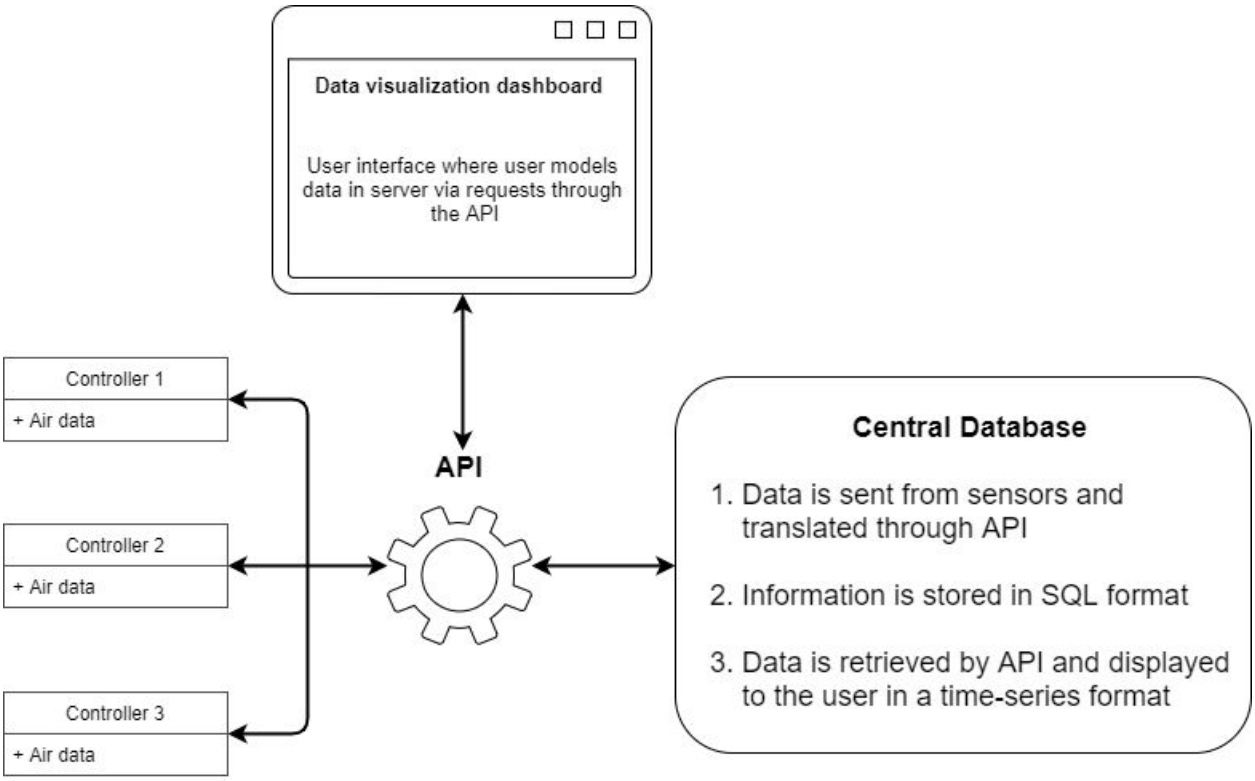
Messages will mainly be flowing out of the database into the UI, but some will be taken in the opposite direction, specifically when a citizen wants to connect a sensor to the network. This case is shown in section 2.1—User.

3.4 Communications

The main form of communication in this system is done over the internet. The sensors will be connected to the internet either via WiFi or a cell connection which will allow the data to be read into the database. A diagram of this model is shown below.



The system shall be comprised of several distinct functional requirements that outline the basic roles required for desired operation. They are detailed in a high-level orientation in the diagram below:



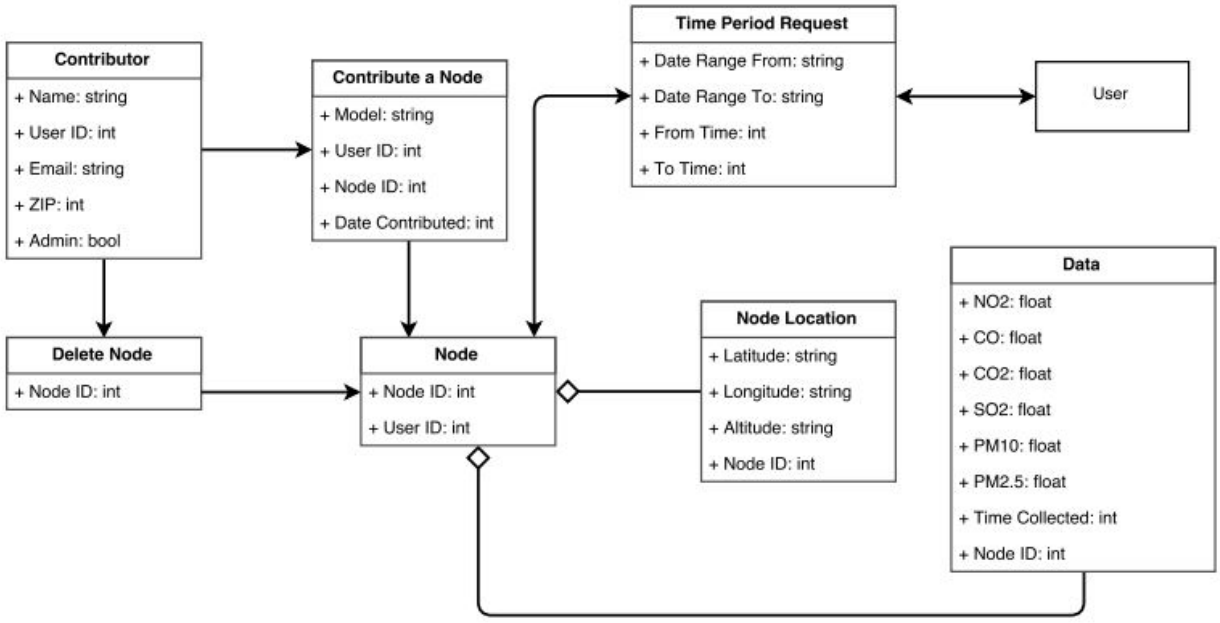
In order to keep the scope manageable at this stage, the overall vision of the system has been condensed into three primary functional requirements:

- A central time-series database
- A Middle API that translates data
- Data visualization dashboard

Later on, requirements will be broken down from these main three as needed. Additional requirements will be detailed as they become apparent.

4 Class Diagrams

The following depicts the different data that will be stored in the system and what the database might look like.

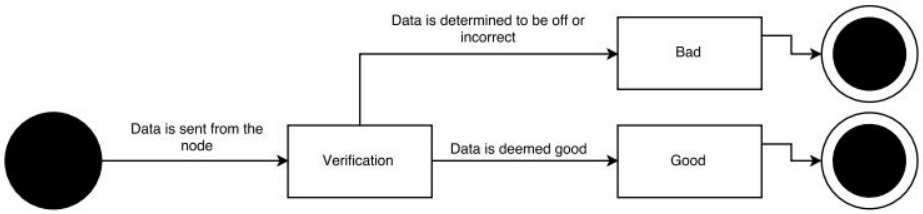


5 State Diagrams

The following diagrams depict items that were considered to be stateful. They include a calibration flag, a manager/admin flag, collecting data, and adding a node.

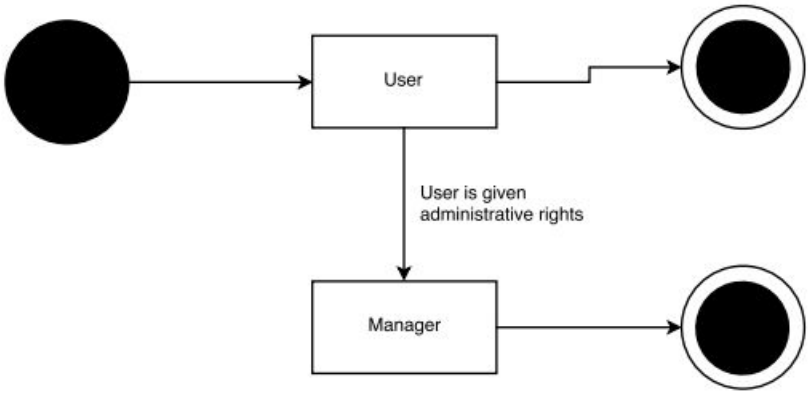
5.1 Calibration Flag

When a node provides consistent data that is shown to be significantly greater than or less than the values provided by other nearby nodes, it can be assumed that the node needs to be calibrated.



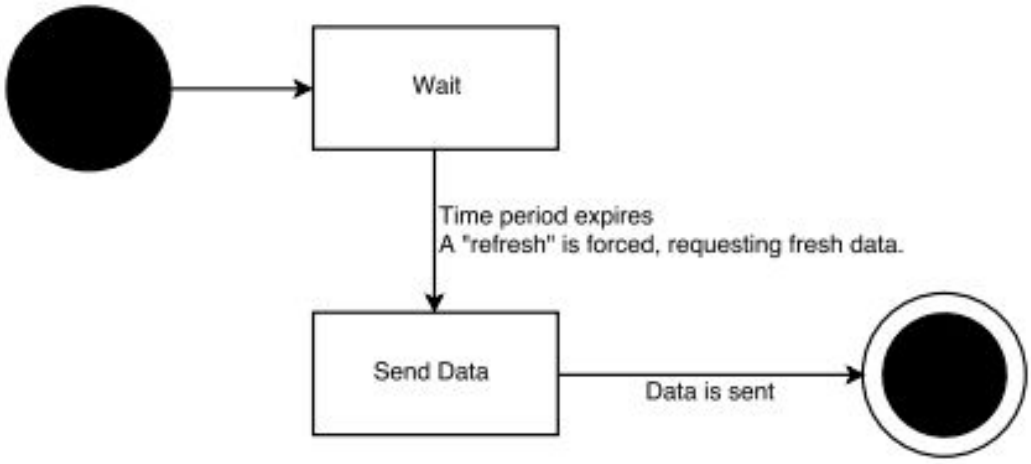
5.2 Manager/Admin Flag

Some users are considered managers or administrators of the system. It needs to be known that they have this superiority, and thus a flag is given.



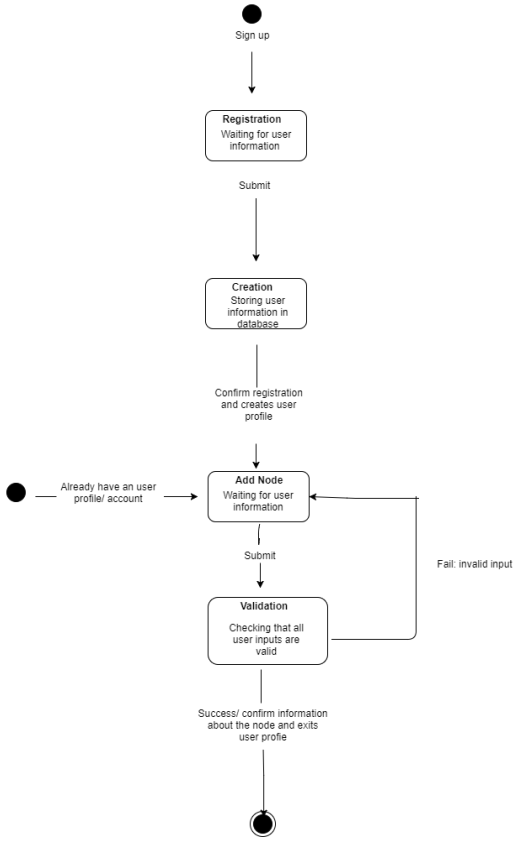
5.3 Collect Data

When the node collects data, this is stateful because the node only needs to sample the data and send that data to the server every five minutes. A diagram is shown below.



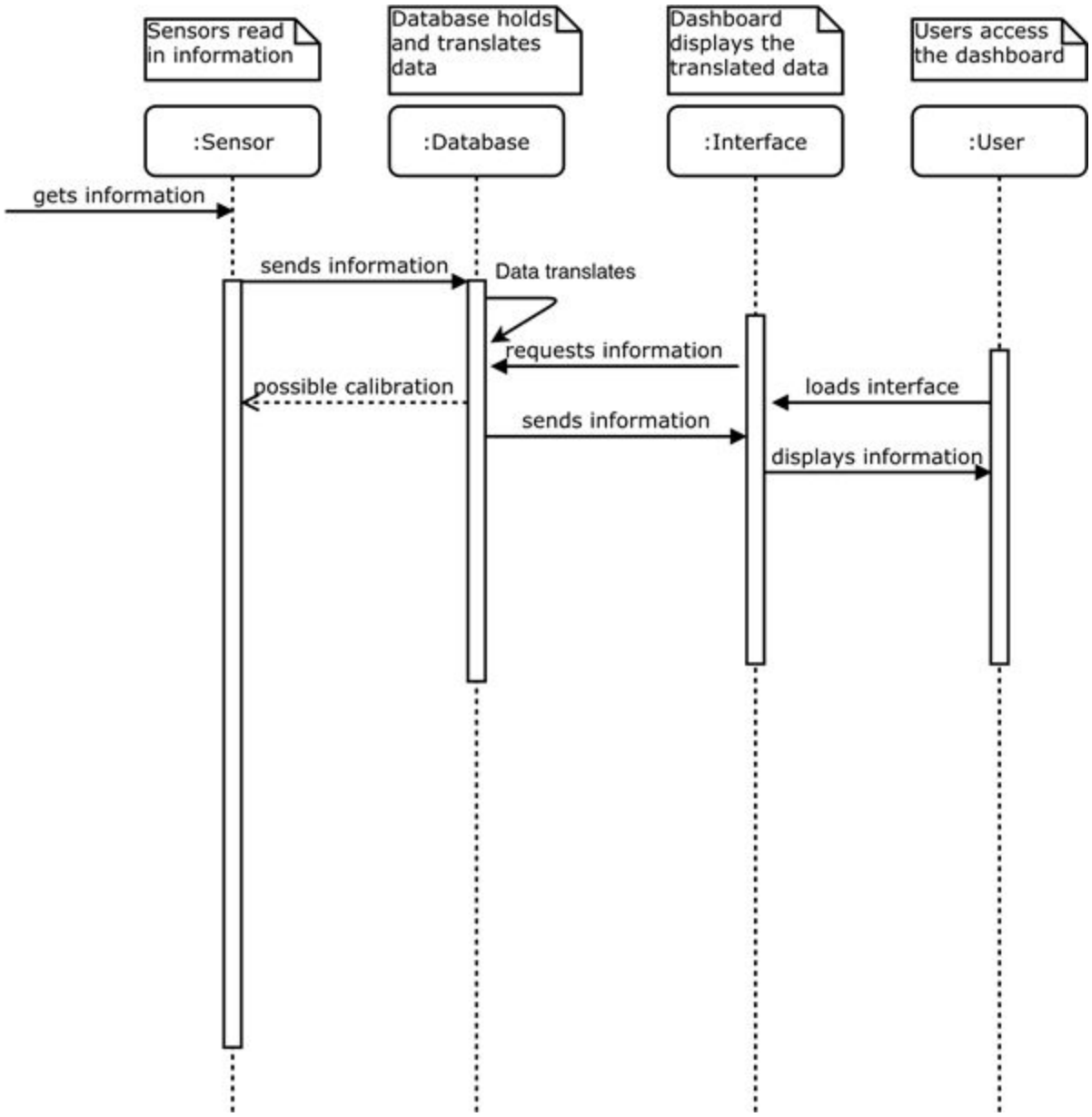
5.4 Adding a Node

The following is the process that a user goes through to add a node, and the states that are encountered.



6 Interaction Diagram

This depicts the different interactions that exist between the different components of the system.



CSCE 4925: Project Aero
Detailed Design

By: Alyssa Thurston, Breuna Riggins, James Sabetti, Travis Goral
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 1 December 2017
Status: Final Draft

Table of Contents

Revisions

Below is a list of any revisions made to this document.

Date	Description of Change Made	Person Making Change
11/7/2018	Draft Created	Breuna, Alyssa, James. Travis
11/23/2018	Draft Compiled	Breuna
11/24/2018	Document Edited	James
12/1/2018	Final Draft Complete	Breuna, Alyssa, James, Travis
2/13/2018	Final Draft Revisions- Reformatting, Editing of the functions	Alyssa
5/1/2018	Final Draft Revisions	Alyssa, Travis

1 Project Summary

The air quality in the City of Denton is notoriously bad. It's not a surprising fact given that there are two universities and multiple major highways and roadways that run through the city's limits. Worse still, there is only a single air quality sensor within city limits, and the next closest one is almost 15 miles away from the city.

The goal of this project is to create a network of air quality sensors to measure the air quality in the city of Denton so that citizens and city personnel alike can monitor air quality and take appropriate actions. The data obtained from the sensors is to be made public, so that air quality trends can be tracked all around the city. With these requirements met, the network would provide secure, real-time, and reliable data on the air quality in the city of Denton.

With the aim of creating a robust and open source of air quality data for the city of Denton, the Open Denton group has set out to establish a network of air quality sensors to gather and compile that data. In order to help achieve this goal, our group has been tasked with the following:

- Create a web client to display data in a highly readable manner.
- Create an API that records and displays air quality data in a readable manner
- Implement a scalable database to hold and model recorded data
- Integrate sensors and controllers that will read and relay data to the database over the internet
- Create documentation that allows community members to add sensors to the network

This document will serve as the guiding documentation for the implementation of this project.

2 Overall Description

For our product, our goal is to design a web based client, a database, and an API to get the data from one to the other. The web based client should be accessible, up-to-date and easy to use. The database should contain as few errors as possible, should be scalable, and presented in a time-series format. End users should be able to access the website at anytime from as many device types as possible. Equally important, if a user wants to contribute a node, this needs to be made as easily as possible as well.

2.1 Functions

The web client will perform the following functions:

- 2.1.0: Display air quality statistics in an easy to read, graphical format
- 2.1.1: Provide users with a map of all sensors within city limits
- 2.1.2: Provide a method of looking up historical data
- 2.1.3: Show what sensors are online and offline
- 2.1.4: If a sensor is offline, show when it was last online or last “seen”
- 2.1.5: Display data for a single sensor
- 2.1.6: Provide a help menu for users needing help with the system
- 2.1.7: Allow a user to login to an account
- 2.1.8: Allow a user to create an account
- 2.1.9: When logged in, show a map of where the user’s sensor’s are located
- 2.1.10: When logged in, show sensor information
- 2.1.11: When logged in, show statistics from the user’s sensor’s
- 2.1.12: Provide location data of a sensor, either via address or through coordinates.
- 2.1.13: Verify that users are able to create an account.
- 2.1.14: The map should use different icons for sensors that are online and offline
- 2.1.15: Users should be able to add sensors into the network
- 2.1.16: Users should be able to delete their sensors from the network

The database will perform the following functions and have the following qualities:

- 2.1.17: Store data from the sensor(s)
- 2.1.18: Accept user reported data
- 2.1.19: The system should accept data from any sensor, regardless of type.
- 2.1.20: The database should be presented in a time series format

Some other requirements of this project include:

- 2.1.21: It should be able to store any air quality value that a user is able to provide with their sensor
- 2.1.22: Verify that the data reported from the sensor is not a outlier, and somewhat matched the data reported from other sensors.
- 2.1.23: A large volume of users should be able to use this system without issue.
- 2.1.24: The system should be secure.
- 2.1.25: Raw data should also be made available for download

2.1.26: The system should refresh itself with new data pulled from the sensors every five minutes.
2.1.27: System should notify a user if the data that is being pulled is an outlier

2.2 Use Cases

This system is designed to be used to report the level of air pollution and the quality of the air dependent on human pollution, the environment, and weather circumstances, to be recorded and reported by both ourselves and other users.

2.3 Operating Environment

The operating environment for this project is presumably a server that houses all of the database information, that is then recorded and fed to a dashboard for users to read and interpret the statistics presented by the users.

2.4 User Documentation

Users will be primarily using this system in one of two ways: they will either be accessing the information via the dashboard (at which point, there would be no need for documentation), or they will be inputting the information recorded by their devices, at which point the system would need to document what the users have input into the system and verify that information to make sure that it is correct and accurate. All user inputs would need to be documented in order to keep the legitimacy of our statistics reported.

2.5 Assumptions and Dependencies

We are assuming, while building these requirements, than the information reported by the users would be reported by the device they have made and own, not information submitted by the user's hand. We are also assuming that our database can correctly store all of this information separate from the dashboard, and that the dashboard merely reads out the information to the user.

2.6 Project Constraints

We are constrained in terms of time and the resources given to us. We will have no control over the machine that will be given to us to record all of this information with, and we are constrained by a year time limit in order to accomplish everything we have set out to do. As a result, while this project should not push the year long time limit, it may lack the polish and finish that a fully realized system with a longer timeline would normally have.

3 External Interfaces

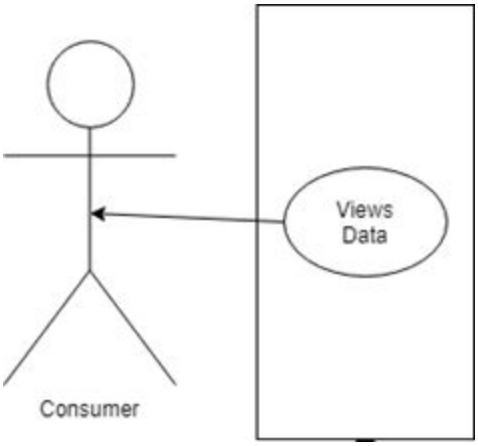
3.1 User

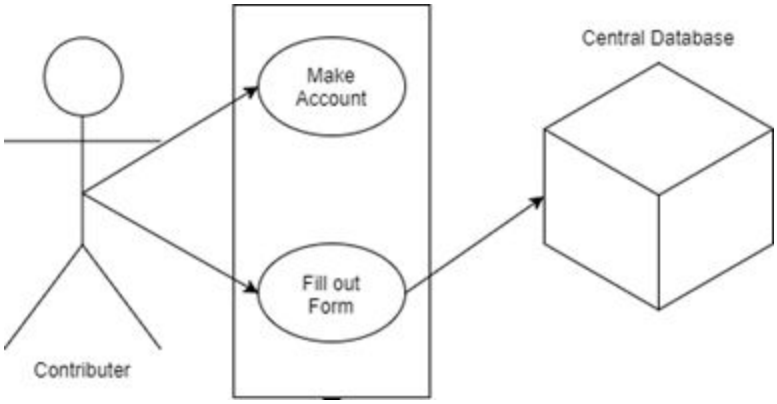
The user interface will be a responsive web based design, so that the project can be accessible from virtually any device with an internet connection. The readings should be shown graphically, using graphs, meters, and more, so users know what the levels are, if they're unhealthy or not, and can view trends over time.

As appropriate, charts from the Data Viz Project will be selected to display the data in a way that is friendly to users. Some of these may include an angular gauge, a pie chart, line graphs, grouped bar graphs, and more.

The user interface will be browser based, and should not be restricted to specific browsers or clients. It should adapt to fit both a desktop computer screen and a mobile screen. It should be available to use on all web browsers, such as Microsoft Edge, Mozilla Firefox, Google Chrome, and Mac's Safari.

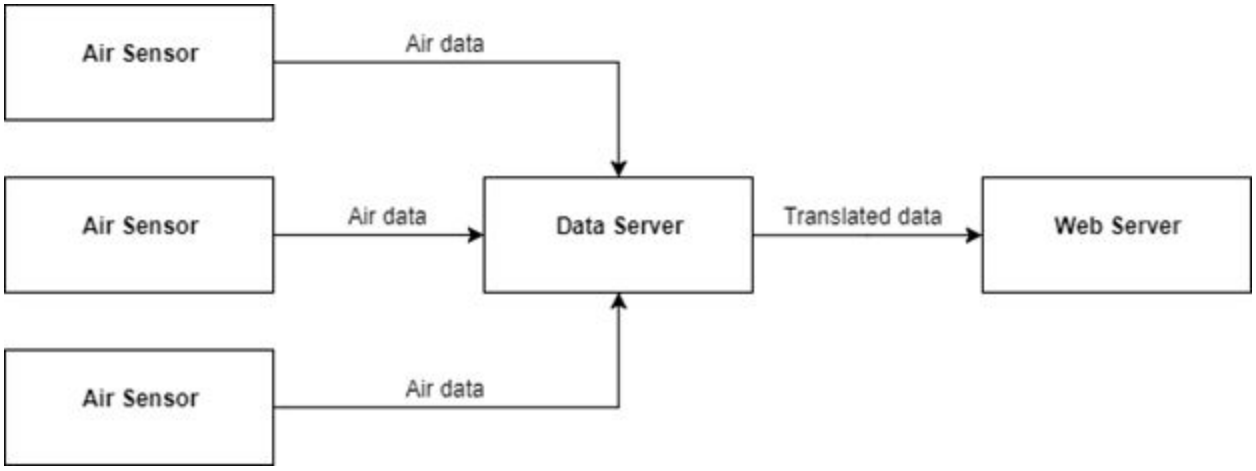
Two different user types were defined in meetings with the client- a consumer and contributor. The consumer simply views data on the UI, while the contributor will be able to create an account and fill out a form in order to contribute a sensor to the network. These functions are shown in the following diagrams.





3.2 Hardware

The project should be able to load data into a database from the air quality sensors, then take the data from the database and represent it in a human readable format. A diagram showing these details is shown on the next page.



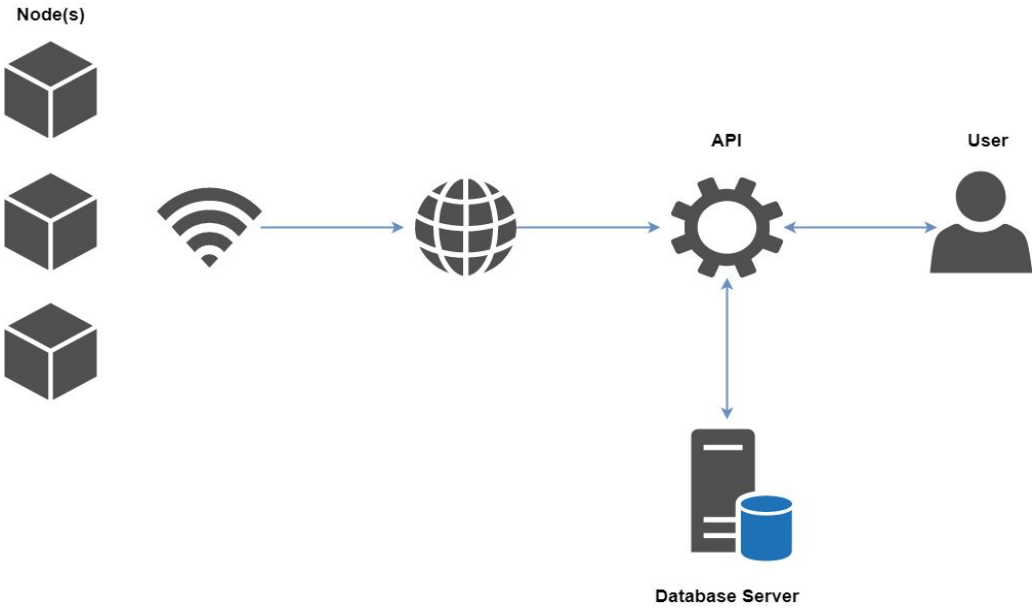
3.3 Software

This product will have several interactions between components. The sensors will send data, and the data will be populated into a database. A middle agent will then take the data and analyze it and make it more user friendly, and then the user interface will then take the prepared data and display it. A diagram is shown of the way this system will work in Section 6.0—System Features.

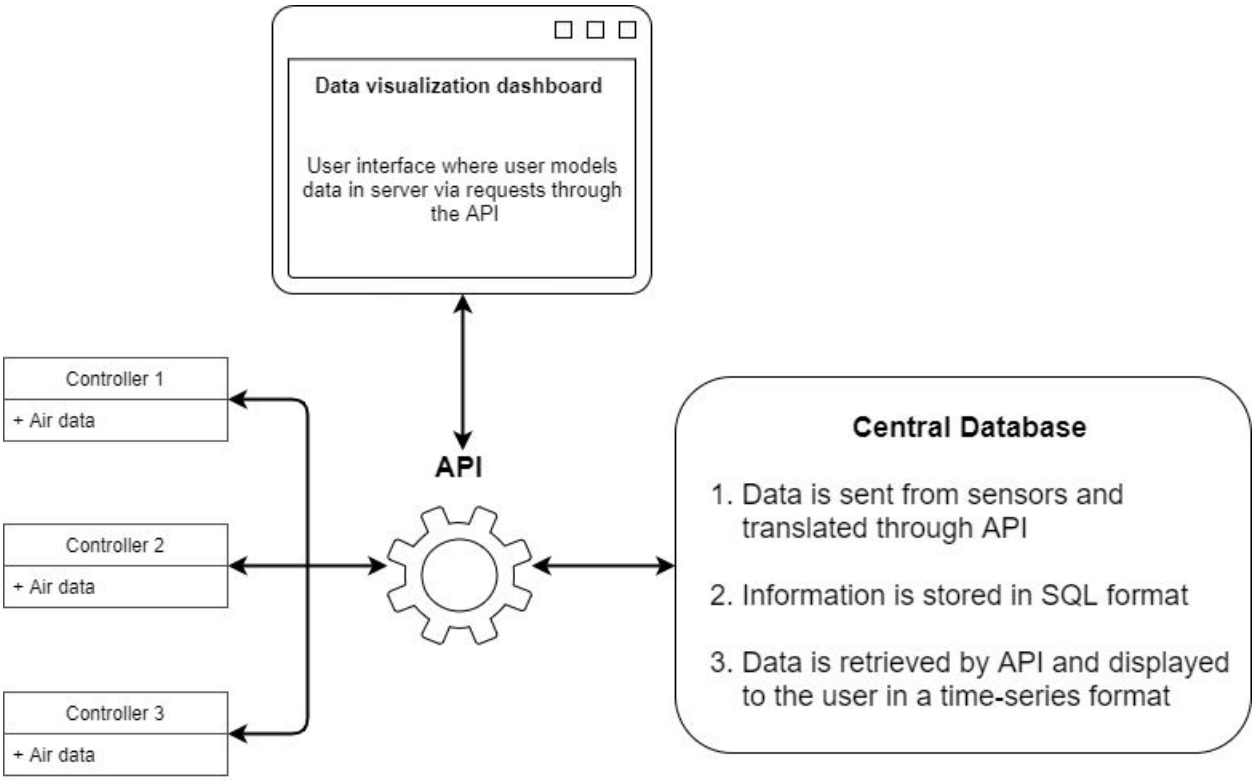
Messages will mainly be flowing out of the database into the UI, but some will be taken in the opposite direction, specifically when a citizen wants to connect a sensor to the network. This case is shown in section 2.1—User.

3.4 Communications

The main form of communication in this system is done over the internet. The sensors will be connected to the internet either via WiFi or a cell connection which will allow the data to be read into the database. A diagram of this model is shown below.



The system shall be comprised of several distinct functional requirements that outline the basic roles required for desired operation. They are detailed in a high-level orientation in the diagram below:



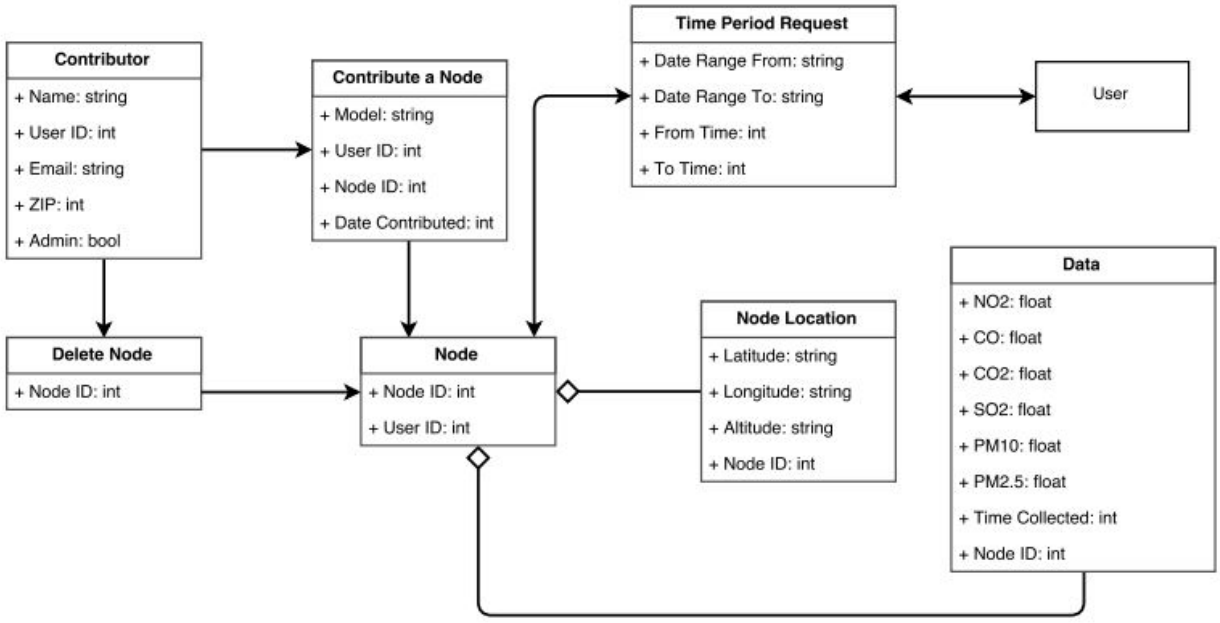
In order to keep the scope manageable at this stage, the overall vision of the system has been condensed into three primary functional requirements:

- A central time-series database
- A Middle API that translates data
- Data visualization dashboard

Later on, requirements will be broken down from these main three as needed. Additional requirements will be detailed as they become apparent.

4 Class Diagrams

The following depicts the different data that will be stored in the system and what the database might look like.

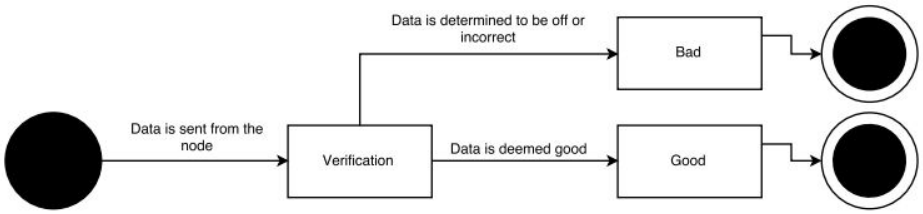


5 State Diagrams

The following diagrams depict items that were considered to be stateful. They include a calibration flag, a manager/admin flag, collecting data, and adding a node.

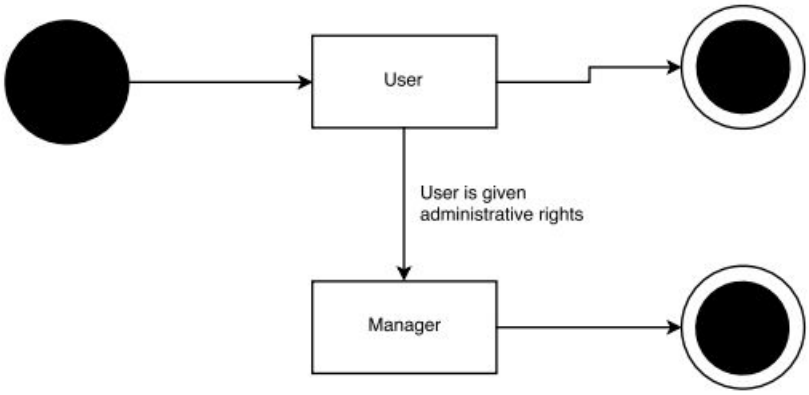
5.1 Calibration Flag

When a node provides consistent data that is shown to be significantly greater than or less than the values provided by other nearby nodes, it can be assumed that the node needs to be calibrated.



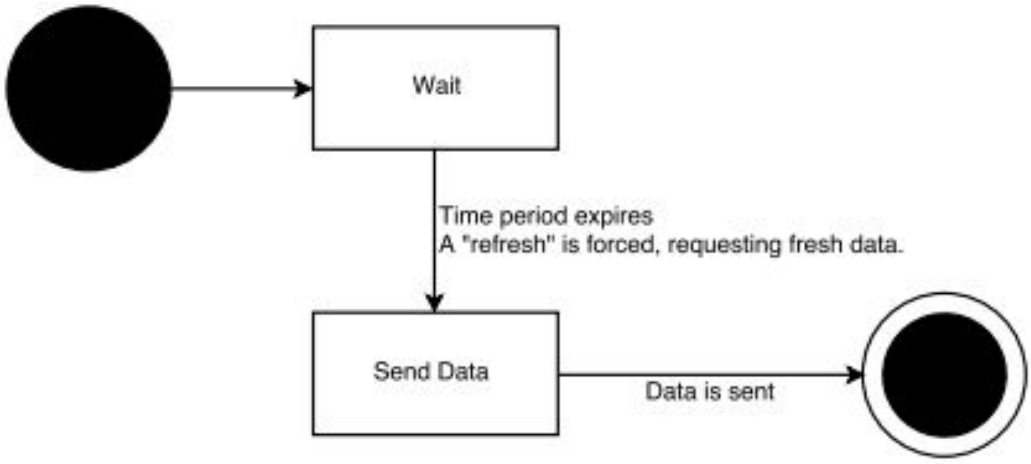
5.2 Manager/Admin Flag

Some users are considered managers or administrators of the system. It needs to be known that they have this superiority, and thus a flag is given.



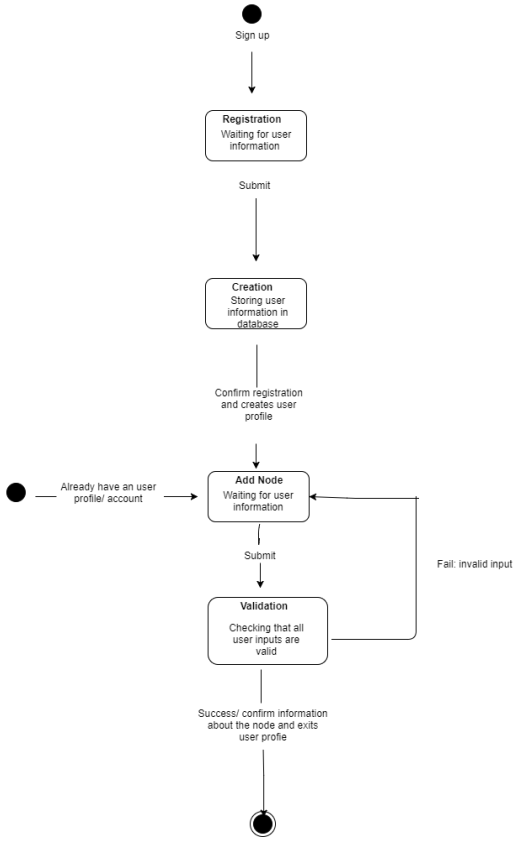
5.3 Collect Data

When the node collects data, this is stateful because the node only needs to sample the data and send that data to the server every five minutes. A diagram is shown below.



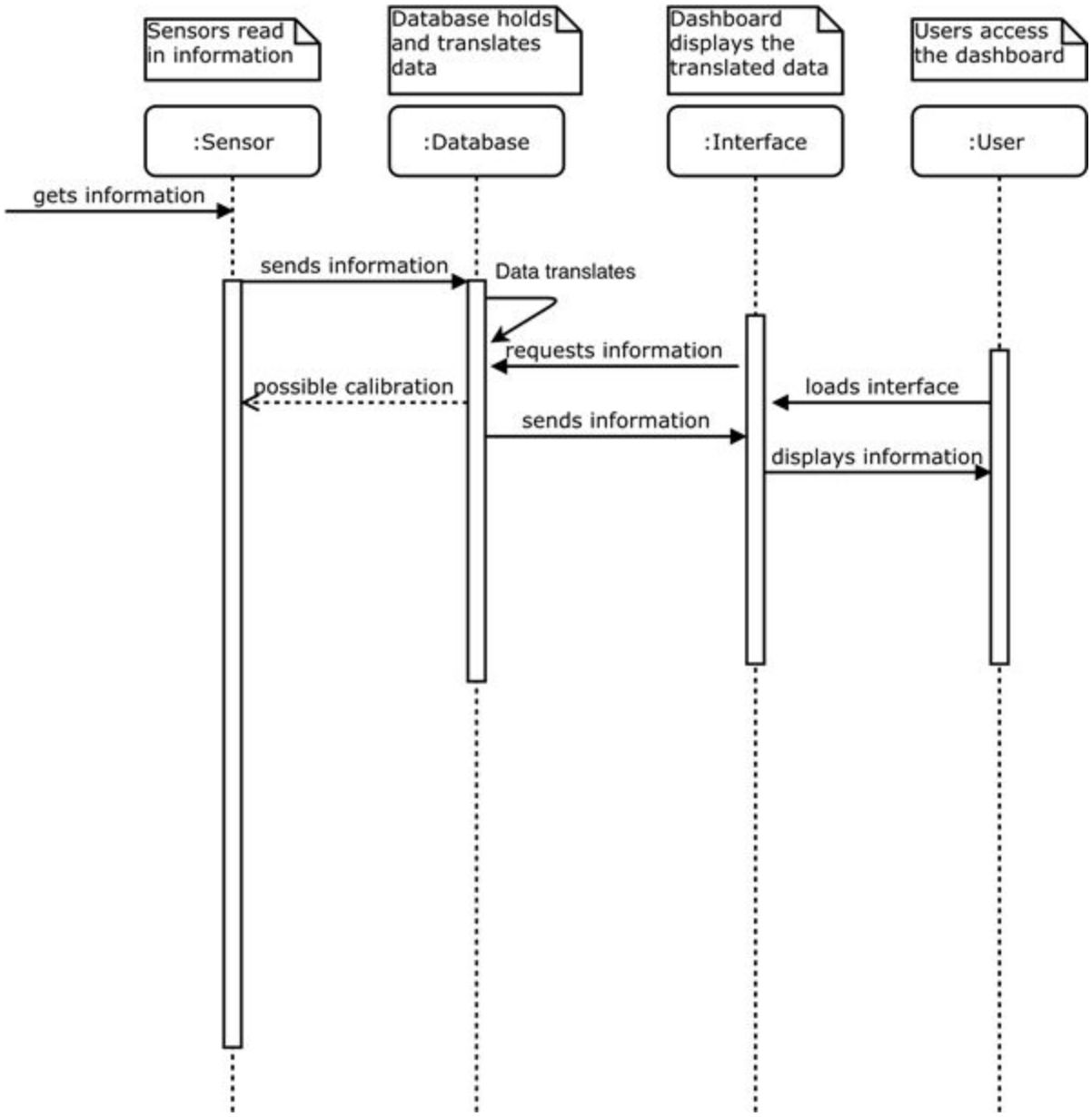
5.4 Adding a Node

The following is the process that a user goes through to add a node, and the states that are encountered.



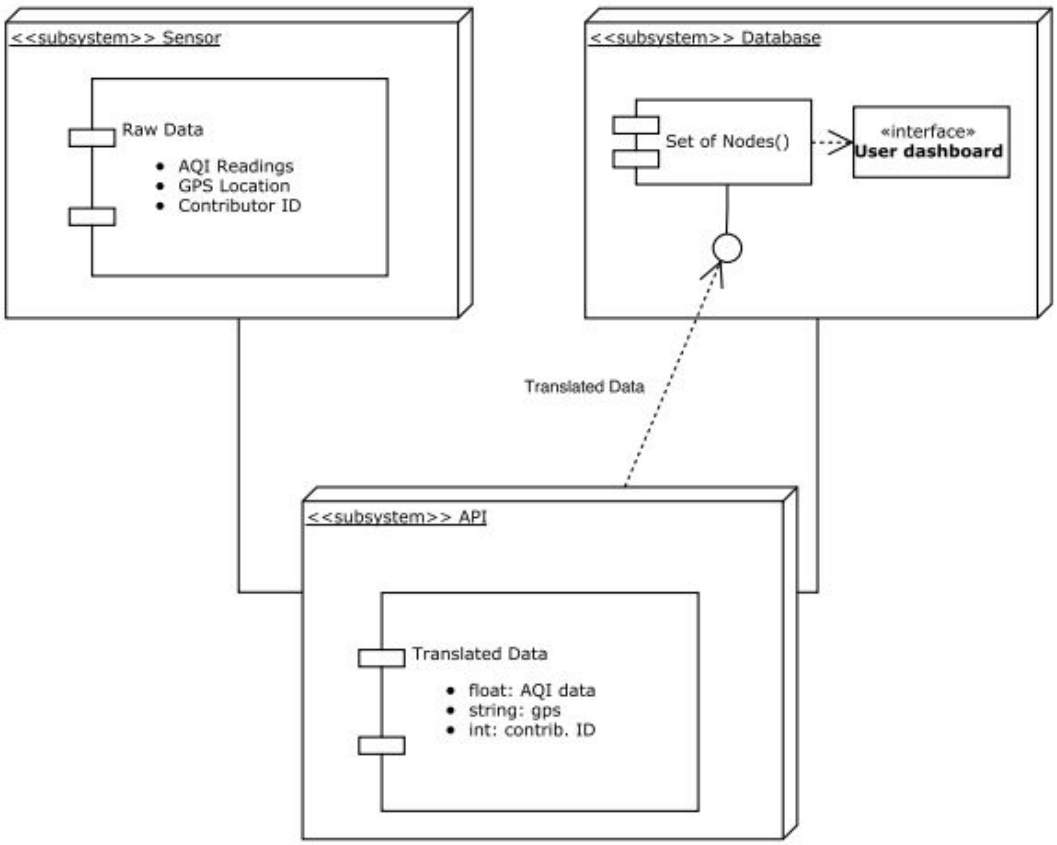
6 Interaction Diagram

This depicts the different interactions that exist between the different components of the system.



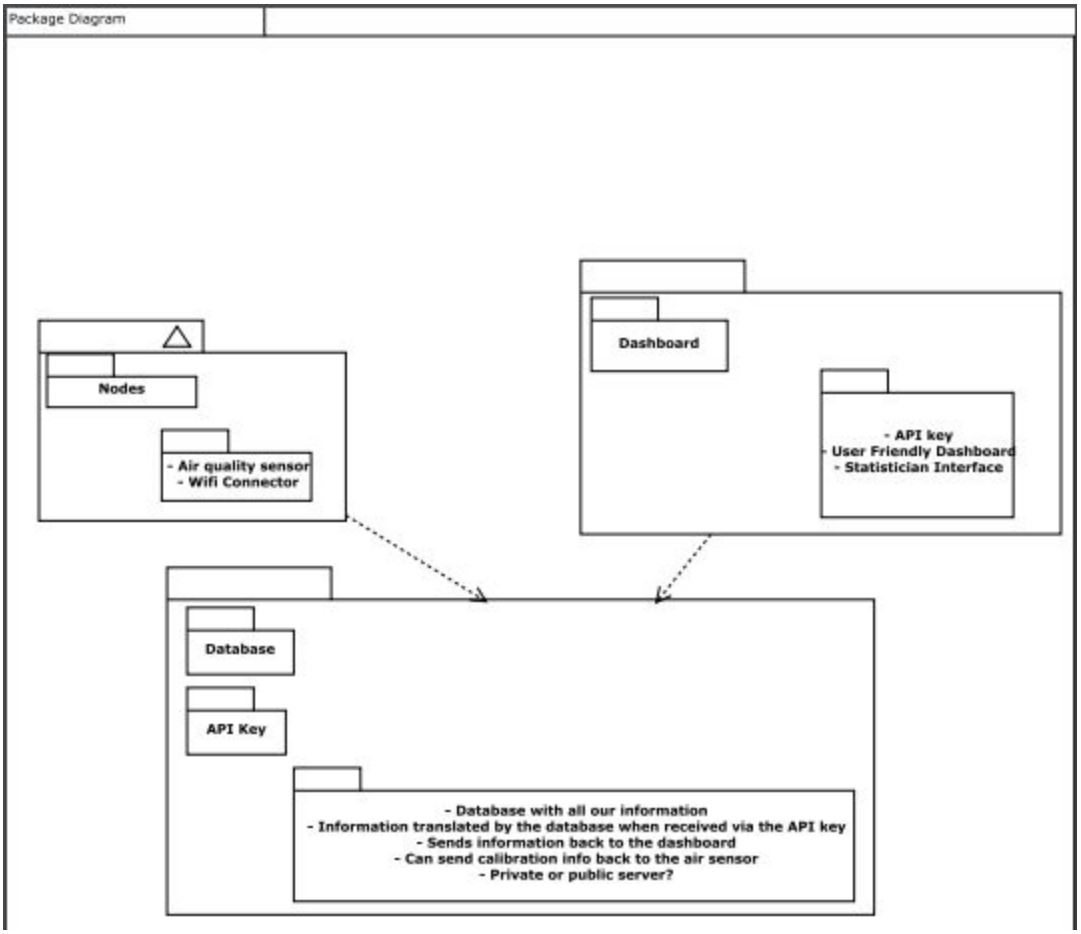
7 Component Diagram

The following depicts the location of different system components.



8 Package Diagrams

The following diagram depicts where each component “lives” on a system.



Project Aero

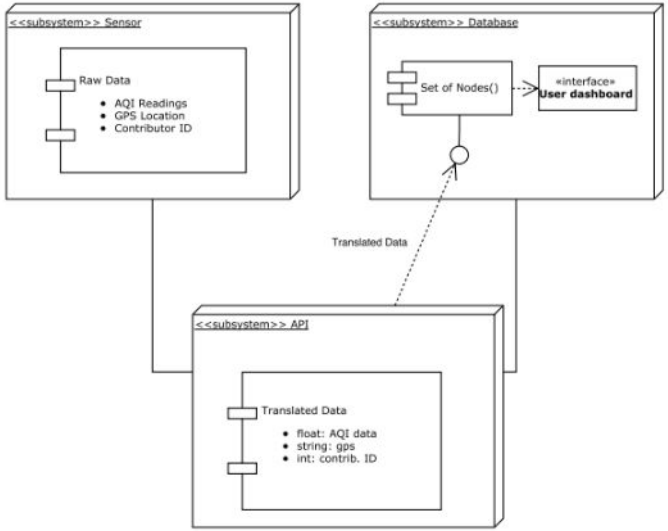
Alyssa Thurston, Breuna Riggins, James Sabetti, Travis Goral

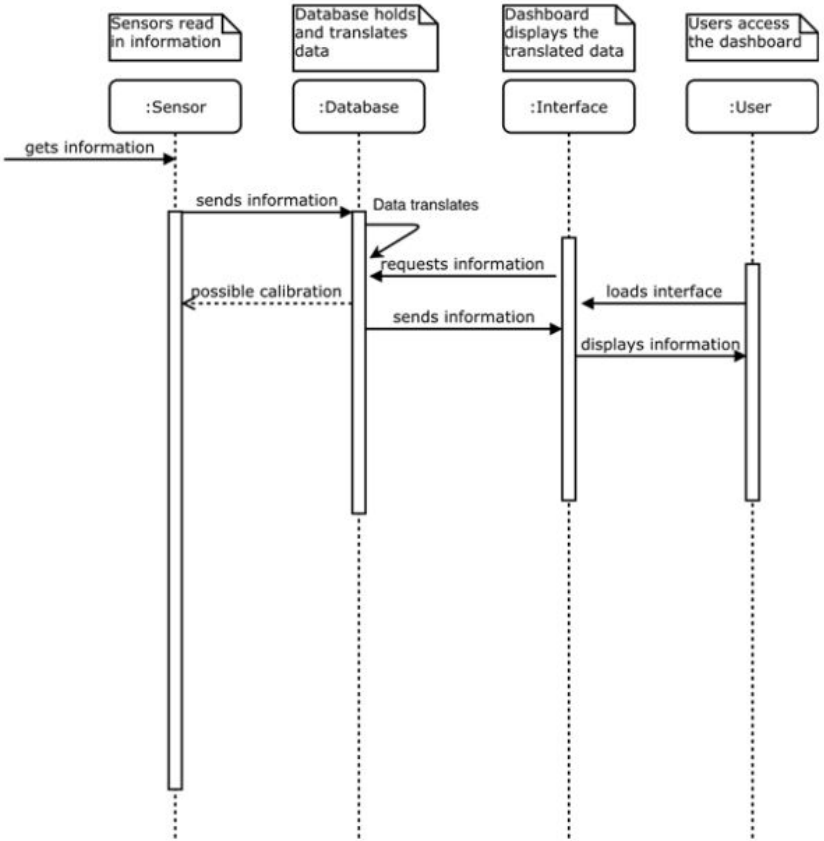
Design Overview

Hard and Challenging things:

- Accepts sensors of all types
- Finding ways to throw out bad data
- Registering contributors

Design Overview (Cont.)





Tools and Platforms

Ruby on Rails

- Object-oriented
- Test-driven development
- Easier on-boarding process
- Highly scalable
- Excellent support and documentation
- Secure

TimescaleDB

- Built for time-series data
- Uses PostgreSQL
- Advanced query optimization
- Everything is run on a LAMP stack

Challenges

Outstanding

Learning Ruby for the design of our system

Calibration?

Using API in the aspect our client wants

Solved

How we were going to get our sensors (engineering, privately, or publicly)

The depths and levels of the system the client wanted us to implement

How we wanted to display our information to our users

Plans for the Break

Research:

- Sensors
- Ruby on Rails
- PostgresSQL

Goals:

- Have a wireframe of the dashboard
- Setup database
- Setup server

Meetings:

- Have two meetings via google hangouts

Conclusions

Start of Spring Semester

- Hoping to have a product prototype based on original requirements.
- Have a fully working system before Spring break.

Questions???

CSCE 4925: Project Aero
Test Plan

By: Alyssa Thurston, Breuna Riggins, James Sabetti, Travis Goral
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 12 February 2018
Status: Final Draft

Table of Contents

Revisions

Below is a list of any revisions made to this document.

Date	Description of Change Made	Person Making Change
2/12/2018	Document Created	Breuna, Alyssa
2/13/2018	Document Revision	James
2/15/2018	Final Draft Edits	Breuna, Alyssa, Travis
2/16/2018	Final Draft Revision	James

1 Introduction

The air quality in the City of Denton is notoriously bad. It's not a surprising fact given that there are two universities and multiple major highways and roadways that run through the city's limits. Worse still, there is only a single air quality sensor within city limits, and the next closest one is almost 15 miles away from the city.

The goal of this project is to create a network of air quality sensors to measure the air quality in the city of Denton so that citizens and city personnel alike can monitor air quality and take appropriate actions. The data obtained from the sensors is to be made public, so that air quality trends can be tracked all around the city. With these requirements met, the network would provide secure, real-time, and reliable data on the air quality in the city of Denton.

With the aim of creating a robust and open source of air quality data for the city of Denton, the Open Denton group has set out to establish a network of air quality sensors to gather and compile that data. In order to help achieve this goal, our group has been tasked with the following:

- Create a web client to display data in a highly readable manner.
- Create an API that records and displays air quality data in a readable manner
- Implement a scalable database to hold and model recorded data
- Integrate sensors and controllers that will read and relay data to the database over the internet
- Create documentation that allows community members to add sensors to the network

This document will outline the test plan to test this system when it is constructed, and serve as a layout for performing the necessary testing before the system is given to the client.

1.1 Purpose

The purpose of this document is to summarize the action plan and to formulate a plan to test the completed system. This plan will help illustrate client and personnel functionality, and prove that both the sensors and the dashboard web application performs correctly and that it satisfies every requirements of the system.

1.2 Scope

This test plan will test all the features outlined in 2.2, Features to be Tested. Each feature that we test needs to pass with minimal or no errors or bugs encountered. Any unfixable bugs encountered need to be minimal at best, and should not interfere with the integrity of the system.

Each test to be ran will be ran under different environments, using a combination of different operating systems and different browsers. Each operating system will be documented with its name and the last installed update package. For each browser tested, we will document the version of that browser.

We are constrained on the number of Operating Systems and Browsers that can be used because we don't have very much money to work with. Some operating systems and browsers will not be tested because of this. An extensive list of the environments to be used will be outlined in Section 3.1- Environmental Needs.

1.3 List of Definitions and Abbreviations

- D.A.M.N. - Denton Air Monitoring Network
- Node - Air Quality Sensor
- AQI- Air Quality Index

1.4 Applicable Documents

- Detailed Design document

1.5 Overview

This Test Plan covers everything that we will test in Project D.A.M.N. It includes the features to be tested, the features not to be tested, the environments in which the platform was tested, a few risks that were identified in bringing this test plan to fruition. Upon successful execution of each item contained in this document, we should be able to deliver a fully functioning system that contains minimal runtime error and other issues. Overall, this process should take us no more than one month to accomplish and bring everything in fruition.

To accomplish all listed on this plan, this project will go through three testings trials to ensure everything is working correctly. Following the test trials will be a revision period to fix bugs and other issues encountered during the testing process.

2 Test Plan

This chapter will focus on the specifics of the test plan, including:

- Test Items
- Features to be tested
- Features not to be tested
- Testing approach and process
- Item pass/fail criteria
- Test deliverables
- Test tasks

2.1 Test Items

For each feature that is tested, we will be looking for full functionality. More specifically, does the feature work, does it throw errors, and if it does throw errors, how human friendly are these errors? Further, for user input, does the system allow for mistakes to be inputted into the system, or does it catch these errors? Lastly, we will ensure that for any errors encountered, the system can recover quickly and easily from the errors.

2.2 Features to be Tested

The features that will be tested prior to delivery include the following items, which are specified in the requirements specifications document:

Feature ID Number	Test ID Number	Description of Test
1.3.1.1	01	Provide users with a map of all sensors within city limits
1.3.1.2	02	Provide a method of looking up historical data
1.3.1.3	03	Show what sensors are online and offline
1.3.1.4	04	If a sensor is offline, show when it was last online or last “seen”
1.3.1.5	05	Display data for a single sensor
1.3.1.7	06	Allow a user to login to an account
1.3.1.8	07	Allow a user to create an account

1.3.1.9	08	When logged in, show a map of where the user's sensor's are located
1.3.1.10	09	When logged in, show sensor information
1.3.1.11	10	When logged in, show statistics from the user's sensor's
1.3.1.13	11	Verify that users are able to create an account
1.3.1.14	12	The map should use different icons for sensors that are online and offline
1.3.1.15	13	Add new sensors to the network
1.3.1.16	14	Delete sensors from the network
1.3.1.17	15	Store data from the sensor(s)
1.3.1.18	16	Accept user reported data
1.3.1.19	17	The system should accept data from any sensor, regardless of type.
1.3.1.21	18	It should be able to store any air quality value that a user is able to provide with their sensor
1.3.1.22	19	Verify that the data reported from the sensor is not an outlier, and somewhat matched the data reported from other sensors.
1.3.1.24	20	The system should be secure (see section 2.3- Features not to be Tested)
1.3.1.25	21	Raw data should be made available for download
1.3.1.26	22	The system should refresh itself with new data pulled from the sensors every five minutes.
1.3.1.27	23	System should notify a user if the data that is being pulled is an outlier

2.3 Features not to be Tested

The features that will not be tested prior to delivery include the following:

Feature ID Number	Feature Description	Reason for not Testing
1.3.1.0	Display air quality statistics in an easy to read, graphical format	Subjective to feedback from users, not suitable for the data that we aim to gather
1.3.1.6	Provide a help menu for users needing help with the system	There will be a help menu available, but the content within the help menu will not be tested as this would require surveys from humans and the data gathered from this would not provide meaningful test result data.
1.3.1.12	Provide location data of a sensor, either via address or through coordinates	This is something that has to be gathered regardless in order for the map system to work, therefore testing it would be nonsensical.
1.3.1.20	The database should be presented in a time series format	This feature is more a requirement than a feature that needs to be tested.
1.3.1.23	A large volume of users should be able to use this system without issue	We don't have a large pool of testers to ensure that this works. We're also limited to the power of the server that was provided to us.
1.3.1.24	The system should be secure.	We don't have the funds to run a thorough test of this, but we will do basic pen-testing.

2.4 Testing Approach & Process

There is just one rule and that is if at any point during the testing trials the tested item fails in any category, the test is deemed as a failure. We will rerun testing after revision when an issue occurs or when the tested item fails in any category.

In order to test both the dashboard and the sensors, we require the following to run the tests:

- 2.4.7- A computer or virtual machine with Windows 7
- 2.4.8- A computer or virtual machine with Windows 8.1
- 2.4.9- A computer or virtual machine with Windows 10
- 2.4.10- A Mac
- 2.4.11- Minimum of three Android Devices
- 2.4.12- Minimum of one iPhone

2.4.13- Sensors
2.4.43- VirtualBox or similar virtual machine software

Each software listed above should be using the latest version available. The version of each software listed will be made available with the test deliverable. We will run the tests using the following configurations:

2.4.14- Internet Explorer, Windows 7
2.4.15- Safari, Windows 7
2.4.16- Mozilla Firefox, Windows 7
2.4.17- Google Chrome, Windows 7
2.4.18- Opera, Windows 7
2.4.19- Internet Explorer, Windows 8.1
2.4.20- Safari, Windows 8.1
2.4.21- Mozilla Firefox, Windows 8.1
2.4.22- Google Chrome, Windows 8.1
2.4.23- Opera, Windows 8.1
2.4.24- Internet Explorer, Windows 10
2.4.25- Microsoft Edge, Windows 10
2.4.26- Safari, Windows 10
2.4.27- Mozilla Firefox, Windows 10
2.4.28- Google Chrome, Windows 10
2.4.29- Opera, Windows 10
2.4.30- Mac OS
2.4.31- Android Device 1, Default Browser Available
2.4.32- Android Device 1, Google Chrome
2.4.33- Android Device 2, Default Browser Available
2.4.34- Android Device 2, Google Chrome
2.4.35- Android Device 3, Default Browser Available
2.4.36- Android Device 3, Google Chrome
2.4.37- iPhone, Safari
2.4.38- iPhone, Google Chrome

The data that we will collect collect from testing includes:

2.4.39- Did the feature work as expected?
2.4.40- If any bugs were encountered, how many bugs were encountered, what were they?
2.4.41- How much data were we able to input?
2.4.42- How many user accounts can we create?

To ensure that both the dashboard and sensors are working and able to gather and display information correctly, the sensor and dashboard will undergo three intensive testing trials. The first trial being that the dashboard works from the dummy data provided and the other two from the live data provided by the sensors.

The process which we will test the system involves identifying the feature to be tested, testing it, and documenting the results of the test, whether the result is expected or not. Testers will input all data in a table similar to the following example:

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 2/13/2018	Test Trial Number: 1	Configuration ID: 2.4.24
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

We have the appropriate amount of people to perform these tests to the extent that we want to run them, and expect only time to be the only constraint on this procedure. We anticipate that we will have three testing cycles, each cycle lasting for four days.

2.5 Item Pass/Fail Criteria

To pass and complete the test, all of the test requirements should be successfully met. Any outstanding issues remaining at the completion of this testing process should be minimal, and should not compromise the integrity of the system.

In order for an item to pass, it must succeed all usability tests prescribed in this document. If the item fails in a single category, it fails the entire test and must therefore be fixed.

When an item passes the test, it should still be tested in subsequent test trials, in case revisions of the system introduced more bugs into the system.

2.6 Test Deliverables

Here are the items that will be delivered upon completion of this test:

2.6.1- Test plan

- This document

2.6.2- Test Results

- Software versions of all software used
- Keep records of tests that were conducted
- Includes any issues identified during testing
- Summarizes the conclusion of testing

2.7 Test Tasks

For each feature to be tested, there are a set of tasks that must also be completed. Briefly, these include testing the item under all applicable environments, ensuring that all potential user errors are caught and handled appropriately, and ensuring the test is documented appropriately.

Each area of the system that allows for user input and interaction needs to be thoroughly tested for functionality and for error handling. If errors are not handled appropriately, the test will fail.

Lastly, the test needs to be documented, regardless of if it passes or fails. Notes regarding why the test passed or failed also needs to be documented. If the test fails, revisions to the system may be necessary, and the test needs to be reran, this too should be documented.

3 Execution

3.1 Environmental Needs

D.A.M.N should be tested on multiple operating systems and browsers and a combination of the two. Configuration information and identification can be found in Section 2.4, Testing Approach and Process. The platforms and software we will use include the following:

- Internet Explorer
- Microsoft Edge
- Safari
- Mozilla Firefox
- Google Chrome
- Opera
- Windows 7
- Windows 8.1
- Windows 10
- Mac OS
- iOS
- Android

3.2 Responsibilities

Each team member will have a role in the testing process. This is as follows:

- Alyssa: Overseer of the testing process, will assist with other tasks on an as needed basis
- Breuna: Tester- one who will perform the tests and document the results
- Travis: Revisionist- one who will work on identifying what causes a failed test and apply fixes as necessary
- James: Tester- one who will perform the tests and document the results

If there is an area in which the team needs more help, Alyssa will be responsible for filling in that gap. Her primary position, however, will be to ensure that this test plan is executed on schedule, and as specified through this document

3.3 Schedule

Task Name	Duration	Start	Finish
Prototype of System	24 Days	January 30, 2018	March 2, 2018
Prototype of Sensor	24 Days	January 30, 2018	March 2, 2018
Test Round 1	4 Days	March 3, 2018	March 7, 2018
Revisions	8 Days	March 8, 2018	March 17, 2018
Testing Round 2	5 Days	March 18, 2018	March 22, 2018
Final Revisions	6 Days	March 22, 2018	March 29, 2018
Final Testing	4 Days	March 30, 2018	April 3, 2018

3.4 Risks and Contingencies

The risks identified, as they pertain to this document, include:

3.4.1- Delays in the development process

- Where possible, the amount of time provided for testing will be extend out, to allow for proper testing. While time can only be extended so far, a “crunch time” period may have to be enforced.

3.4.2- Changes to the original requirements

- Meetings with the client and the team should help avoid these, though they do come up and should be dealt with as they are identified

3.4.3- Exceptional numbers of test failures

- To be resolved by fixing the systems, or in dire circumstances the requirements should be adjusted.

3.4.4- Lack of experience

- Outside help through other classmates and through resources provided by our client can be consulted.

3.5 Approvals

The following people will be in charge of approving this test plan and other items:

- Test Level Plan - Fantastic Four Group
- Integration Level Plan - Fantastic Four Group
- System Level Plan - Fantastic Four Group, Dan Minshew, and Kyle Taylor
- Master Level Plan - Dan Minshew and Kyle Taylor

CSCE 4925: Project Aero
Final Acceptance Test Procedure

By: Alyssa Thurston, Breuna Riggins, James Sabetti, Travis Goral
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 1 May 2018

Table of Contents

Revisions

Below is a list of any revisions made to this document.

Date	Description of Change Made	Person Making Change
5/1/2018	Document Created	Breuna, Alyssa, Travis, James
5/1/2018	Document Revision and Editing	Breuna, Alyssa, Travis, James

1 Test Configurations

Below are a list of the configurations used to perform the tests. In the testing results, these correspond to the Configuration ID.

- 2.4.14- Internet Explorer, Windows 7
- 2.4.15- Safari, Windows 7
- 2.4.16- Mozilla Firefox, Windows 7
- 2.4.17- Google Chrome, Windows 7
- 2.4.18- Opera, Windows 7
- 2.4.19- Internet Explorer, Windows 8.1
- 2.4.20- Safari, Windows 8.1
- 2.4.21- Mozilla Firefox, Windows 8.1
- 2.4.22- Google Chrome, Windows 8.1
- 2.4.23- Opera, Windows 8.1
- 2.4.24- Internet Explorer, Windows 10
- 2.4.25- Microsoft Edge, Windows 10
- 2.4.26- Safari, Windows 10
- 2.4.27- Mozilla Firefox, Windows 10
- 2.4.28- Google Chrome, Windows 10
- 2.4.29- Opera, Windows 10
- 2.4.31- Android Device 1, Default Browser Available
- 2.4.32- Android Device 1, Google Chrome
- 2.4.33- Android Device 2, Default Browser Available
- 2.4.34- Android Device 2, Google Chrome
- 2.4.35- Android Device 3, Default Browser Available
- 2.4.36- Android Device 3, Google Chrome
- 2.4.37- iPhone, Safari
- 2.4.38- iPhone, Google Chrome

2 Test Results

Below are the test results of all the features tested.

Feature 1.3.1.1

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.14
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.15
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.16
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number:	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.17

01			
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.18
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.19
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.20
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.21

Expected Result: A map shown with sensors on a map
Actual Result: The map showed all sensors in the system on a map
Test Result: Pass

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.22
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.23
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.24
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.25
Expected Result: A map shown with sensors on a map			

Actual Result: The map showed all sensors in the system on a map
Test Result: Pass

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.26
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.27
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.28
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.29
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			

Test Result: Pass

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.31
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.32
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.33
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.34
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.35
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.36
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.37
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.1: Provide users with a map of all sensors within city limits			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.38
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			
Test Result: Pass			

Feature 1.3.1.5

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 05	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.14
Expected Result: Display data for a single sensor			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.15
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.16
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.17
Expected Result: A map shown with sensors on a map			
Actual Result: The map showed all sensors in the system on a map			

Test Result: Pass

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.18
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.19
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.20
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.21
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.22
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.23
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.24
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.25
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.26
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.27
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.28
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.29
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor

Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.31
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.32
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.33
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.34
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number:	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.35

01			
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.36
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.37
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

Feature 1.3.1.5: Display Data for a single sensor			
Test ID Number: 01	Date: 5/1/2018	Test Trial Number: 1	Configuration ID: 2.4.38
Expected Result: A map shown with sensors on a map			
Actual Result: Data is displayed for a single (test) sensor.			
Test Result: Pass			

3 Features Not Tested

These features were not tested because the code for them could not be properly implemented.

1.3.1.2- Provide a method of looking up historical data
1.3.1.3- Show what sensors are online and offline
1.3.1.4- If a sensor is offline, show when it was last online or last “seen”
1.3.1.7- Allow a user to login to an account
1.3.1.8- Allow a user to create an account
1.3.1.9- When logged in, show a map of where the user’s sensor’s are located.
1.3.1.10- When logged in, show sensor information
1.3.1.11- When logged in, show statistics from the user’s sensor’s
1.3.1.13- Verify that users are able to create an account
1.3.1.14- The map should use different icons for sensors that are online and offline
1.3.1.15- Add new sensors to the network
1.3.1.16- Delete sensors from the network
1.3.1.17- Store data from the sensors
1.3.1.18- Accept user reported data.
1.3.1.19- The system should accept data from any sensor, regardless of type
1.3.1.21- The system should be able to store any air quality value that a user is able to provide with their sensor
1.3.1.22- Verify that the data reported from the sensor is not an outlier, and somewhat matched the data reported from other sensors.
1.3.1.25- Raw data should be made available for download
1.3.1.26- The system should refresh itself with new data pulled from the sensors every five minutes.
1.3.1.27- System should notify a user if the data that is being pulled is an outlier

These features weren’t planned on being tested.

1.3.1.0- Display air quality statistics in an easy to read, graphical format.
1.3.1.6- Provide a help menu for users needing help with the system.
1.3.1.12- Provide location data of a sensor, either via address or through coordinates.
1.3.1.20- The database should be presented in a time series format
1.3.1.23- A large volume of users should be able to use this system without issue.
1.3.1.24- THe system should be secure.

CSCE 4925: Project Aero
User's Manual

By: Alyssa Thurston, Breuna Riggins, James Sabetti, Travis Goral
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 16 February 2018
Status: Final Draft

Table of Contents

Revisions

Below is a list of any revisions made to this document.

Date	Description of Change Made	Person Making Change
2/12/2018	Document Created	Breuna, Alyssa
5/5/2018	Document Revision	Breuna, Travi

1 Introduction

The air quality in the City of Denton is notoriously bad. It's not a surprising fact given that there are two universities and multiple major highways and roadways that run through the city's limits. Worse still, there is only a single air quality sensor within city limits, and the next closest one is almost 15 miles away from the city.

The goal of this project is to create a network of air quality sensors to measure the air quality in the city of Denton so that citizens and city personnel alike can monitor air quality and take appropriate actions. The data obtained from the sensors is to be made public, so that air quality trends can be tracked all around the city. With these requirements met, the network would provide secure, real-time, and reliable data on the air quality in the city of Denton.

With the aim of creating a robust and open source of air quality data for the city of Denton, the Open Denton group has set out to establish a network of air quality sensors to gather and compile that data. In order to help achieve this goal, our group has been tasked with the following:

- Create a web client to display data in a highly readable manner.
- Create an API that records and displays air quality data in a readable manner
- Implement a scalable database to hold and model recorded data
- Integrate sensors and controllers that will read and relay data to the database over the internet
- Create documentation that allows community members to add sensors to the network

1.1 Purpose

The purpose of this document is to provide a thorough user guide for the user on how to use the site to either contribute their own personal node to the air quality system or how to navigate throughout the site to view different air qualities or node information data.

1.2 Definitions and Abbreviations

- D.A.M.N - Denton Air Monitoring Network
- Node- Air Quality Sensor
- AQI- Air Quality Index

1.3 Applicable Documents

- Maintenance Manual
- Test Plan

2 Getting Started

This section provides an outline of what the user will need to do to or have before jump starting on using the projectaerodenton.com site. Additionally, an overview of supported platforms is provided as well.

2.1 Supported Platforms

Browsers

- Internet Explorer
- Safari
- Mozilla Firefox
- Google Chrome
- Opera
- Microsoft Edge
- Default Browser of 3 Android Phones
- Safari on iPhone
- Google Chrome on iPhone

Softwares on the server

- XAMPP
- Python Library
- PHP
- PostgreSQL

Operating Systems

- Windows 7
- Windows 8 / 8.1
- Windows 10
- MacOS
- iOS
- Android

2.2 Getting Started

If a user simply just want to view the air quality in their local area without contributing a sensor, all they need to have and use is a browser to view the end client, the website. However; if they also want to contribute a sensor, the user simply needs to enter in the information about the sensor when adding the sensor to their account.

3 User's Manual

The purpose of the manual is for the users to use when they want to know essential information about the air quality system such as system functions, procedures, capabilities, how to use and navigate the website, and information about the nodes.

3.1 General Description

Provided below are the general descriptions of what an user will see from each menu item.

Navigating the Menu Items

Located on the top of the homepage are different menu items: 'Dashboard', 'Sensors', and 'About'. Click on which menu item you wish to navigate to.

Dashboard

The dashboard is the homepage of projectaerodenton.com. On the homepage, the user is able to view the daily AQI Index Averages, of Denton and Dallas, the current hazard level, the different air quality gauges, a map where all nearby sensors are located at, and a table displaying information about sensors.

Sensors

This is where you will see the AQI index averages data of your own personal sensor and other sensors nearby you, gauges of different air qualities, a map that will display where that sensor is located at, and a table showing a list of sensors that shows additional information about each sensor such as the sensor's name, status, and location.

About

The 'About' page is where the user will be able to go to learn more information about the AQI, the numerical system used to calculate the pollution in the air. Additionally, this is where the user can also view the team behind projectaerodenton.com, contact information, and additional resources.

Index

CSCE 4925: Project Aero
Maintenance Manual

By: Alyssa Thurston, Breuna Riggins, James Sabetti, Travis Goral
Project Manager: Alyssa Thurston
Instructor: Professor Keathly
Client: Dan Minshew and Kyle Tayle, Denton Techmill
Date: 16 February 2018
Status: Final Draft

Table of Contents

Revisions

Below is a list of any revisions made to this document.

Date	Description of Change Made	Person Making Change
2/12/2018	Document Created	Breuna, Alyssa, Travis, James
3/6/2018	Document Revision	Breuna, Alyssa, Travis, James
3/7/2018	Final Draft Edits	Breuna, Alyssa, Travis, James

1 Introduction

The air quality in the City of Denton is notoriously bad. It's not a surprising fact given that there are two universities and multiple major highways and roadways that run through the city's limits. Worse still, there is only a single air quality sensor within city limits, and the next closest one is almost 15 miles away from the city.

The goal of this project is to create a network of air quality sensors to measure the air quality in the city of Denton so that citizens and city personnel alike can monitor air quality and take appropriate actions. The data obtained from the sensors is to be made public, so that air quality trends can be tracked all around the city. With these requirements met, the network would provide secure, real-time, and reliable data on the air quality in the city of Denton.

With the aim of creating a robust and open source of air quality data for the city of Denton, the Open Denton group has set out to establish a network of air quality sensors to gather and compile that data. In order to help achieve this goal, our group has been tasked with the following:

- Create a web client to display data in a highly readable manner.
- Create an API that records and displays air quality data in a readable manner
- Implement a scalable database to hold and model recorded data
- Integrate sensors and controllers that will read and relay data to the database over the internet
- Create documentation that allows community members to add sensors to the network

This document will outline the maintenance manual with procedures and guidelines on how to maintain the system, software, and database which will be used by the client and the users in the future.

1.1 Purpose

The purpose of this document is to help the client and users have a better understanding of the technical requirements on maintaining the overall product of the air quality system which will help prolong the lifetime of the different operating systems, equipment, and structure.

1.3 Scope

This manual provides detailed guidelines and information on the maintenance of the overall system to ensure the longevity of air quality sensors , the different operating systems, structure, and equipment. This manual contains the following guidelines to guarantee:

- That the air quality system performance satisfies every requirements of the system

- That the air quality system is maintained properly
- That the performance of the air quality system is optimised

1.4 List of Definitions and Abbreviations

The following is a list of abbreviations and definitions to words that are used in this document;

- D.A.M.N - Denton Air Monitoring Network
- Node- Air Quality Sensor
- AQI- Air Quality Index
- Project Aero: This Project, which encapsulates the D.A.M.N.
- UFW: Uncomplicated Firewall; Explained in section 2.2

1.5 Overview

This manual serves as the general documentation on everything that needs to be done to keep the server functional. It also serves as the documentation on getting software put on the server, in addition to configuring that software.

The various sections of this document can be found in the table of contents.

2 System Information

This section provides an overview of what is involved in the system, and also includes a section regarding security concerns. Additionally, an overview of the communications network is provided as well.

2.1 System Architecture

This system is comprised of a front end, database, web server, and API. The API is responsible for communication information from the Web Server and Database to the front end and vice versa. Additionally, the API will take in information from the Nodes and send it to the server as necessary.

The front-end is on the client's side, and really won't have much to do with us. This front end is simply the internet browser of the user's choice. We are going to support the following browsers:

- Internet Explorer
- Safari
- Mozilla Firefox
- Google Chrome
- Opera
- Microsoft Edge
- Default Browser of 3 Android Phones
- Safari on iPhone
- Google Chrome on iPhone

The database, web server, and API all reside on the server. The server that we are running is Windows Server 2016, with all available updates installed. The server has the following installed on it:

- XAMPP
- Python Library
- MySQL
- PostgreSQL
- JQuery
- Anaconda
- Firefox
- Teamviewer 13

The database is running a PostgreSQL database that presents the data in a time-series format which is ideal for a project such as ours that is designed to take any input and show it in a user friendly matter.

Our web server component is utilizing Apache. Apache is a common, powerful web server. This is what is used to push the HTML based front end out to the internet. This front end is what displays all of the applicable data from the website.

The last component in our system is the API. The API is responsible for communicating with the database and web server, and getting the data to and from the client. Additionally, our API will also be responsible for communicating with the nodes that are put onto our network.

2.2 Security

The main line of security in our system is the firewall. For our firewall, we are utilizing Windows Firewall which comes with the standard Windows Server 2016 installation Some of the technologies that are being utilized require ports to be open and exposed. These include:

- 22
- 25
- 80
- 443
- 5091

3 Environment

This section includes information on the environment that is used on this system. It will include the storage, software, and hardware requirements, in addition to versions of software and other useful information.

3.1 Equipment

There are few requirements for equipment. In order for the system to work properly, all that is needed is a server, and the client side requires a device with an internet browser.

In order to get data, a node has to be connected to the system. This node can be purchased online, or can be one made from a Raspberry Pi.

3.2 Storage Requirements

Client-Side, there are no storage requirements as the system is cloud based, and thus, is not stored locally. Server-side, 50GB is recommended for full system functionality and performance, and additional space is necessary to store data sent from the users.

3.3 Software

For the client side of things, we support the following operating systems or devices:

- Windows 7
- Windows 8 / 8.1
- Windows 10
- MacOS
- iOS
- Android

4 System Maintenance

Detailed below are the maintenance procedures for the system. Included are procedures for updates, back-ups, any encountered errors, and responsibilities of different people.

4.1 Responsibilities

Responsibility for system maintenance may be covered by the development team for a period no longer than 90 days after end of 2018 Spring semester. After this point, responsibility for system maintenance is left to owner's discretion. No more than one person may be needed for routine maintenance issues.

4.2 Error Conditions

The most likely error codes one may encounter would be common HTTP Status Messages if the database container or apache server was down. Listed below are common HTTP error codes and their respective explanations:

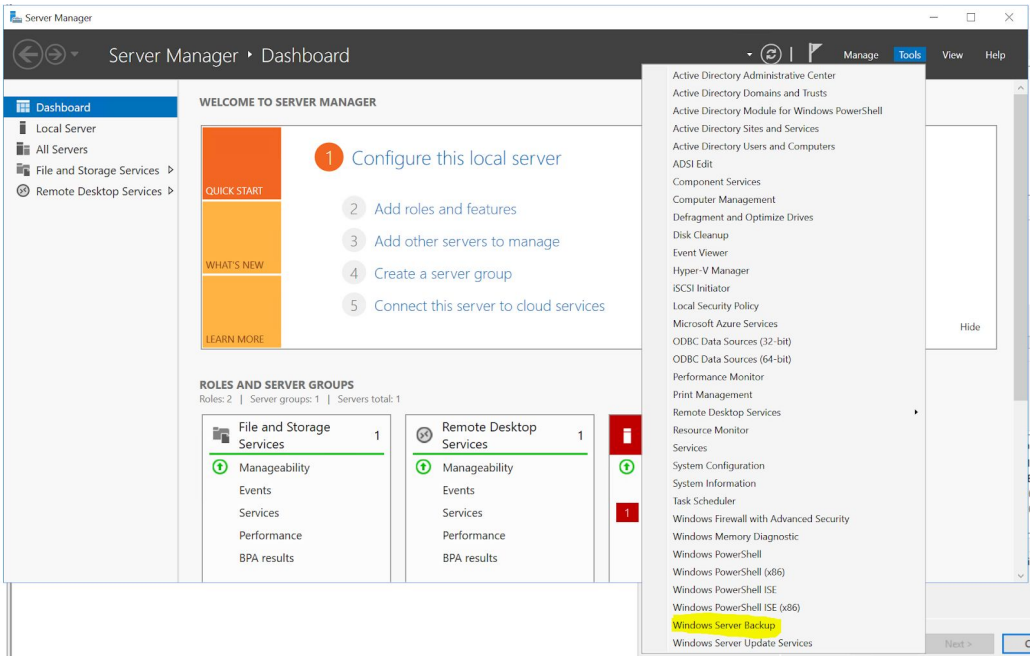
404: The page cannot be found
500: Internal Server Error: Generic error
502: Bad Gateway: Invalid response from upstream server
503: Service Unavailable: Server is overloaded or down
504: Gateway Timeout: Did not receive timely response from upstream server.

4.3 Maintenance Procedures

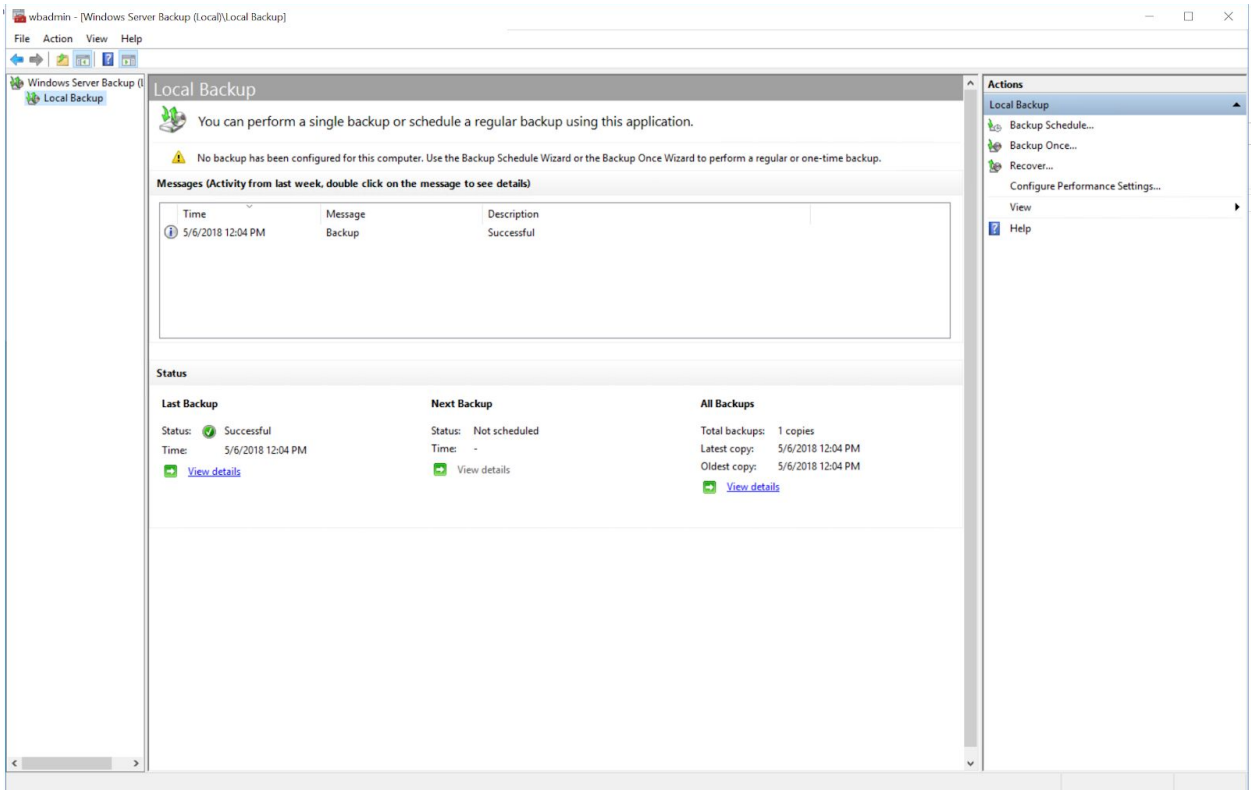
Listed below are procedures for backups, updates, and other miscellaneous maintenance items.

4.3.1 Backing Up

Backups are scheduled and configured via the Windows Server Manager too. As a prerequisite, Windows Server Backup needs to be installed in Server Manager > Add roles and features > Next > Role-based or feature-based installation > Next > Next > Next > Select "Windows Server Backup" and Install.



To create a new backup, go to the Server manager and select Tools > Windows Server Backup From here, you can schedule backups, create a system image, and restore from a system image. To create a system image click Backup Once under the Actions menu and follow the setup. A Full Metal backup will create a system image with all of the current programs and settings preserved. To restore from this image, choose the Recover option under Actions and select your system image file.



4.3.4 Updates

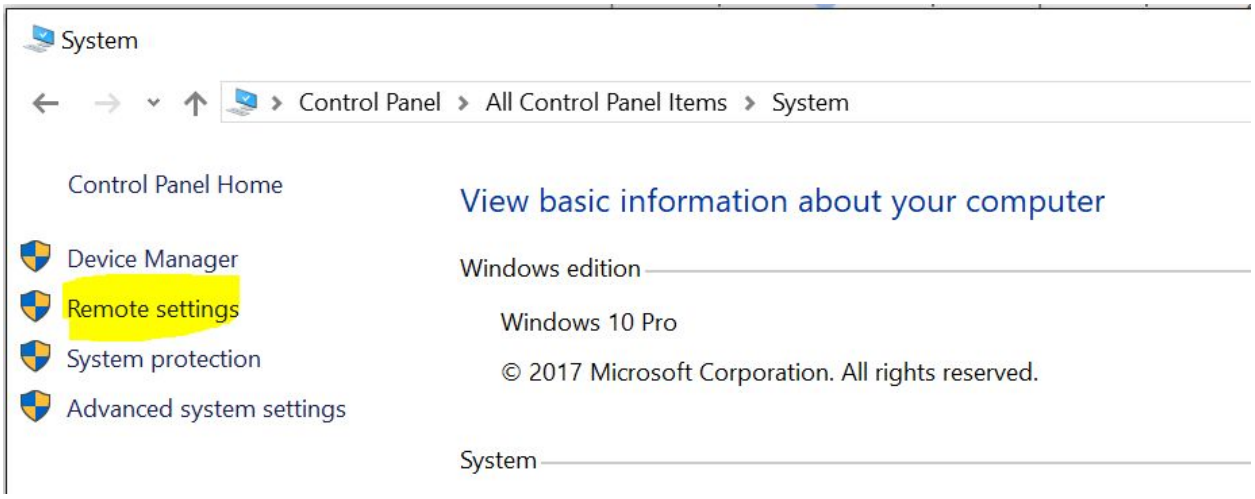
Occasional updates may be desirable to keep packages working and optimal. To do this, log into the server and Start > Settings > Update & Security. Here you can schedule updates or install them manually. It is recommended to install updates at a time when the server is experiencing the least demand.

4.3.5 Accessing the Server Remotely

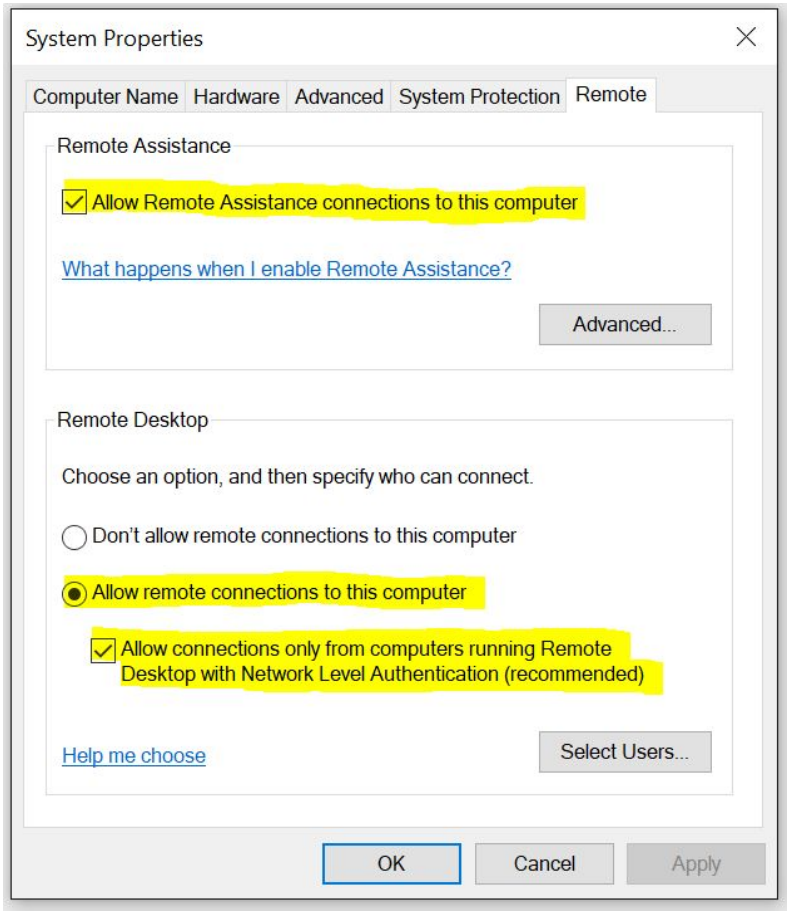
Windows Remote Desktop Connection is required to connect to the server. To run RDP from a Windows computer, press the Win + R key and type “mstsc.exe” and click Run.

To find the right IP address, go to the browser on the server and type “What is my IP Address,” it should spit it out to you. Type this address in and click connect. You will be asked to provide credentials upon login.

If Remote Desktop is not already configured on both the Server and Client computers, it will need to be enabled in order to connect to the server from a remote client. To enable Remote Desktop on Windows, go to the Control Panel, select Large icons under the View by: menu, then select System. From here, select Remote settings from the top-left.



Make sure all the following options are selected on both the client and server computers so that you can use Remote Desktop connections:



Depending on your router settings, you may also need to allow the default RDP port 3389 in order to allow Remote Desktop connections. If problems still persist, check your firewall settings for any deny rules that would forbid connections on port 3389.

5.0 Software Maintenance Procedures

Listed below are detailed descriptions for any maintenance procedures the software units may need to go under

5.1 Consolidated Unit List

5.2 PostgresQL Database
5.3 XAMPP

5.2 PostgreSQL Database

This is the database we are using to house all of the recorded information sent to us by the sensors. This will be available on a remote server via a virtual machine.

5.2.1 Description

The PostgreSQL Database will serve as one of the major focal points for our system. It will house all recorded information, sent to us by the sensors in the field, and will relay that information, via an API, to our front end website. Our database runs with the standard SQL language in place and is designed for ease of use.

In terms of its relation to other softwares, PostgreSQL will run in conjunction with the API in order to deliver the necessary data to the front end, which will be hosted by the Apache2 Server and displayed on the front end.

5.2.2 Functions

The SQL database will be housing the records for all of the sensors and then sending this information, as needed, to the front-end website.

5.2.3. Input

Data input into the SQL database will come as a series of numbers, all with predetermined attributes attached to them, from the sensors in the field. Depending on the sensor, there will be a different magnitude of numbers coming into the database, and the order in which they arrive, as well as their attributes, will determine where they go within the database.

In review, as previously stated, the SQL database will take in a series of numbers sent by the sensors within the field. These numbers will be stored in the database and organized there for use by the LocalHost via the API.

5.2.4 Processing

5.2.4.1 Initiation Procedures

The PostgreSQL database should be up and running upon startup and requires no initiation on the user's part. Configuration and administration is performed via the pgAdmin application located on the desktop

5.2.4.2 Core Processing Procedures

The SQL database is designed to intake numerical and character values and assign them to certain categorical attributes based on the design of the sensor inputting the information to the system.

5.2.4.3 Branching Conditions

The SQL database has no branching conditions.

5.2.4.4 Restrictions

As of writing this, the SQL database has the ability to intake any kind of value, whether it be numerical, character, timestamp, string, etc. Values and be dependent on the sensor reporting the information in order to attribute the categories for the values. The SQL database also does not have a filter gap in place to prevent incorrect on inaccurate numbers from being reported.

5.2.4.5 Exit Requirements

If the SQL database ever needs to be stopped, for whatever reason, open the task manager and locate the PostgreSQL process and end it.

Do not forget to start the database up again whenever the issue has been resolved.

5.2.4.6 Communications

The SQL database will communicate with the front-end via queries run by php files that should output the information to a chart. As of today, there is no chart functionality however data can be displayed correctly on front-end.

5.2.4.7 Output

The SQL database will simply output the information requested from it via a query from the php code.

5.2.4.8 Unique Features

As of the writing of this manual, the SQL has no unique features.

5.2.5 Data Structures

As per the EPA guidelines, our sensors will report any of the following information back to the database and categorize them based on the attributes applicable:

- Carbon Monoxide
- Lead
- Nitrogen Dioxide
- Ozone
- Particle Pollution
- Sulfur Dioxide
- Location (address or longitude or latitude coordinates)
- Time
- Identifying number/name for the sensor

5.2.6 Verification Procedures

A user will need to have an account with our front end in order to have their information reported to the database in order to ensure responsibility and accuracy for all information reported.

5.2.7 Listings

SQL utilizes the standard software listings.

5.2.8 Interfaces

The SQL database will not necessarily graphically interface with any other software, but it will utilize an API connection in order to communicate back and forth with the front end and display the relevant information.

5.3 XAMPP

This is what we will be using to design out front end and make sure that everything is working properly.

5.3.1 Description

XAMPP is the platform that the front end of the website will be broadcasted from.

5.3.2 Functions

XAMPP is the Apache distribution that our website will be hosted from. It is run locally on the server itself and is currently using the projectaerodenton.com domain

5.3.3. Input

All inputs will come from the html, php, or javascript files stored within the Public Documents folder. These files can be altered and changes will be reflected on the projectaerodenton website.

5.3.4 Processing

5.3.4.1 Initiation Procedures

Once an html or javascript file is ready to be a webpage, simply upload or save it to the Public Documents folder and if necessary, restart the Apache service from the XAMPP control panel.

5.3.4.2 Core Processing Procedures

All information input into the website will be via an html page of a javascript page.

5.3.4.3 Branching Conditions

The webpage has no branching conditions.

5.3.4.4 Restrictions

As of writing this, the webpage is restricted based on its handling capabilities of both html and javascript.

5.3.4.5 Exit Requirements

If the webpage ever needs to be stopped, for whatever reason, simply stop the Apache process from the XAMPP control panel

Do not forget to start Apache again whenever the issue has been resolved.

5.3.4.6 Communications

The webpage will communicate with the PostgreSQL database via queries run by javascript and php files.

5.3.4.7 Output

The webpage will display the information within the html files in the Public Documents folder.

5.3.4.8 Unique Features

As of the writing of this manual, the webpage no unique features.

5.3.5 Data Structures

All data stored within the webpage can be found in the Public Documents files and folders of html, conf, or javascript file extensions.

5.3.6 Verification Procedures

The only way to edit any of the information within the webpage is to go into the server's Public Documents and edit the files from there.

5.3.7 Listings

The webpage will follow the basic software listings.

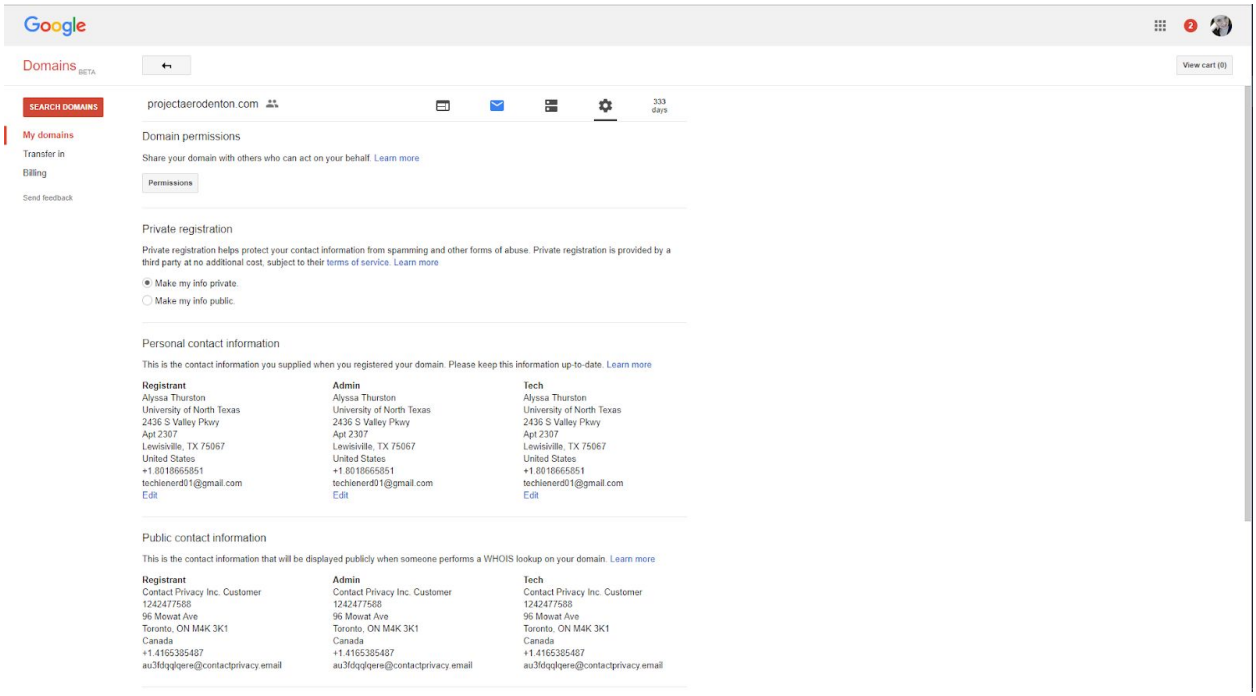
5.3.8 Interfaces

The webpage files can be edited via a text editor such as Notepad++ within the server desktop itself.

5.4 Google DNS Information

Before delving into the depths that you need to know about Google DNS, it's important to first understand what DNS is. DNS, Domain Name Service, is a service that says "this IP address (192.168.1.1 for example) belongs to this name (www.google.com, or google.com, for example)." We have purchased the name "projectaerodenton.com" for a year. In order for this to work, it is important that DNS service is set up properly.

First, go to domains.google.com/registrar, and click "settings" next to projectaerodenton.com. This should bring you to a screen similar to the following:



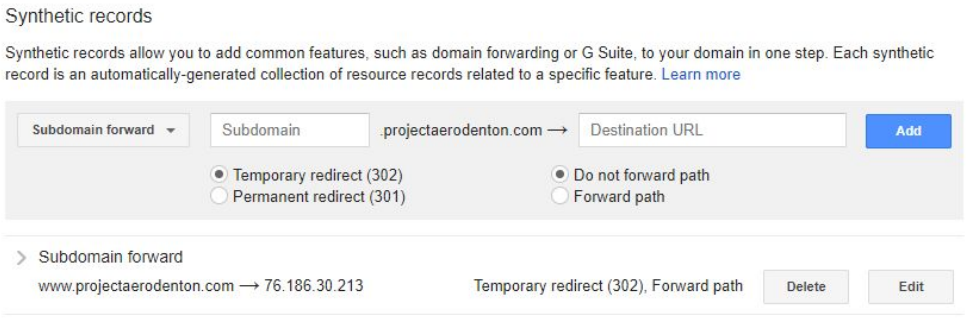
As a basic walk-through of the most important things here:

- If you click the Permissions button, you can share the domain administration with other people. At the time of this writing, we have shared them with Dan Minshew and Kyle Taylor. No other people will have access to these domain settings.
- We recommend keeping the “Make my info private” setting turned on. This way, you don’t have the people of the internet showing up to your doorstep.
- Change the personal contact information to match your own information. Alyssa has included hers there as a placeholder.
- The public information is only ever displayed if someone performs a WHOIS lookup on your domain. To get your information, they’d have to go through the company listed there.

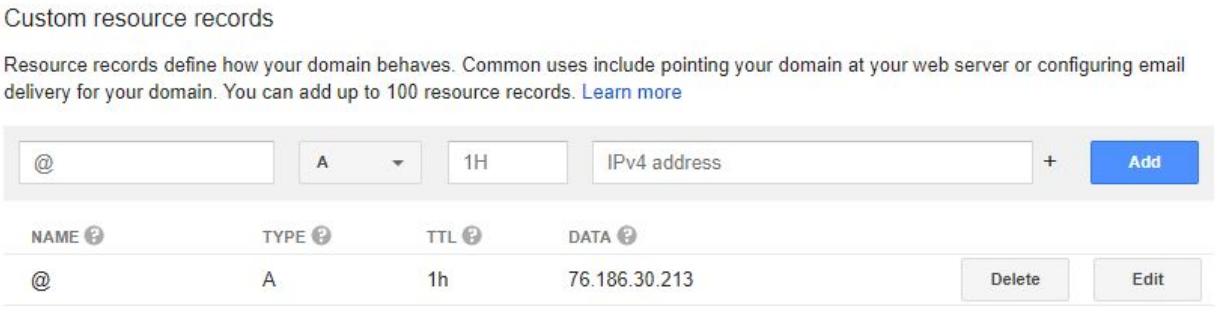
The next essential screen is the “Configure DNS” page, from the settings screen, you can access that by clicking the icon to the left of the settings gear, it looks like two rectangles stacked on top of each other. There is only two settings that you’ll want to edit or change here:

- Synthetic records: These allow for you to put the “www” in front of “projectaerodenton.com” and will still direct you to the right page. We have configured “www.projectaerodenton.com” to forward to our server. When your own server is up and

running, you'll need to change the IP Address shown here to match your own.



- Custom resouce records: Simply put, these point your domain to the address that they need to map to. Here, we have set the domain to point to the IP Address of our server. Ensure that you change this when you get your own server setup.



One last cool feature that you can set up is your own custom e-mail forwarding, which is done by clicking the envelope to the left of the "configure DNS icon." If you wanted to create an email such as "admin@projectaerodenton.com" you would type "admin" in the first box, then type the email address of where you would want emails sent to "admin@projectaerodenton.com" to go. This feature doesn't create an email account, per se, but will forward all emails sent to a specified address to the email listed here.

Billing information is provided on the last icon, which shows the number of days before registration of your domain expires. You can turn on auto-renewal here, or add years. Transfer out is the process you go through to transfer ownership of your domain to someone else.

Project Aero

CSCE 4905/4925 Senior IT Capstone
Fall 2017 - Spring 2018
Faculty Sponsor: Professor David Keathly
Group Name: Fantastic Four
Group Members: Alyssa Thurston, Breuna Riggins, Travis Goal, James Sabetti
Sponsor Organization: Denton Techmill
Sponsor Names: Dan Minshew & Kyle Taylor

Overview

Project Aero is meant to tackle the question of air quality in the city of Denton. There has been a lot of research done on the topic of air quality, specifically in the city of Denton. The city itself is shaped almost like a bowl, which traps pollutants in the city. Moreover, Texas has predominant southeast to northwest winds that carry pollution from the coast, through the coal and gas patches of East and Central Texas, over the Midlothian cement plants, and through many urban areas, straight to the heart of Denton. Project Aero will enable citizens of the city to create an air sensor, contribute it to the network, and help answer the question of how bad the air quality is in the city of Denton.

Currently, there is only one air sensor which is near the Denton Municipal Airport, which collects data hourly. It too, has a front end where the public may view the data, but it is not displayed in an easy to read format. We worked to develop a front end that would enable users to quickly glance at the data and get a complete picture of what the air quality is like in the city. Users can also get historical data from past dates, view a map of all sensors on the network, and view data collected from a specific node. All the data on a PostgreSQL database on a central server, and the front end pulls the data from that database using an API in real time.

Requirements

The main requirements are outlined below:

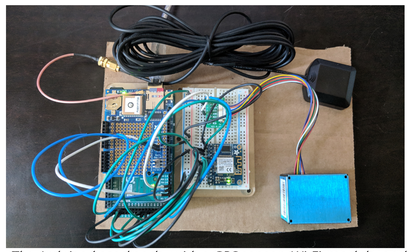
- Database capable of scaling with zero down-time
- Modeling of data in a time series format
- An open API that collects and retrieves network traffic
- Wi-Fi or 3G capable Microcontrollers that can act as network nodes
- Drivers for particular sensors to interface with Microcontrollers
- Web-based “homepage” where anyone can view and interpret datasets
- Human-readable graphs to help provide data analytics
- Open-source practices, with all code available for review and contribution by the team, or anyone else in the Open-Source community

Features

The most unique part of Project Aero is that the data it gets comes from the public. In order to help encourage user input, users have access to a Do-It-Yourself guide to make their own sensors. This gives Project Aero the potential to combine the community aggregate of information to create a full picture for the air quality within Denton and report it to the populace, all created by the people it serves.

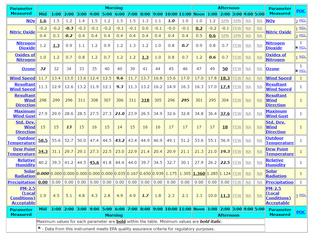
Abstract

Due to the copious amounts of traffic from various highways, a multitude of large businesses, and two universities, it is quite shocking to find that there is only one air quality sensor in Denton (located at the Denton Municipal Airport). The next closest is over 15 miles away. Given these facts, we don’t have an accurate and widespread view of how bad the air quality is in the city. It is the goal of this project to rectify the lack of air quality data. This project will help the citizens of Denton become publicly aware of the air quality, the standards set by the EPA, and take civic action. This project’s innovation is derived from the fact that the data collected is publicly available and displayed in a manner that the average citizen can easily digest. Additionally, citizens can elect to create their own sensor and contribute the data collected from it into the publicly accessible database.

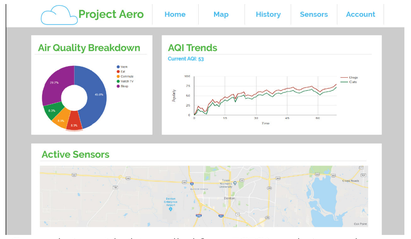


The Arduino based node, with a GPS sensor, Wi-Fi module, and particulate matter sensor.

The team. Alyssa (front), James (middle, right), Travis (middle, left), and Breuna (back).



TCEQ’s website with the data from the Denton Sensor.



Our website, with data pulled from TCEQ’s website, and our own personal node.

EPA Standards, TCEQ, & Denton

The EPA (Environmental Protection Agency) has implemented national laws in specific regards to the air quality in the United States. TCEQ (Texas Commission on Environmental Quality) has been put in charge of ensuring that the standards set by the EPA are met in the state of Texas.

TCEQ is the organization that is also in charge for providing air quality forecasts, managing about 150 air quality sensors statewide, and for punishing other organizations who do not meet state and federal regulations.

So how is Denton doing in regards to these regulations? In short, not well. There are two types of ozone, some in our atmosphere that protects us from the sun, and some that forms at ground level from vehicles and refineries which react in sunlight. Below is a table of the number of high days, where ozone levels exceeded 71 parts per billion.

Year	Days above 71 PPB
2013	40
2014	15
2015	30
2016	13
2017	16

Source: www.tceq.texas.gov

Challenges

The team faced many varied challenges. Chief among them was the construction of a working sensor controller from the ground up. This involved physically soldering wires to the circuit boards, learning a new proprietary programming language for the controller, and configuring each sensor component so they would all work together.

Another challenge was trying to find the right kind of server setup that could host all of our necessary applications and produce desired functionality. We started out using an Ubuntu Server distribution hosted by UNT on their network. In order to access the server, you had to be on a VPN, which introduced a host of new issues, including getting sensors to connect and deliver data to them. In addition, many compatibility issues as well as problems with web hosting while on the UNT virtual network were experienced.

The biggest lesson learned was that when working with a project of this size with so many new aspects and unexplored territory, it is best to jump into development as early as possible. Doing so would have provided the greatest chance at discovering and solving unforeseen issues that impacted the team’s overall project requirements.

Project Aero

Denton Air Monitoring Network
(D.A.M.N.)

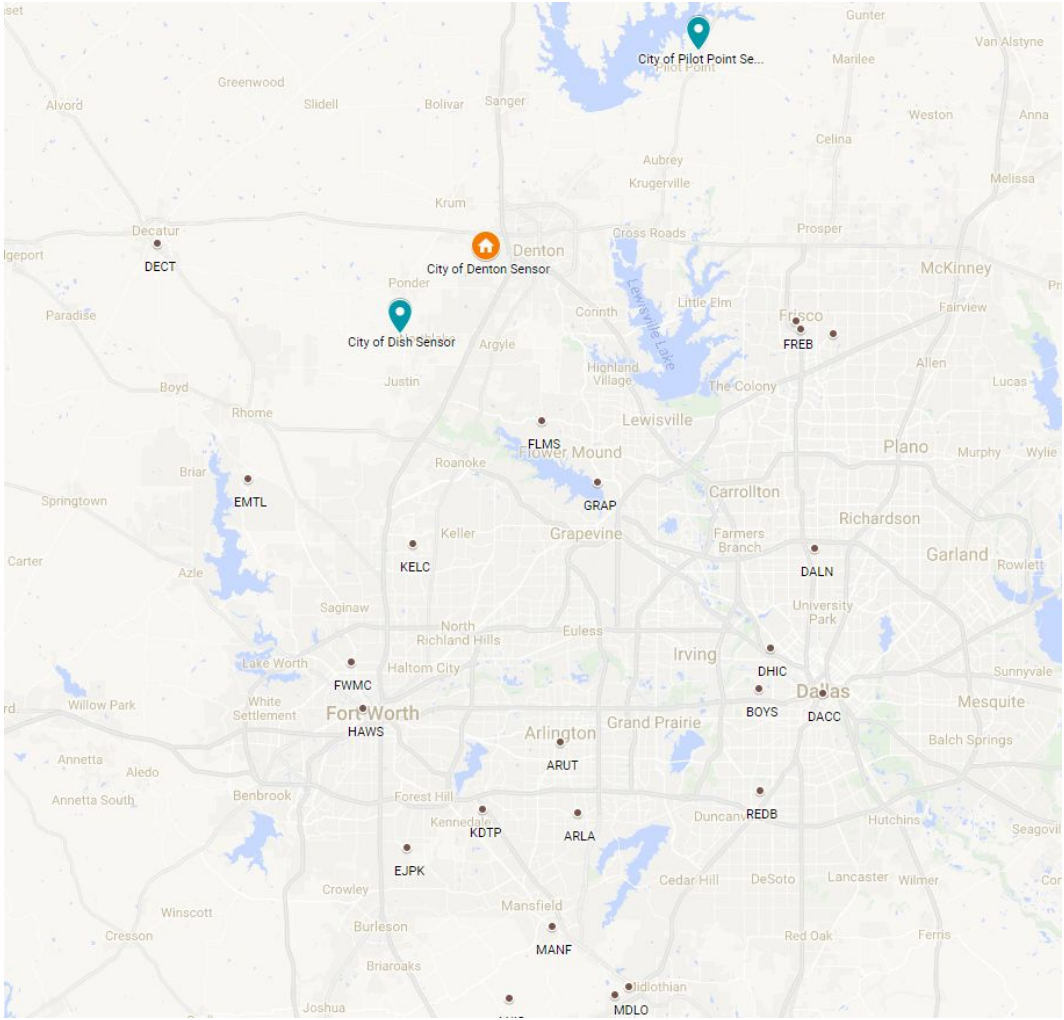
The Fantastic Four & Our Client

Denton Techmill: Dan Minshew & Kyle Taylor

“...the worst ozone levels were found in the northwest quadrant of the DFW area... caused by the predominant SE to NW winds that blow pollution from the coast up through the coal and gas patches of East and Central Texas, over the Midlothian Industrial Complex and North Texas central urban cores into Northwest Tarrant, Wise and Denton counties.”

The problem

- One sensor in Denton (at the Airport)
- No sensors directly North of Denton
- Bottom-line



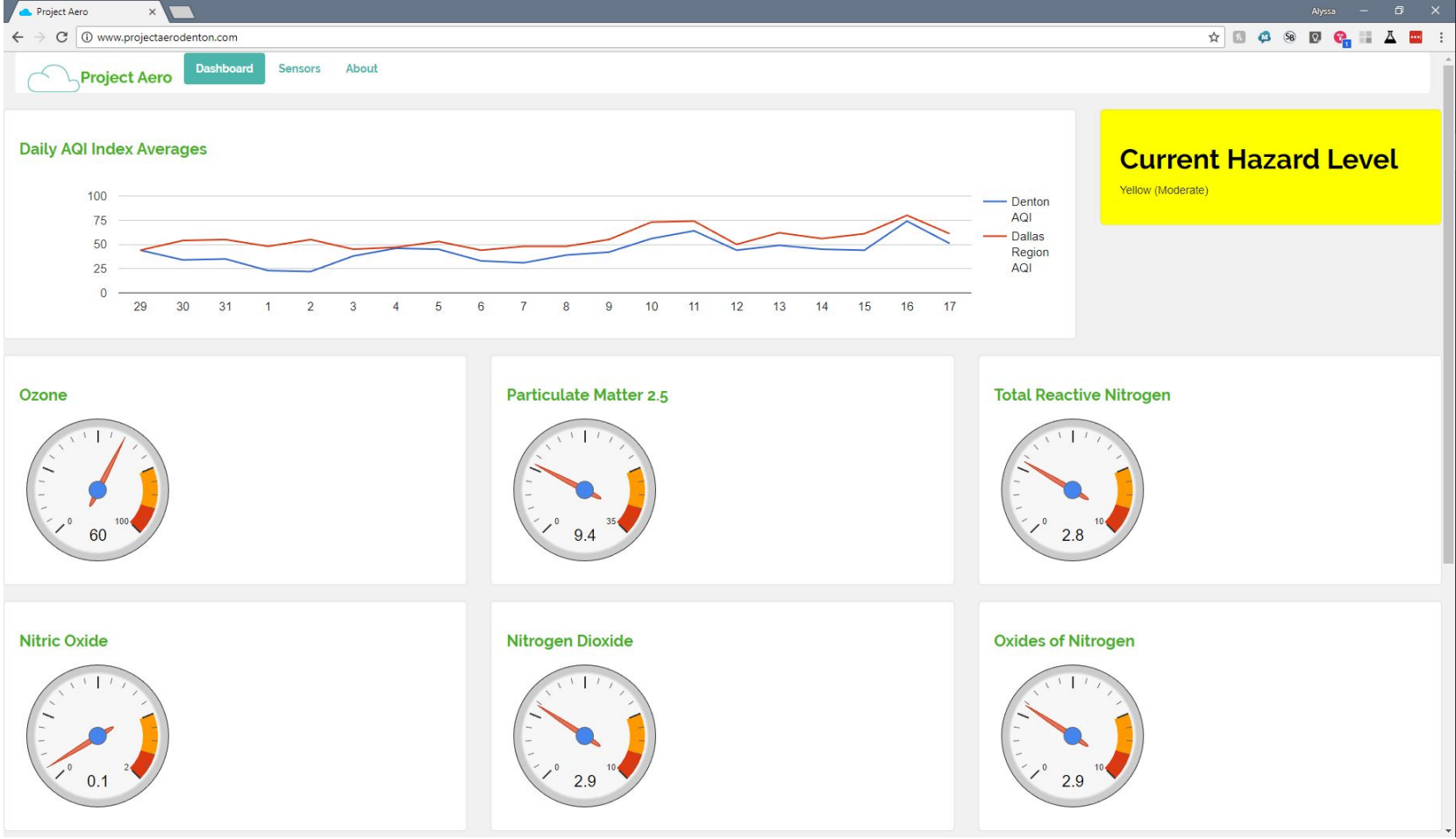
Introducing the Denton Air Monitoring Network (D.A.M.N.)

What is Project Aero?

- Partly a website, partly a database
- Allows for a network of air quality sensors (also known as nodes)
- Measures different pollutants and qualities of the air
- Breaks the data down into an easy to read format
- Enables citizens to create their own sensor and contribute their data

What is the solution?

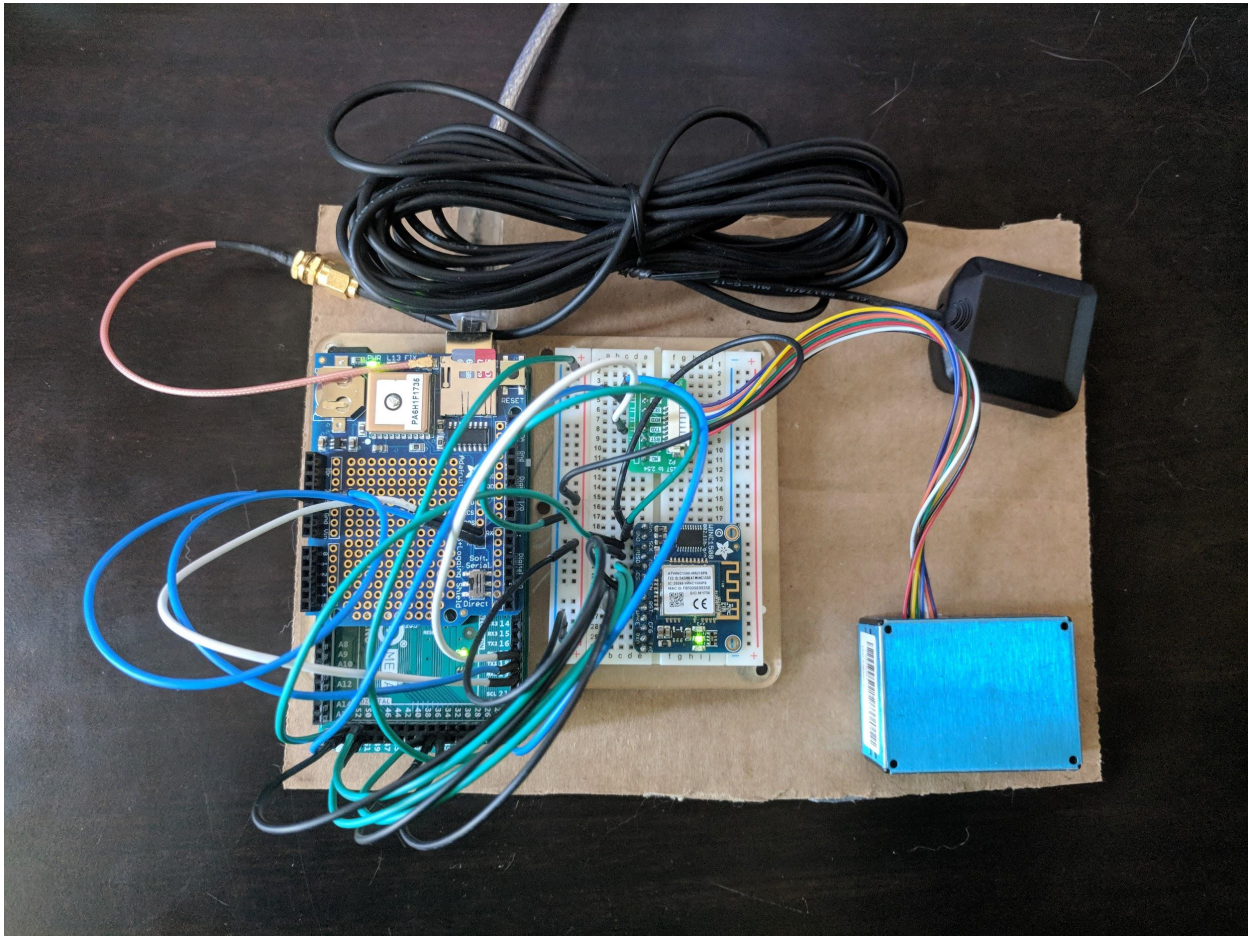
You!


Project Aero
www.projectaerodenton.com
Project Aero
Dashboard
Sensors
About
Daily AQI Index Averages
100
75
50
25
0
29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Denton AQI
Dallas Region AQI
Current Hazard Level
Yellow (Moderate)
Ozone
60
Particulate Matter 2.5
9.4
Total Reactive Nitrogen
2.8
Nitric Oxide
0.1
Nitrogen Dioxide
2.9
Oxides of Nitrogen
2.9

Homepage

What does all that mean?

- EPA and TCEQ Governed Regulations
- Six Critical Pollutants
  - Ozone
  - Particulate Matter 2.5
  - Total Reactive Nitrogen
  - Nitric Oxide
  - Nitrogen Dioxide
  - Oxides of Nitrogen
- AQI: Main Average



The node

Empowering Citizens

- Contribute any data to the network
- See the air quality around Denton
- Ultimately, take civic action

“Denton’s ‘average ozone level stands at about 80 ppb,’ the highest in Texas in 2016.”

Ups and Downs

The Ups

- Worked effectively as a team
- Availability of existing data
- Relatively straight forward

The Downs

- Google Charts
- Front-end to back-end communication
- Bootstrap Issues
- Conflicting Requirements

Why Project Aero?

Breuna

Want to save the world, one air quality sensor at a time.

Alyssa

I chase storms, am deeply intrigued by the weather, and was drawn to the environmental aspects of this project.

James

This system not only informs, but empowers the average citizen to participate.

Travis

We only get one planet and we need to take better care of it.

Questions?

Bi-Weekly Status Report

CSCE 4905/4925

Date: 28 September 2017
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2-week period

- Meeting the client
- Completed Project Plan draft
- Established a communication channel with the team
- Established group member work role

Problems Encountered

- Defining the scope of the project
- Identifying what was expected of us

Both problems were solved on 9/26 during a meeting with a client

Plans for the upcoming 2-week period

- Requirements Specifications
    - Lecture, Plan, Draft compilation
- 1 Team meeting planned for 10/10/2017, number of meetings may change
- Research databases to use with the project

Project Plan Reconciliation
Because the document was recently compiled, no changes to the original document have been made.

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
Class 1- 9/19	Present	Present	Present	Present
Class 2- 9/21	Present	Present	Present	Present
Class 3- 9/26	Present	Present	Present	Present
Class 4- 9/28	Present	Present	Present	Present
Team Meeting 1- 9/21, during class	Present	Present	Present	Present
Team Meeting 2- 9/26, during class	Present	Present	Present	Present
Team Meeting 3- 9/26, with the client	Present	Present	Present	Present

Status Report #2

CSCE 4905

Date: 12 October 2017
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2-week period

- Work is almost complete on the Requirements Document

Problems Encountered

- No problems so far

Plans for the upcoming 2-week period

- Meet with the client to discuss the requirements document on 10/17, and make any edits necessary before turning it in.

Project Plan Reconciliation

- We are still on track with our project plan, and everyone is planning on having things ready tonight or tomorrow so the draft can be turned in on time.

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
Requirements Lecture- 10/3	Present	Present	Present	Present
Meeting- 10/10	Present	Present	Present	Present

Status Report #3

CSCE 4905

Date: 26 October 2017
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2-week period

- Requirements review presentation
- Work is in progress on the Preliminary Design

Problems Encountered

- Not sure what's happening with the sensors. We have a meeting with the client on 10/31 and will attempt to get that straightened out then before turning to Professor Keathly for help. The client is attempting to work with an outside source in order to design and make the sensors, however, the two seem to have differing ideas about how the sensors should work, which could lead to be very problematic down the road

Plans for the upcoming 2-week period

- Meet with the client to discuss and go over the Preliminary Design
- Set milestones for the project
- Complete the Preliminary Design

Project Plan Reconciliation

- We are still on track with our project plan, no reconciliation needed

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
Prelim Design Lecture- 10/17	Present	Present	Present	Present
Prelim Design Lecture- 10/19	Present	Present	Present	Present
Team Meeting- Requirements Review Prep	Present	Present	Present	Present
Requirements Review Prep	Present	Present	Present	Present

Status Report #5

CSCE 4905

Date: 30 November 2017
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2-week period

- Work was completed on the Detailed Design
- We were ale to determine languages and tools that we are going to use for the project.

Problems Encountered

- None this two weeks.

Plans for the upcoming 2-week period

- Work on learning the languages and tools well, and begin thinking about what the UI is going to look like.
- Work on the Design Presentation

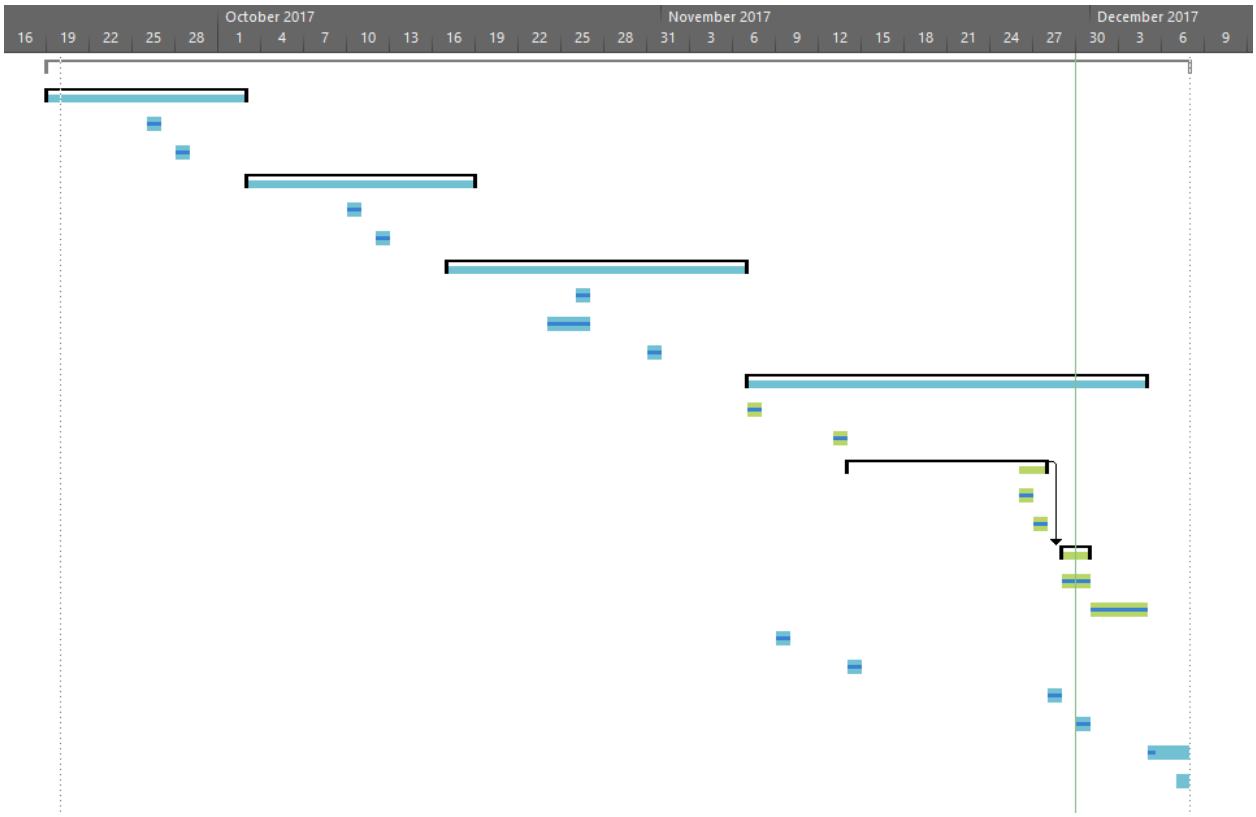
Project Plan Reconciliation

- We are still on track with our project plan, no reconciliation needed. Gantt Chart included on the next page.

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
Meeting with Keathly- 11/14	Present	Present	Present	Present
Lecture- 11/14	Present	Present	Present	Present
Hangouts call with the Client- 11/28	Present	Present	Present	Present

		Schedule	**59 days**	**Tue 9/19/17**	**Thu 12/7/17**		
✓		**Project Plan**	**11 days**	**Tue 9/19/17**	**Mon 10/2/17**		
✓		Team Meeting (During Class)	1 day	Tue 9/26/17	Tue 9/26/17		
✓		Status Report 1	1 day	Thu 9/28/17	Thu 9/28/17		
✓		**Requirements Specifications**	**12 days**	**Tue 10/3/17**	**Wed 10/18/17**		
✓		Team Meeting (During Class)	1 day	Tue 10/10/17	Tue 10/10/17		
✓		Status Report 2	1 day	Thu 10/12/17	Thu 10/12/17		
✓		**Preliminary Design**	**15 days**	**Tue 10/17/17**	**Mon 11/6/17**		
✓		Status Report 3	1 day	Thu 10/26/17	Thu 10/26/17		
✓		Requirements Review	3 days	Tue 10/24/17	Thu 10/26/17		
✓		Team Meeting (During Class)	1 day	Tue 10/31/17	Tue 10/31/17		
✓		**Detailed Design**	**20 days**	**Tue 11/7/17**	**Mon 12/4/17**		
✓		Detailed Design Lecture	1 day	Tue 11/7/17	Tue 11/7/17		
✓		Detatiled Design Plan	1 day	Mon 11/13/17	Mon 11/13/17		
✓		**Detailed Design Draft**	**10 days**	**Tue 11/14/17**	**Mon 11/27/17**		
✓		Document Compilation	1 day	Sun 11/26/17	Sun 11/26/17		
✓		Document Editing	1 day	Mon 11/27/17	Mon 11/27/17		
✓		**Detailed Design- Final**	**2 days**	**Wed 11/29/17**	**Thu 11/30/17**	**38**	
✓		Necessary Revisions	2 days	Wed 11/29/17	Thu 11/30/17		
✓		Detailed Design Report	2 days	Fri 12/1/17	Mon 12/4/17		
✓		Status Report 4	1 day	Thu 11/9/17	Thu 11/9/17		
✓		Team Meeting (During Class)	1 day	Tue 11/14/17	Tue 11/14/17		
✓		Wrapup Lecture	1 day	Tue 11/28/17	Tue 11/28/17		
✓		Status Report 5	1 day	Thu 11/30/17	Thu 11/30/17		
		Design Review	3 days	Tue 12/5/17	Thu 12/7/17		
		Status Report 6	1 day	Thu 12/7/17	Thu 12/7/17		


October 2017
November 2017
December 2017
16 19 22 25 28 1 4 7 10 13 16 19 22 25 28 31 3 6 9 12 15 18 21 24 27 30 3 6 9

Bi-Weekly Status Report

CSCE 4905

Date: 7 December 2017
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2-week period

- Assembled a to-do list for the winter break
- Got a server from UNT

Problems Encountered

- None

Plans for ~~the upcoming 2-week period~~ winter break

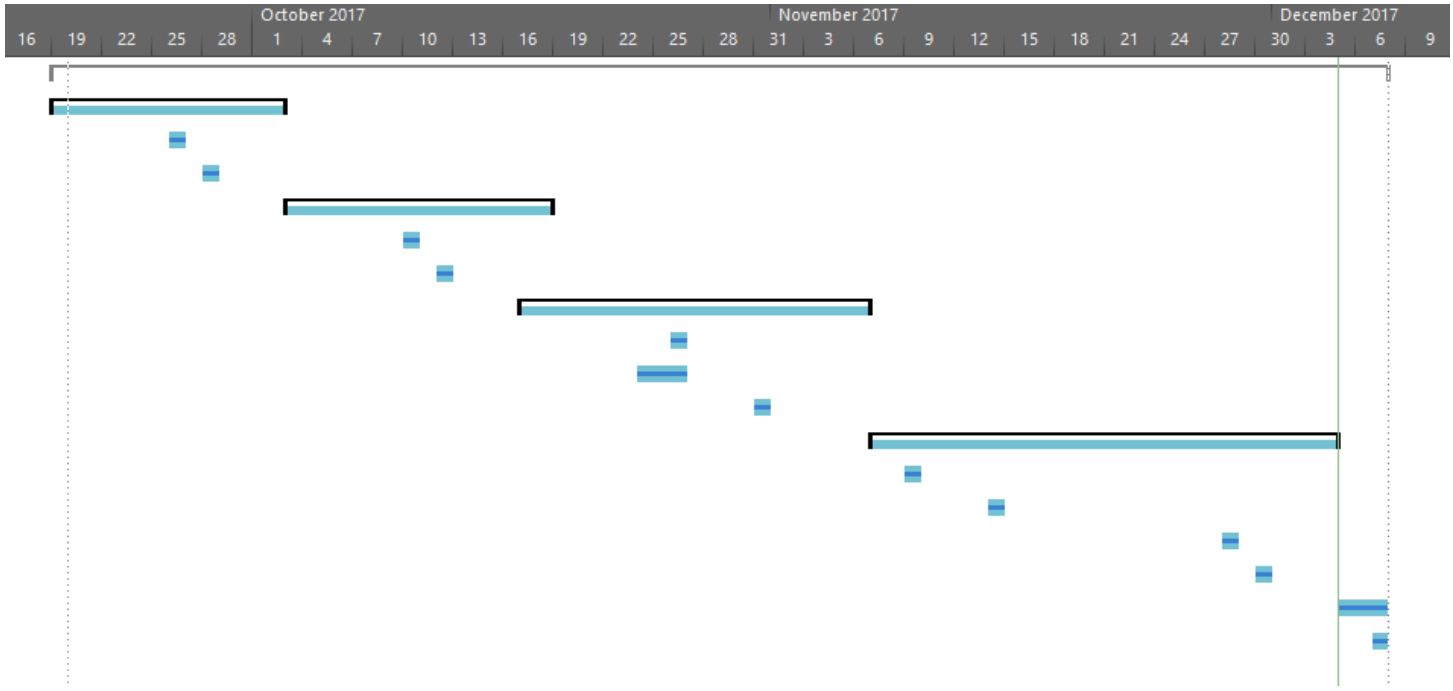
- Learn Ruby and PostgreSQL
- Setup the server and get it ready to go
- Setup database
- Create a "Wireframe" of sorts/figuring out what the front end will look like

Project Plan Reconciliation
On track, no reconciliation needed. Gantt Chart attached.

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
Design Review Day- Thursday	Present	Present	Present	Present

✓		**Schedule**	**59 days**	**Tue 9/19/17**	**Thu 12/7/17**
✓		**Project Plan**	**11 days**	**Tue 9/19/17**	**Mon 10/2/17**
✓		Team Meeting (During Class)	1 day	Tue 9/26/17	Tue 9/26/17
✓		Status Report 1	1 day	Thu 9/28/17	Thu 9/28/17
✓		**Requirements Specifications**	**12 days**	**Tue 10/3/17**	**Wed 10/18/17**
✓		Team Meeting (During Class)	1 day	Tue 10/10/17	Tue 10/10/17
✓		Status Report 2	1 day	Thu 10/12/17	Thu 10/12/17
✓		**Preliminary Design**	**15 days**	**Tue 10/17/17**	**Mon 11/6/17**
✓		Status Report 3	1 day	Thu 10/26/17	Thu 10/26/17
✓		Requirements Review	3 days	Tue 10/24/17	Thu 10/26/17
✓		Team Meeting (During Class)	1 day	Tue 10/31/17	Tue 10/31/17
✓		**Detailed Design**	**20 days**	**Tue 11/7/17**	**Mon 12/4/17**
✓		Status Report 4	1 day	Thu 11/9/17	Thu 11/9/17
✓		Team Meeting (During Class)	1 day	Tue 11/14/17	Tue 11/14/17
✓		Wrapup Lecture	1 day	Tue 11/28/17	Tue 11/28/17
✓		Status Report 5	1 day	Thu 11/30/17	Thu 11/30/17
✓		Design Review	3 days	Tue 12/5/17	Thu 12/7/17
✓		Status Report 6	1 day	Thu 12/7/17	Thu 12/7/17



Bi-Weekly Status Report

CSCE 4905

Date: 17 January 2017
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous ~~2-week~~ winter break

- Finalized the Programming Language to be used- Python, mastered it and learned it well
- Got the server set up and ready to go
- Began working on the wireframe/layout of the front-end
- Began researching sensors, ran into some problems though so one could not be purchased.

Problems Encountered

- Originally, we wanted a test node that would measure Nitrogen Dioxide, Carbon Monoxide, Carbon Dioxide, Sulfur Dioxide, and two forms of Particulate Matter similar to what the state uses for their sensors. However, we are finding that a sensor that is capable of doing that is really expensive and not feasible for a consumer or for this project's budget. We're researching the different types of things that need to be measured, why they need to be measured, and picking a top three to recommend to our clients, and will provide them with our research.
  As far as the sensor is concerned. We will either A) Find one that measures the three qualities that we want to measure, or B) create a raspberry pi hat using sensors from adafruit. It will likely end up boiling down to cost and availability. We have a meeting set up to discuss this with the client, provide a recommendation, and get their input on it.

Plans for the upcoming 2-week period

- Solve the sensor issue
- Meeting with the client on February 1st

Project Plan Reconciliation
Project Plan Gantt Chart is currently a work in progress, but we are on track with our plans that were previously made.

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
December 19th- Hangouts Call	Present	Present	Present	Present
January 9th- Hangouts Call	Present	Present	Present	Present
January 16th- Class	Present	Present	Present	Present

Bi-Weekly Status Report

CSCE 4925

Date: 1 February 2017
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2-weeks

- Created wireframes and presented these to the client
- Completed revisions on the detailed design and the project plan
- Solved the sensor issue from last time- Can use any sensor that measures anything, or scrape data from the state's website.

Problems Encountered

- None

Plans for the upcoming 2-week period

- Set up PostgreSQL server
- Set up a localhost page to confirm connectivity
- Set up Git Repos
- Set up the tech stack
- Create database tables
- Create a draft version of the test plan

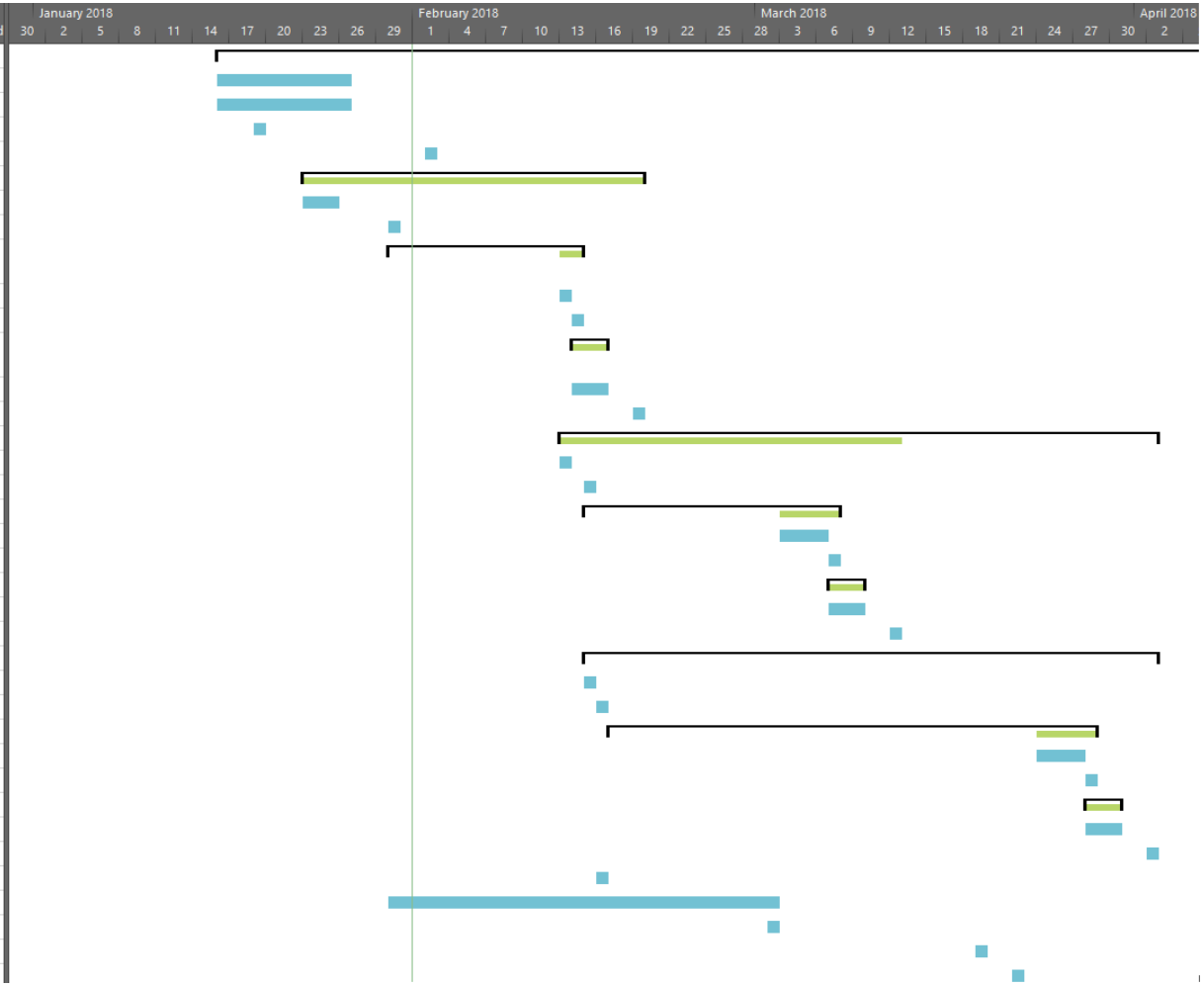
Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
January 23rd- Class	Present	Present	Present	Present
January 30th- Class/Team Meeting	Present	Present	Present	Present
February 2nd- Meeting with the Client	Present	Present	Present	Present

Project Plan Reconciliation

(See Next Page)

Task Mode	Task Name	Duration	Start	Finish	Pred
	Spring Schedule	**78 days**	**Tue 1/16/18**	**Thu 5/3/18**	
	Project Plan Updates	9 days	Tue 1/16/18	Fri 1/26/18	
	Detailed Design Updates	9 days	Tue 1/16/18	Fri 1/26/18	
	Status Report 1	1 day	Fri 1/19/18	Fri 1/19/18	
	Status Report 2	1 day	Fri 2/2/18	Fri 2/2/18	
	Test Plan and Procedure	**20 days**	**Tue 1/23/18**	**Mon 2/19/18**	
	Lecture	3 days	Tue 1/23/18	Thu 1/25/18	
	Test Plan and Procedure Plan	1 day	Tue 1/30/18	Tue 1/30/18	
	Test Plan and Procedure Plan Draft	**12 days**	**Tue 1/30/18**	**Wed 2/14/18**	
	Document Compilation	1 day	Tue 2/13/18	Tue 2/13/18	
	Document Editing	1 day	Wed 2/14/18	Wed 2/14/18	
	Test Plan and Procedure Final Draft	**3 days**	**Wed 2/14/18**	**Fri 2/16/18**	
	Necessary Revisions	3 days	Wed 2/14/18	Fri 2/16/18	
	Test Plan and Procedure Report	1 day	Mon 2/19/18	Mon 2/19/18	
	Maintenance Manual	**35 days**	**Tue 2/13/18**	**Mon 4/2/18**	
	Lecture	1 day	Tue 2/13/18	Tue 2/13/18	
	Maintenance Manual Plan	1 day	Thu 2/15/18	Thu 2/15/18	
	Maintenance Manual Draft	**15 days**	**Thu 2/15/18**	**Wed 3/7/18**	
	Document Compilation	3 days	Sat 3/3/18	Tue 3/6/18	
	Document Editing	1 day	Wed 3/7/18	Wed 3/7/18	
	Maintenance Manual Final Draft	**3 days**	**Wed 3/7/18**	**Fri 3/9/18**	
	Necessary Revisions	3 days	Wed 3/7/18	Fri 3/9/18	
	Maintenance Manual Report	1 day	Mon 3/12/18	Mon 3/12/18	
	User Manual	**33 days**	**Thu 2/15/18**	**Mon 4/2/18**	
	Lecture	1 day	Thu 2/15/18	Thu 2/15/18	
	User Manual Plan	1 day	Fri 2/16/18	Fri 2/16/18	
	User Manual Draft	**29 days**	**Sat 2/17/18**	**Wed 3/28/18**	
	Document Compilation	3 days	Sat 3/24/18	Tue 3/27/18	
	Document Editing	1 day	Wed 3/28/18	Wed 3/28/18	
	User Manual Final Draft	**3 days**	**Wed 3/28/18**	**Fri 3/30/18**	
	Necessary Revisions	3 days	Wed 3/28/18	Fri 3/30/18	
	User Manual Report	1 day	Mon 4/2/18	Mon 4/2/18	
	Status Report 3	1 day	Fri 2/16/18	Fri 2/16/18	
	Application Prototype	24 days	Tue 1/30/18	Fri 3/2/18	
	Status Report 4	1 day	Fri 3/2/18	Fri 3/2/18	
	Status Report 5	1 day	Mon 3/19/18	Mon 3/19/18	
	Test Readiness Presentation	1 day	Thu 3/22/18	Thu 3/22/18	



Bi-Weekly Status Report 3

CSCE 4925

Date: 16 February 2018
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2 weeks

- Set up a PostgreSQL Server
- Set up a localhost page to confirm connectivity
- Set Up the tech stack
- Created Database tables
- Found a backup solution for the server
- Completed work on the Test Plan
- Reformatted and revised the Detailed Design document

Problems Encountered

- Setting up the localhost page proved quite difficult

Plans for the upcoming 2-week period

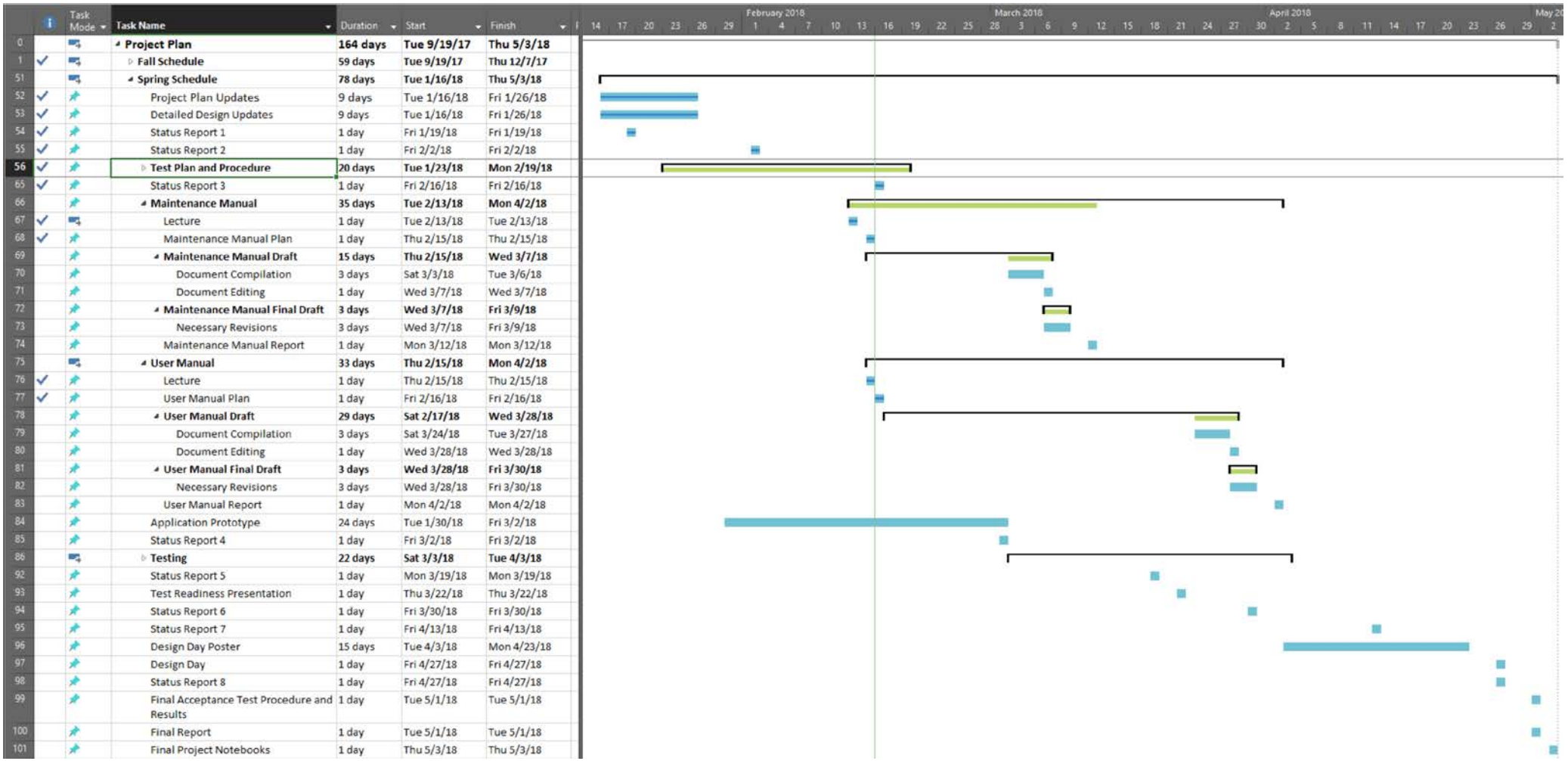
- Skeleton of the frontend
- Database setup completed
- Find some sensors compatible with Raspberry Pi
- Progress on the API
- Make progress on scraping data from the state's website.
- Begin work on the Maintenance Manual

Project Plan Reconciliation

- Modified the Project Plan Gantt Chart to include the test plan schedule
- Gantt chart included on the following pages
- We are on schedule currently

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
February 13th- Class	Present	Present	Present	Present
February 15th- Class	Present	Present	Absent (Also excused)	Present



	Task Mode	Task Name	Duration	Start	Finish
0		**Project Plan**	**164 days**	**Tue 9/19/17**	**Thu 5/3/18**
1	✓	**Fall Schedule**	**59 days**	**Tue 9/19/17**	**Thu 12/7/17**
51		**Spring Schedule**	**78 days**	**Tue 1/16/18**	**Thu 5/3/18**
52	✓	Project Plan Updates	9 days	Tue 1/16/18	Fri 1/26/18
53	✓	Detailed Design Updates	9 days	Tue 1/16/18	Fri 1/26/18
54	✓	Status Report 1	1 day	Fri 1/19/18	Fri 1/19/18
55	✓	Status Report 2	1 day	Fri 2/2/18	Fri 2/2/18
56	✓	**Test Plan and Procedure**	**20 days**	**Tue 1/23/18**	**Mon 2/19/18**
65	✓	Status Report 3	1 day	Fri 2/16/18	Fri 2/16/18
66		**Maintenance Manual**	**35 days**	**Tue 2/13/18**	**Mon 4/2/18**
67	✓	Lecture	1 day	Tue 2/13/18	Tue 2/13/18
68	✓	Maintenance Manual Plan	1 day	Thu 2/15/18	Thu 2/15/18
69		**Maintenance Manual Draft**	**15 days**	**Thu 2/15/18**	**Wed 3/7/18**
70		Document Compilation	3 days	Sat 3/3/18	Tue 3/6/18
71		Document Editing	1 day	Wed 3/7/18	Wed 3/7/18
72		**Maintenance Manual Final Draft**	**3 days**	**Wed 3/7/18**	**Fri 3/9/18**
73		Necessary Revisions	3 days	Wed 3/7/18	Fri 3/9/18
74		Maintenance Manual Report	1 day	Mon 3/12/18	Mon 3/12/18
75		**User Manual**	**33 days**	**Thu 2/15/18**	**Mon 4/2/18**
76	✓	Lecture	1 day	Thu 2/15/18	Thu 2/15/18
77	✓	User Manual Plan	1 day	Fri 2/16/18	Fri 2/16/18
78		**User Manual Draft**	**29 days**	**Sat 2/17/18**	**Wed 3/28/18**
79		Document Compilation	3 days	Sat 3/24/18	Tue 3/27/18
80		Document Editing	1 day	Wed 3/28/18	Wed 3/28/18
81		**User Manual Final Draft**	**3 days**	**Wed 3/28/18**	**Fri 3/30/18**
82		Necessary Revisions	3 days	Wed 3/28/18	Fri 3/30/18
83		User Manual Report	1 day	Mon 4/2/18	Mon 4/2/18
84		Application Prototype	24 days	Tue 1/30/18	Fri 3/2/18
85		Status Report 4	1 day	Fri 3/2/18	Fri 3/2/18
86		**Testing**	**22 days**	**Sat 3/3/18**	**Tue 4/3/18**
92		Status Report 5	1 day	Mon 3/19/18	Mon 3/19/18
93		Test Readiness Presentation	1 day	Thu 3/22/18	Thu 3/22/18
94		Status Report 6	1 day	Fri 3/30/18	Fri 3/30/18
95		Status Report 7	1 day	Fri 4/13/18	Fri 4/13/18
96		Design Day Poster	15 days	Tue 4/3/18	Mon 4/23/18
97		Design Day	1 day	Fri 4/27/18	Fri 4/27/18
98		Status Report 8	1 day	Fri 4/27/18	Fri 4/27/18
99		Final Acceptance Test Procedure and Results	1 day	Tue 5/1/18	Tue 5/1/18
100		Final Report	1 day	Tue 5/1/18	Tue 5/1/18
101		Final Project Notebooks	1 day	Thu 5/3/18	Thu 5/3/18

Bi-Weekly Status Report 4

CSCE 4925

Date: 16 February 2018
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2 weeks

- Finally got the UI up and running using Gnome and Real VNC Viewer
- Making Progress on the Maintenance manual

Problems Encountered

- The UI was a lot more difficult than expected
- Two group members got sick at the same time, so we experienced a resource shortage. We expect to be back on track within the next week, pending the recovery of one member.

Plans for the upcoming 2-week period

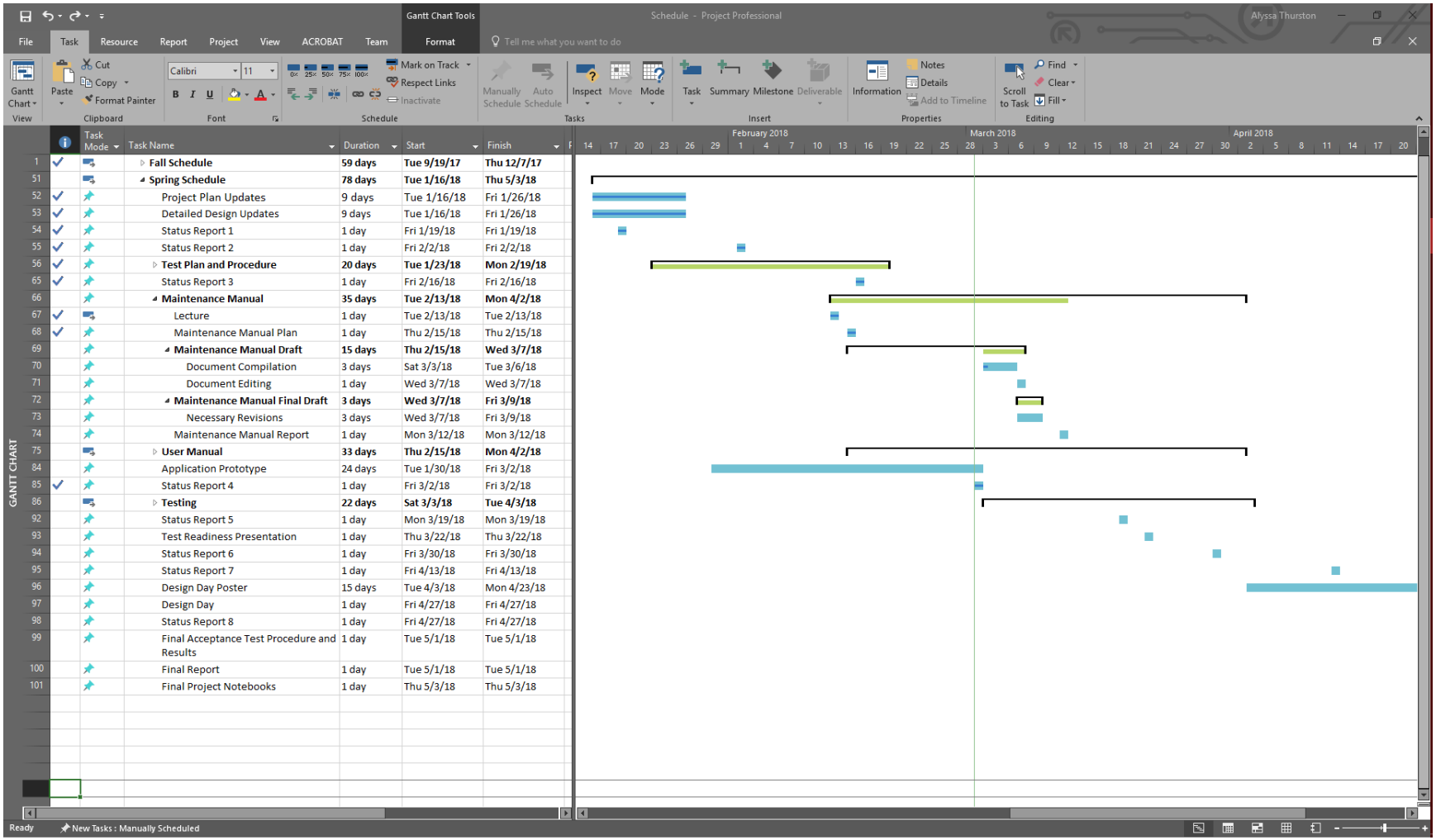
- Completing tasks and continuing work on tasks from previous status reports that have not yet been complete

Project Plan Reconciliation

- Gantt chart included on the following pages
- We are on schedule currently (according to the Gantt Chart, in reality we are behind)

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
February 20th- Flight Meeting	Present	Absent (excused)	Present	Present


Gantt Chart Tools
Schedule - Project Professional
Alyssa Thurston
File
Task
Resource
Report
Project
View
ACROBAT
Team
Format
Tell me what you want to do
Gantt Chart
Paste
Cut
Copy
Format Painter
Calibri
11
Mark on Track
Respect Links
Inactivate
Manually Schedule
Auto Schedule
Inspect
Move
Mode
Task
Summary
Milestone
Deliverable
Information
Notes
Details
Add to Timeline
Scroll to Task
Find
Clear
Fill
View
Clipboard
Font
Schedule
Tasks
Insert
Properties
Editing
GANTT CHART
Task Mode
Task Name
Duration
Start
Finish
February 2018
March 2018
April 2018
1 Fall Schedule 59 days Tue 9/19/17 Thu 12/7/17
51 Spring Schedule 78 days Tue 1/16/18 Thu 5/3/18
52 Project Plan Updates 9 days Tue 1/16/18 Fri 1/26/18
53 Detailed Design Updates 9 days Tue 1/16/18 Fri 1/26/18
54 Status Report 1 1 day Fri 1/19/18 Fri 1/19/18
55 Status Report 2 1 day Fri 2/2/18 Fri 2/2/18
56 Test Plan and Procedure 20 days Tue 1/23/18 Mon 2/19/18
65 Status Report 3 1 day Fri 2/16/18 Fri 2/16/18
66 Maintenance Manual 35 days Tue 2/13/18 Mon 4/2/18
67 Lecture 1 day Tue 2/13/18 Tue 2/13/18
68 Maintenance Manual Plan 1 day Thu 2/15/18 Thu 2/15/18
69 Maintenance Manual Draft 15 days Thu 2/15/18 Wed 3/7/18
70 Document Compilation 3 days Sat 3/3/18 Tue 3/6/18
71 Document Editing 1 day Wed 3/7/18 Wed 3/7/18
72 Maintenance Manual Final Draft 3 days Wed 3/7/18 Fri 3/9/18
73 Necessary Revisions 3 days Wed 3/7/18 Fri 3/9/18
74 Maintenance Manual Report 1 day Mon 3/12/18 Mon 3/12/18
75 User Manual 33 days Thu 2/15/18 Mon 4/2/18
84 Application Prototype 24 days Tue 1/30/18 Fri 3/2/18
85 Status Report 4 1 day Fri 3/2/18 Fri 3/2/18
86 Testing 22 days Sat 3/3/18 Tue 4/3/18
92 Status Report 5 1 day Mon 3/19/18 Mon 3/19/18
93 Test Readiness Presentation 1 day Thu 3/22/18 Thu 3/22/18
94 Status Report 6 1 day Fri 3/30/18 Fri 3/30/18
95 Status Report 7 1 day Fri 4/13/18 Fri 4/13/18
96 Design Day Poster 15 days Tue 4/3/18 Mon 4/23/18
97 Design Day 1 day Fri 4/27/18 Fri 4/27/18
98 Status Report 8 1 day Fri 4/27/18 Fri 4/27/18
99 Final Acceptance Test Procedure and Results 1 day Tue 5/1/18 Tue 5/1/18
100 Final Report 1 day Tue 5/1/18 Tue 5/1/18
101 Final Project Notebooks 1 day Thu 5/3/18 Thu 5/3/18
Ready
New Tasks : Manually Scheduled

Bi-Weekly Status Report 4

CSCE 4925

Date: 19 March 2018
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2 weeks

- Completed skeletal work on a test localhost page, working to refine the UI further
- Database tables have been created, and a test table has been made and filled with dummy data.
- Making Progress on the User Manual

Problems Encountered

- Differentiating data types from each other without limiting the types of data that can be added and contributed to the network. We're still working on finding a solution to this issue.

Plans for the upcoming 2-week period

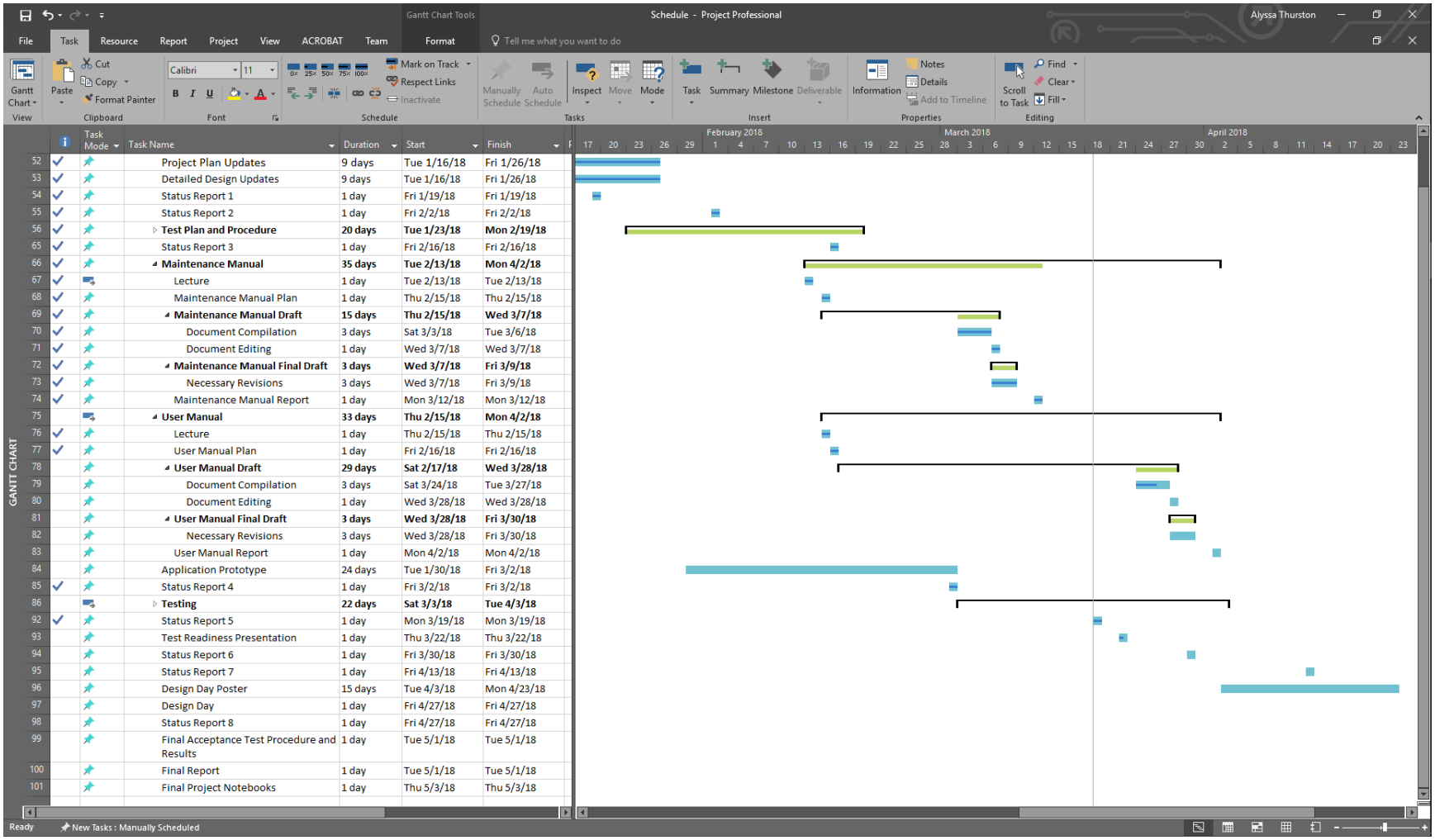
- Work on the UI
- Work on the API
- Work on integrating Google Maps API into our front end
- Work on a web scraper to get more legitimate data into the front end.
- Complete work on an Arduino based Node, all the supplies have been ordered it's just a matter of assembling and programming them.

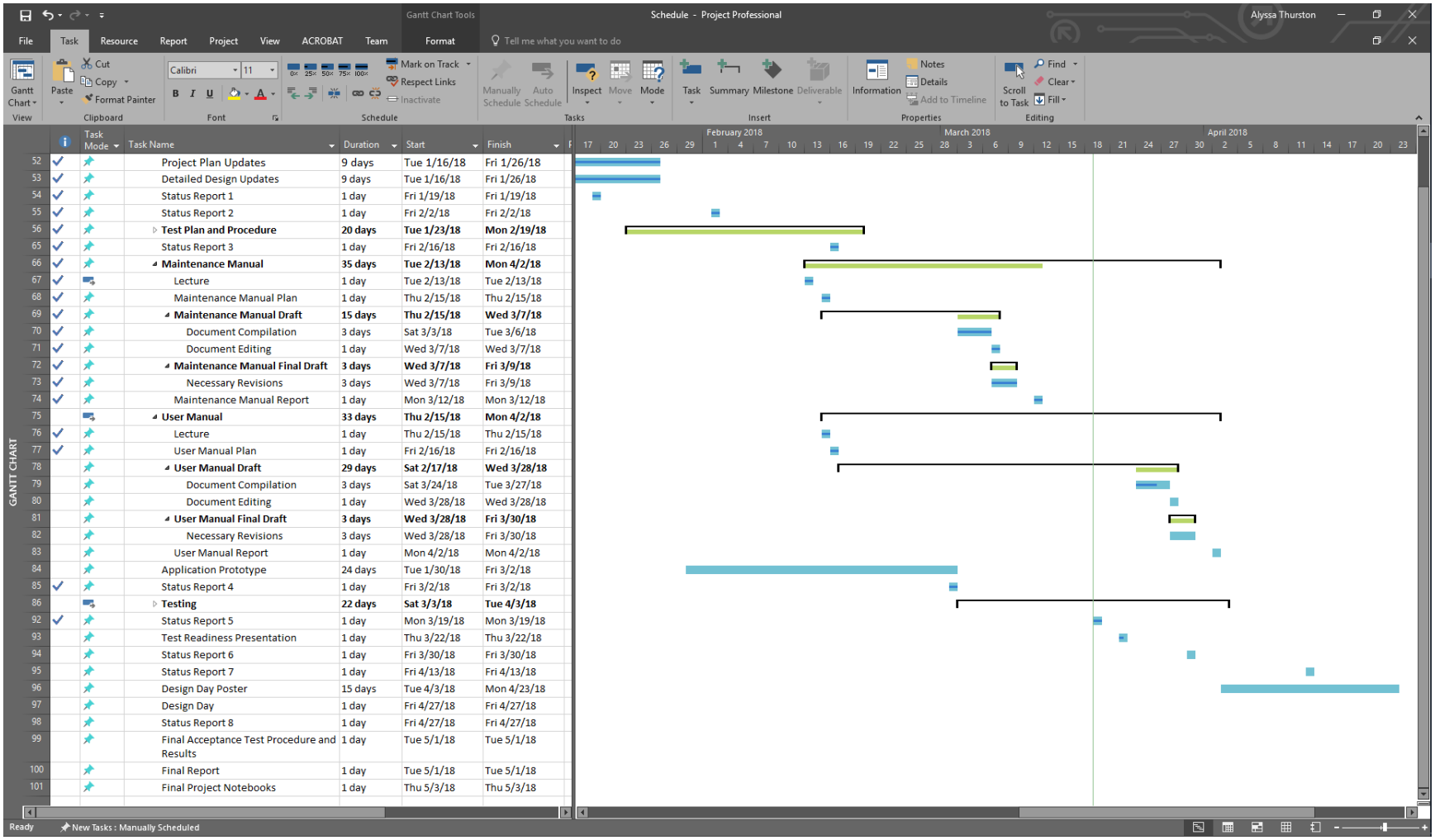
Project Plan Reconciliation

- Gantt chart included on the following pages
- We are falling behind schedule on the development side of things.

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
March 6th - Flight Meeting	Present	Present	Present	Present


Schedule - Project Professional
Gantt Chart Tools
Alyssa Thurston
File
Task
Resource
Report
Project
View
ACROBAT
Team
Format
Tell me what you want to do
Gantt Chart
Paste
Cut
Copy
Format Painter
Calibri
11
Mark on Track
Respect Links
Inactivate
Manually Schedule
Auto Schedule
Inspect
Move
Mode
Task
Summary
Milestone
Deliverable
Information
Notes
Details
Add to Timeline
Scroll to Task
Find
Clear
Fill
View
Clipboard
Font
Schedule
Tasks
Insert
Properties
Editing
GANTT CHART
Task Mode
Task Name
Duration
Start
Finish
February 2018
March 2018
April 2018
52 Project Plan Updates 9 days Tue 1/16/18 Fri 1/26/18
53 Detailed Design Updates 9 days Tue 1/16/18 Fri 1/26/18
54 Status Report 1 1 day Fri 1/19/18 Fri 1/19/18
55 Status Report 2 1 day Fri 2/2/18 Fri 2/2/18
56 Test Plan and Procedure 20 days Tue 1/23/18 Mon 2/19/18
65 Status Report 3 1 day Fri 2/16/18 Fri 2/16/18
66 Maintenance Manual 35 days Tue 2/13/18 Mon 4/2/18
67 Lecture 1 day Tue 2/13/18 Tue 2/13/18
68 Maintenance Manual Plan 1 day Thu 2/15/18 Thu 2/15/18
69 Maintenance Manual Draft 15 days Thu 2/15/18 Wed 3/7/18
70 Document Compilation 3 days Sat 3/3/18 Tue 3/6/18
71 Document Editing 1 day Wed 3/7/18 Wed 3/7/18
72 Maintenance Manual Final Draft 3 days Wed 3/7/18 Fri 3/9/18
73 Necessary Revisions 3 days Wed 3/7/18 Fri 3/9/18
74 Maintenance Manual Report 1 day Mon 3/12/18 Mon 3/12/18
75 User Manual 33 days Thu 2/15/18 Mon 4/2/18
76 Lecture 1 day Thu 2/15/18 Thu 2/15/18
77 User Manual Plan 1 day Fri 2/16/18 Fri 2/16/18
78 User Manual Draft 29 days Sat 2/17/18 Wed 3/28/18
79 Document Compilation 3 days Sat 3/24/18 Tue 3/27/18
80 Document Editing 1 day Wed 3/28/18 Wed 3/28/18
81 User Manual Final Draft 3 days Wed 3/28/18 Fri 3/30/18
82 Necessary Revisions 3 days Wed 3/28/18 Fri 3/30/18
83 User Manual Report 1 day Mon 4/2/18 Mon 4/2/18
84 Application Prototype 24 days Tue 1/30/18 Fri 3/2/18
85 Status Report 4 1 day Fri 3/2/18 Fri 3/2/18
86 Testing 22 days Sat 3/3/18 Tue 4/3/18
92 Status Report 5 1 day Mon 3/19/18 Mon 3/19/18
93 Test Readiness Presentation 1 day Thu 3/22/18 Thu 3/22/18
94 Status Report 6 1 day Fri 3/30/18 Fri 3/30/18
95 Status Report 7 1 day Fri 4/13/18 Fri 4/13/18
96 Design Day Poster 15 days Tue 4/3/18 Mon 4/23/18
97 Design Day 1 day Fri 4/27/18 Fri 4/27/18
98 Status Report 8 1 day Fri 4/27/18 Fri 4/27/18
99 Final Acceptance Test Procedure and Results 1 day Tue 5/1/18 Tue 5/1/18
100 Final Report 1 day Tue 5/1/18 Tue 5/1/18
101 Final Project Notebooks 1 day Thu 5/3/18 Thu 5/3/18
Ready
New Tasks : Manually Scheduled


Schedule - Project Professional
Gantt Chart Tools
Alyssa Thurston
File
Task
Resource
Report
Project
View
ACROBAT
Team
Format
Tell me what you want to do
Gantt Chart
Paste
Cut
Copy
Format Painter
Calibri
11
Mark on Track
Respect Links
Inactivate
Manually Schedule
Auto Schedule
Inspect
Move
Mode
Task
Summary
Milestone
Deliverable
Information
Notes
Details
Add to Timeline
Scroll to Task
Find
Clear
Fill
View
Clipboard
Font
Schedule
Tasks
Insert
Properties
Editing
GANTT CHART
Task Mode
Task Name
Duration
Start
Finish
February 2018
March 2018
April 2018
52 Project Plan Updates 9 days Tue 1/16/18 Fri 1/26/18
53 Detailed Design Updates 9 days Tue 1/16/18 Fri 1/26/18
54 Status Report 1 1 day Fri 1/19/18 Fri 1/19/18
55 Status Report 2 1 day Fri 2/2/18 Fri 2/2/18
56 Test Plan and Procedure 20 days Tue 1/23/18 Mon 2/19/18
65 Status Report 3 1 day Fri 2/16/18 Fri 2/16/18
66 Maintenance Manual 35 days Tue 2/13/18 Mon 4/2/18
67 Lecture 1 day Tue 2/13/18 Tue 2/13/18
68 Maintenance Manual Plan 1 day Thu 2/15/18 Thu 2/15/18
69 Maintenance Manual Draft 15 days Thu 2/15/18 Wed 3/7/18
70 Document Compilation 3 days Sat 3/3/18 Tue 3/6/18
71 Document Editing 1 day Wed 3/7/18 Wed 3/7/18
72 Maintenance Manual Final Draft 3 days Wed 3/7/18 Fri 3/9/18
73 Necessary Revisions 3 days Wed 3/7/18 Fri 3/9/18
74 Maintenance Manual Report 1 day Mon 3/12/18 Mon 3/12/18
75 User Manual 33 days Thu 2/15/18 Mon 4/2/18
76 Lecture 1 day Thu 2/15/18 Thu 2/15/18
77 User Manual Plan 1 day Fri 2/16/18 Fri 2/16/18
78 User Manual Draft 29 days Sat 2/17/18 Wed 3/28/18
79 Document Compilation 3 days Sat 3/24/18 Tue 3/27/18
80 Document Editing 1 day Wed 3/28/18 Wed 3/28/18
81 User Manual Final Draft 3 days Wed 3/28/18 Fri 3/30/18
82 Necessary Revisions 3 days Wed 3/28/18 Fri 3/30/18
83 User Manual Report 1 day Mon 4/2/18 Mon 4/2/18
84 Application Prototype 24 days Tue 1/30/18 Fri 3/2/18
85 Status Report 4 1 day Fri 3/2/18 Fri 3/2/18
86 Testing 22 days Sat 3/3/18 Tue 4/3/18
92 Status Report 5 1 day Mon 3/19/18 Mon 3/19/18
93 Test Readiness Presentation 1 day Thu 3/22/18 Thu 3/22/18
94 Status Report 6 1 day Fri 3/30/18 Fri 3/30/18
95 Status Report 7 1 day Fri 4/13/18 Fri 4/13/18
96 Design Day Poster 15 days Tue 4/3/18 Mon 4/23/18
97 Design Day 1 day Fri 4/27/18 Fri 4/27/18
98 Status Report 8 1 day Fri 4/27/18 Fri 4/27/18
99 Final Acceptance Test Procedure and Results 1 day Tue 5/1/18 Tue 5/1/18
100 Final Report 1 day Tue 5/1/18 Tue 5/1/18
101 Final Project Notebooks 1 day Thu 5/3/18 Thu 5/3/18
Ready
New Tasks : Manually Scheduled

Bi-Weekly Status Report 6

CSCE 4925

Date: 30 March 2018
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2 weeks

- Created a node

Problems Encountered

- Accessing the server from the node, the server is behind a firewall and we don't have control over the firewall so we are migrating our server from a linux server to a windows server. This process is almost complete

Plans for the upcoming 2-week period

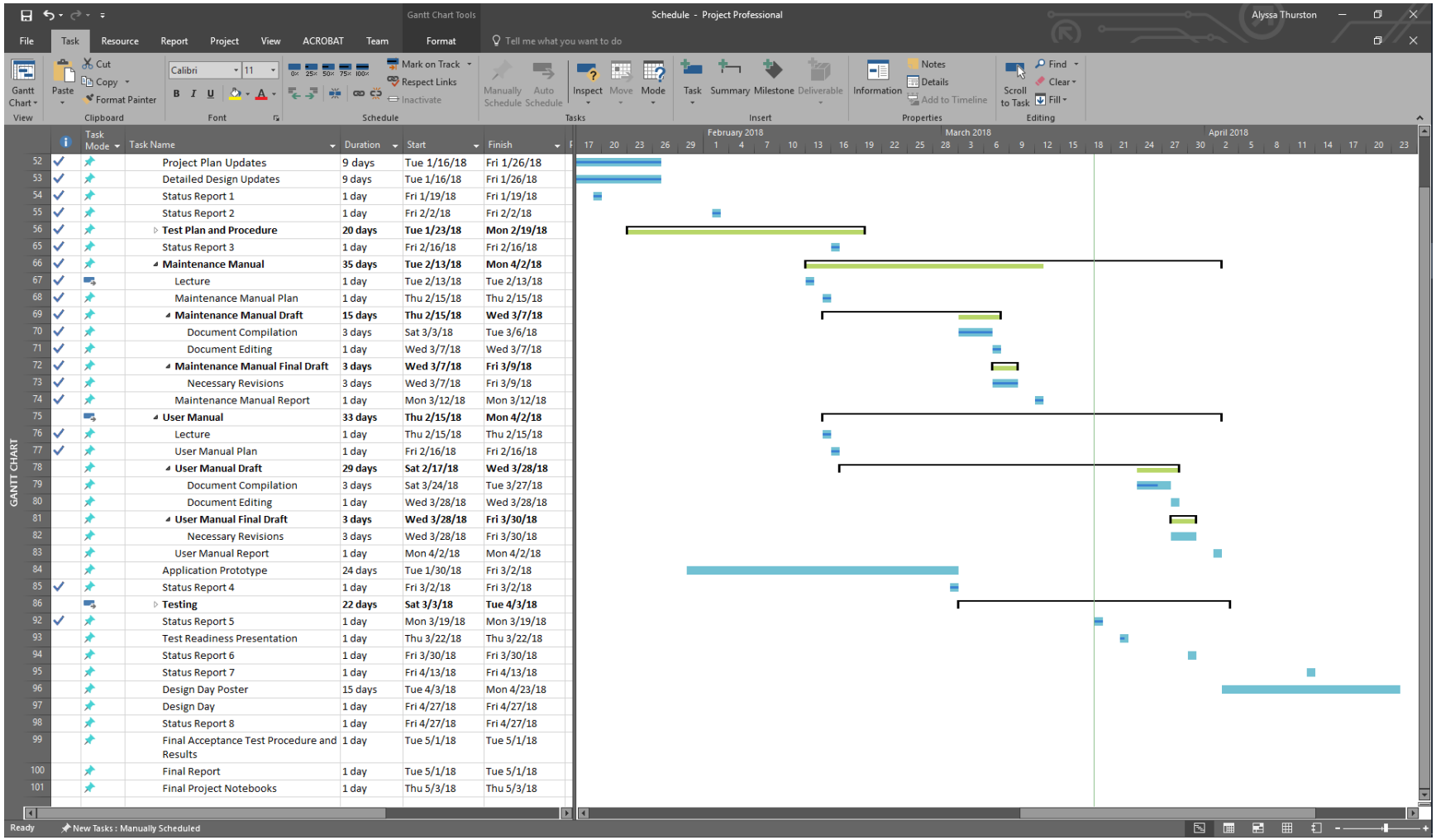
- Work on the UI
- Work on the API
- Work on integrating Google Maps API into our front end
- Work on a web scraper to get more legitimate data into the front end.

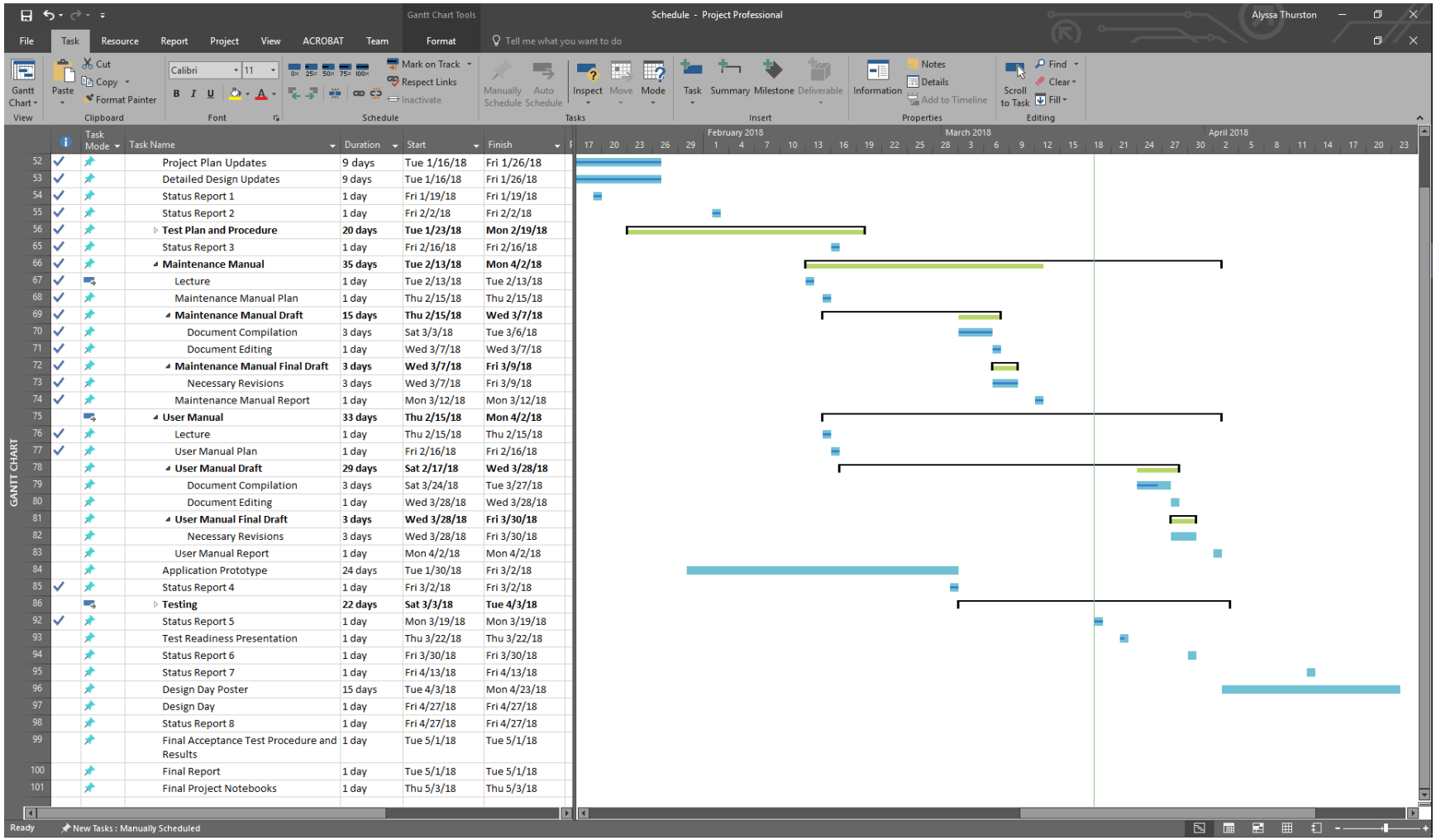
Project Plan Reconciliation

- Gantt chart included on the following pages
- We are falling behind schedule on the development side of things.

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
March 27th - Lecture	Present	Present	Present	Present


Gantt Chart Tools
Schedule - Project Professional
Alyssa Thurston
File
Task
Resource
Report
Project
View
ACROBAT
Team
Format
Tell me what you want to do
Gantt Chart
Paste
Cut
Copy
Format Painter
Calibri
11
Mark on Track
Respect Links
Inactivate
Manually Schedule
Auto Schedule
Inspect
Move
Mode
Task
Summary
Milestone
Deliverable
Information
Notes
Details
Add to Timeline
Scroll to Task
Find
Clear
Fill
View
Clipboard
Font
Schedule
Tasks
Insert
Properties
Editing
GANTT CHART
Task Mode
Task Name
Duration
Start
Finish
February 2018
March 2018
April 2018
52 Project Plan Updates 9 days Tue 1/16/18 Fri 1/26/18
53 Detailed Design Updates 9 days Tue 1/16/18 Fri 1/26/18
54 Status Report 1 1 day Fri 1/19/18 Fri 1/19/18
55 Status Report 2 1 day Fri 2/2/18 Fri 2/2/18
56 Test Plan and Procedure 20 days Tue 1/23/18 Mon 2/19/18
65 Status Report 3 1 day Fri 2/16/18 Fri 2/16/18
66 Maintenance Manual 35 days Tue 2/13/18 Mon 4/2/18
67 Lecture 1 day Tue 2/13/18 Tue 2/13/18
68 Maintenance Manual Plan 1 day Thu 2/15/18 Thu 2/15/18
69 Maintenance Manual Draft 15 days Thu 2/15/18 Wed 3/7/18
70 Document Compilation 3 days Sat 3/3/18 Tue 3/6/18
71 Document Editing 1 day Wed 3/7/18 Wed 3/7/18
72 Maintenance Manual Final Draft 3 days Wed 3/7/18 Fri 3/9/18
73 Necessary Revisions 3 days Wed 3/7/18 Fri 3/9/18
74 Maintenance Manual Report 1 day Mon 3/12/18 Mon 3/12/18
75 User Manual 33 days Thu 2/15/18 Mon 4/2/18
76 Lecture 1 day Thu 2/15/18 Thu 2/15/18
77 User Manual Plan 1 day Fri 2/16/18 Fri 2/16/18
78 User Manual Draft 29 days Sat 2/17/18 Wed 3/28/18
79 Document Compilation 3 days Sat 3/24/18 Tue 3/27/18
80 Document Editing 1 day Wed 3/28/18 Wed 3/28/18
81 User Manual Final Draft 3 days Wed 3/28/18 Fri 3/30/18
82 Necessary Revisions 3 days Wed 3/28/18 Fri 3/30/18
83 User Manual Report 1 day Mon 4/2/18 Mon 4/2/18
84 Application Prototype 24 days Tue 1/30/18 Fri 3/2/18
85 Status Report 4 1 day Fri 3/2/18 Fri 3/2/18
86 Testing 22 days Sat 3/3/18 Tue 4/3/18
92 Status Report 5 1 day Mon 3/19/18 Mon 3/19/18
93 Test Readiness Presentation 1 day Thu 3/22/18 Thu 3/22/18
94 Status Report 6 1 day Fri 3/30/18 Fri 3/30/18
95 Status Report 7 1 day Fri 4/13/18 Fri 4/13/18
96 Design Day Poster 15 days Tue 4/3/18 Mon 4/23/18
97 Design Day 1 day Fri 4/27/18 Fri 4/27/18
98 Status Report 8 1 day Fri 4/27/18 Fri 4/27/18
99 Final Acceptance Test Procedure and Results 1 day Tue 5/1/18 Tue 5/1/18
100 Final Report 1 day Tue 5/1/18 Tue 5/1/18
101 Final Project Notebooks 1 day Thu 5/3/18 Thu 5/3/18
Ready
New Tasks : Manually Scheduled


Gantt Chart Tools
Schedule - Project Professional
Alyssa Thurston
File
Task
Resource
Report
Project
View
ACROBAT
Team
Format
Tell me what you want to do
Gantt Chart
Paste
Cut
Copy
Format Painter
Calibri
11
Mark on Track
Respect Links
Inactivate
Manually Schedule
Auto Schedule
Inspect
Move
Mode
Task
Summary
Milestone
Deliverable
Information
Notes
Details
Add to Timeline
Scroll to Task
Find
Clear
Fill
View
Clipboard
Font
Schedule
Tasks
Insert
Properties
Editing
GANTT CHART
Task Mode
Task Name
Duration
Start
Finish
February 2018
March 2018
April 2018
52 Project Plan Updates 9 days Tue 1/16/18 Fri 1/26/18
53 Detailed Design Updates 9 days Tue 1/16/18 Fri 1/26/18
54 Status Report 1 1 day Fri 1/19/18 Fri 1/19/18
55 Status Report 2 1 day Fri 2/2/18 Fri 2/2/18
56 Test Plan and Procedure 20 days Tue 1/23/18 Mon 2/19/18
65 Status Report 3 1 day Fri 2/16/18 Fri 2/16/18
66 Maintenance Manual 35 days Tue 2/13/18 Mon 4/2/18
67 Lecture 1 day Tue 2/13/18 Tue 2/13/18
68 Maintenance Manual Plan 1 day Thu 2/15/18 Thu 2/15/18
69 Maintenance Manual Draft 15 days Thu 2/15/18 Wed 3/7/18
70 Document Compilation 3 days Sat 3/3/18 Tue 3/6/18
71 Document Editing 1 day Wed 3/7/18 Wed 3/7/18
72 Maintenance Manual Final Draft 3 days Wed 3/7/18 Fri 3/9/18
73 Necessary Revisions 3 days Wed 3/7/18 Fri 3/9/18
74 Maintenance Manual Report 1 day Mon 3/12/18 Mon 3/12/18
75 User Manual 33 days Thu 2/15/18 Mon 4/2/18
76 Lecture 1 day Thu 2/15/18 Thu 2/15/18
77 User Manual Plan 1 day Fri 2/16/18 Fri 2/16/18
78 User Manual Draft 29 days Sat 2/17/18 Wed 3/28/18
79 Document Compilation 3 days Sat 3/24/18 Tue 3/27/18
80 Document Editing 1 day Wed 3/28/18 Wed 3/28/18
81 User Manual Final Draft 3 days Wed 3/28/18 Fri 3/30/18
82 Necessary Revisions 3 days Wed 3/28/18 Fri 3/30/18
83 User Manual Report 1 day Mon 4/2/18 Mon 4/2/18
84 Application Prototype 24 days Tue 1/30/18 Fri 3/2/18
85 Status Report 4 1 day Fri 3/2/18 Fri 3/2/18
86 Testing 22 days Sat 3/3/18 Tue 4/3/18
92 Status Report 5 1 day Mon 3/19/18 Mon 3/19/18
93 Test Readiness Presentation 1 day Thu 3/22/18 Thu 3/22/18
94 Status Report 6 1 day Fri 3/30/18 Fri 3/30/18
95 Status Report 7 1 day Fri 4/13/18 Fri 4/13/18
96 Design Day Poster 15 days Tue 4/3/18 Mon 4/23/18
97 Design Day 1 day Fri 4/27/18 Fri 4/27/18
98 Status Report 8 1 day Fri 4/27/18 Fri 4/27/18
99 Final Acceptance Test Procedure and Results 1 day Tue 5/1/18 Tue 5/1/18
100 Final Report 1 day Tue 5/1/18 Tue 5/1/18
101 Final Project Notebooks 1 day Thu 5/3/18 Thu 5/3/18
Ready
New Tasks : Manually Scheduled

Bi-Weekly Status Report 7

CSCE 4925

Date: 19 March 2018
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2 weeks

- Converted to a Windows Server, which is completely operational.
- All 3 sensors on the node are working simultaneously
- Web scraper is working mostly
- Domain name was purchased and is functional (www.projectaerodenton.com)
- We have a working homepage

Problems Encountered

- Time
- Homepage is proving to be a difficult
- Filtering out information retrieved from the webscraper

Plans for the upcoming 2-week period

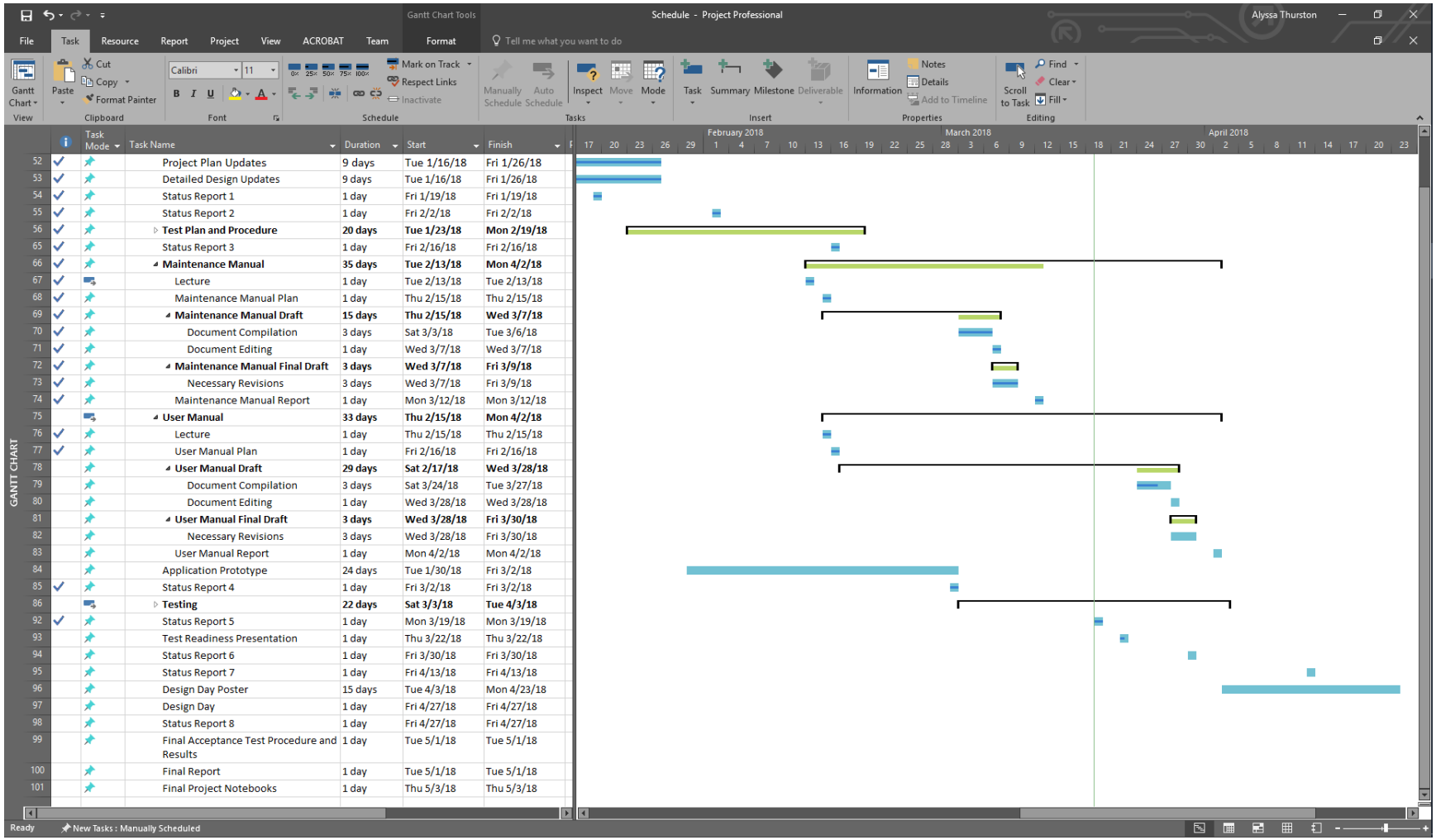
- Work on the UI
- Work on the API
- Work on integrating Google Maps API into our front end
- Work on a web scraper to get more legitimate data into the front end.
- Complete work on an Arduino based Node, all the supplies have been ordered it's just a matter of assembling and programming them.

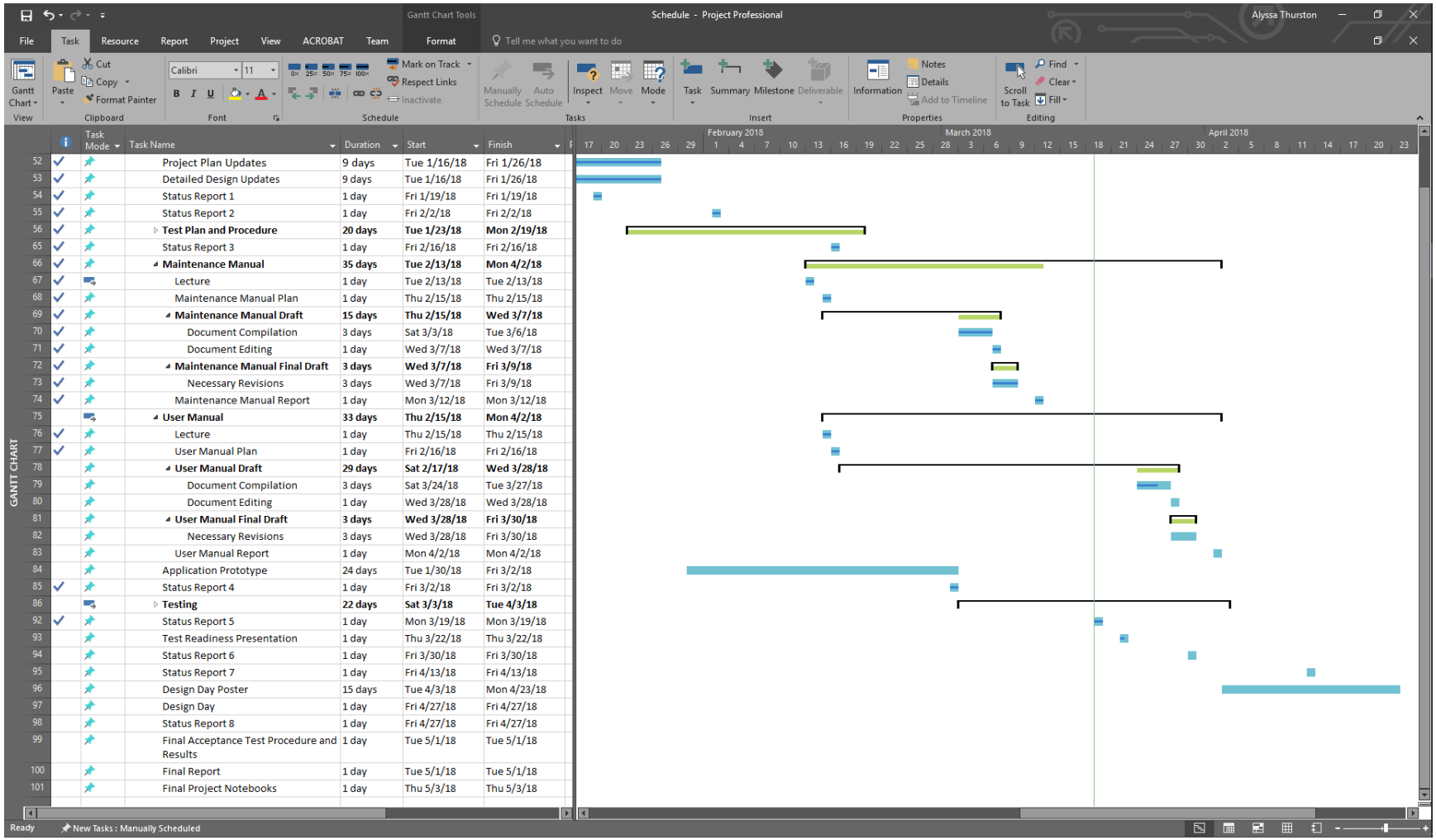
Project Plan Reconciliation

- Gantt chart included on the following pages
- We are falling behind schedule on the development side of things.

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
March 12th	Present	Present	Present	Present
Marth 13th	Present	Present	Present	Present


Gantt Chart Tools
Schedule - Project Professional
Alyssa Thurston
File
Task
Resource
Report
Project
View
ACROBAT
Team
Format
Tell me what you want to do
Gantt Chart
Paste
Cut
Copy
Format Painter
Calibri
11
Mark on Track
Respect Links
Inactivate
Manually Schedule
Auto Schedule
Inspect
Move
Mode
Task
Summary
Milestone
Deliverable
Information
Notes
Details
Add to Timeline
Scroll to Task
Find
Clear
Fill
View
Clipboard
Font
Schedule
Tasks
Insert
Properties
Editing
GANTT CHART
Task Mode
Task Name
Duration
Start
Finish
February 2018
March 2018
April 2018
52 Project Plan Updates 9 days Tue 1/16/18 Fri 1/26/18
53 Detailed Design Updates 9 days Tue 1/16/18 Fri 1/26/18
54 Status Report 1 1 day Fri 1/19/18 Fri 1/19/18
55 Status Report 2 1 day Fri 2/2/18 Fri 2/2/18
56 Test Plan and Procedure 20 days Tue 1/23/18 Mon 2/19/18
65 Status Report 3 1 day Fri 2/16/18 Fri 2/16/18
66 Maintenance Manual 35 days Tue 2/13/18 Mon 4/2/18
67 Lecture 1 day Tue 2/13/18 Tue 2/13/18
68 Maintenance Manual Plan 1 day Thu 2/15/18 Thu 2/15/18
69 Maintenance Manual Draft 15 days Thu 2/15/18 Wed 3/7/18
70 Document Compilation 3 days Sat 3/3/18 Tue 3/6/18
71 Document Editing 1 day Wed 3/7/18 Wed 3/7/18
72 Maintenance Manual Final Draft 3 days Wed 3/7/18 Fri 3/9/18
73 Necessary Revisions 3 days Wed 3/7/18 Fri 3/9/18
74 Maintenance Manual Report 1 day Mon 3/12/18 Mon 3/12/18
75 User Manual 33 days Thu 2/15/18 Mon 4/2/18
76 Lecture 1 day Thu 2/15/18 Thu 2/15/18
77 User Manual Plan 1 day Fri 2/16/18 Fri 2/16/18
78 User Manual Draft 29 days Sat 2/17/18 Wed 3/28/18
79 Document Compilation 3 days Sat 3/24/18 Tue 3/27/18
80 Document Editing 1 day Wed 3/28/18 Wed 3/28/18
81 User Manual Final Draft 3 days Wed 3/28/18 Fri 3/30/18
82 Necessary Revisions 3 days Wed 3/28/18 Fri 3/30/18
83 User Manual Report 1 day Mon 4/2/18 Mon 4/2/18
84 Application Prototype 24 days Tue 1/30/18 Fri 3/2/18
85 Status Report 4 1 day Fri 3/2/18 Fri 3/2/18
86 Testing 22 days Sat 3/3/18 Tue 4/3/18
92 Status Report 5 1 day Mon 3/19/18 Mon 3/19/18
93 Test Readiness Presentation 1 day Thu 3/22/18 Thu 3/22/18
94 Status Report 6 1 day Fri 3/30/18 Fri 3/30/18
95 Status Report 7 1 day Fri 4/13/18 Fri 4/13/18
96 Design Day Poster 15 days Tue 4/3/18 Mon 4/23/18
97 Design Day 1 day Fri 4/27/18 Fri 4/27/18
98 Status Report 8 1 day Fri 4/27/18 Fri 4/27/18
99 Final Acceptance Test Procedure and Results 1 day Tue 5/1/18 Tue 5/1/18
100 Final Report 1 day Tue 5/1/18 Tue 5/1/18
101 Final Project Notebooks 1 day Thu 5/3/18 Thu 5/3/18
Ready
New Tasks : Manually Scheduled


Gantt Chart Tools
Schedule - Project Professional
Alyssa Thurston
File
Task
Resource
Report
Project
View
ACROBAT
Team
Format
Tell me what you want to do
Gantt Chart
Paste
Cut
Copy
Format Painter
Calibri
11
Mark on Track
Respect Links
Inactivate
Manually Schedule
Auto Schedule
Inspect
Move
Mode
Task
Summary
Milestone
Deliverable
Information
Notes
Details
Add to Timeline
Scroll to Task
Find
Clear
Fill
View
Clipboard
Font
Schedule
Tasks
Insert
Properties
Editing
GANTT CHART
Task Mode
Task Name
Duration
Start
Finish
February 2018
March 2018
April 2018
52 Project Plan Updates 9 days Tue 1/16/18 Fri 1/26/18
53 Detailed Design Updates 9 days Tue 1/16/18 Fri 1/26/18
54 Status Report 1 1 day Fri 1/19/18 Fri 1/19/18
55 Status Report 2 1 day Fri 2/2/18 Fri 2/2/18
56 Test Plan and Procedure 20 days Tue 1/23/18 Mon 2/19/18
65 Status Report 3 1 day Fri 2/16/18 Fri 2/16/18
66 Maintenance Manual 35 days Tue 2/13/18 Mon 4/2/18
67 Lecture 1 day Tue 2/13/18 Tue 2/13/18
68 Maintenance Manual Plan 1 day Thu 2/15/18 Thu 2/15/18
69 Maintenance Manual Draft 15 days Thu 2/15/18 Wed 3/7/18
70 Document Compilation 3 days Sat 3/3/18 Tue 3/6/18
71 Document Editing 1 day Wed 3/7/18 Wed 3/7/18
72 Maintenance Manual Final Draft 3 days Wed 3/7/18 Fri 3/9/18
73 Necessary Revisions 3 days Wed 3/7/18 Fri 3/9/18
74 Maintenance Manual Report 1 day Mon 3/12/18 Mon 3/12/18
75 User Manual 33 days Thu 2/15/18 Mon 4/2/18
76 Lecture 1 day Thu 2/15/18 Thu 2/15/18
77 User Manual Plan 1 day Fri 2/16/18 Fri 2/16/18
78 User Manual Draft 29 days Sat 2/17/18 Wed 3/28/18
79 Document Compilation 3 days Sat 3/24/18 Tue 3/27/18
80 Document Editing 1 day Wed 3/28/18 Wed 3/28/18
81 User Manual Final Draft 3 days Wed 3/28/18 Fri 3/30/18
82 Necessary Revisions 3 days Wed 3/28/18 Fri 3/30/18
83 User Manual Report 1 day Mon 4/2/18 Mon 4/2/18
84 Application Prototype 24 days Tue 1/30/18 Fri 3/2/18
85 Status Report 4 1 day Fri 3/2/18 Fri 3/2/18
86 Testing 22 days Sat 3/3/18 Tue 4/3/18
92 Status Report 5 1 day Mon 3/19/18 Mon 3/19/18
93 Test Readiness Presentation 1 day Thu 3/22/18 Thu 3/22/18
94 Status Report 6 1 day Fri 3/30/18 Fri 3/30/18
95 Status Report 7 1 day Fri 4/13/18 Fri 4/13/18
96 Design Day Poster 15 days Tue 4/3/18 Mon 4/23/18
97 Design Day 1 day Fri 4/27/18 Fri 4/27/18
98 Status Report 8 1 day Fri 4/27/18 Fri 4/27/18
99 Final Acceptance Test Procedure and Results 1 day Tue 5/1/18 Tue 5/1/18
100 Final Report 1 day Tue 5/1/18 Tue 5/1/18
101 Final Project Notebooks 1 day Thu 5/3/18 Thu 5/3/18
Ready
New Tasks : Manually Scheduled

Bi-Weekly Status Report 8

CSCE 4925

Date: 27 April 2018
Team Name: Fantastic Four
Prepared by: Alyssa Thurston

Accomplishments for the previous 2 weeks

- Completed design work on the front end
- Completed work on the php file to query data from the database
- Completed work on the poster and powerpoint for design day

Problems Encountered

- Coding an API for a node is proving to be difficult
- Google Charts isn't working out well…. At all. We can hard code the data, but when we try to import data from our database, it doesn't pull through. We've spent about 20 hours on this issue, and are currently researching alternatives.
- Filling up the database using the webscraper
- (solved) Bootstrap wasn't working properly.

Plans for the upcoming 2-week period

- Work on data integration
- Work on editing documentation
- Prepare for product delivery

Project Plan Reconciliation

- Gantt chart included on the following pages
- We are very behind schedule

Group Attendance

Meetings (Dates)	**Alyssa Thurston**	**Travis Goral**	**Breuna Riggins**	**James Sabetti**
April 19th- Lecture	Present	Present	Present	Present
April 25th- Hangouts call	Present	Present	Present	Present
April 26th- Team meeting	Present	Present	Present	Present
April 26th- Hangouts call	Present	Present	Present	Present
April 27th- Design Date	Present	Present	Present	Present

Project Plan Updates	9 days	Tue 1/16/18	Fri 1/26/18
Detailed Design Updates	9 days	Tue 1/16/18	Fri 1/26/18
Status Report 1	1 day	Fri 1/19/18	Fri 1/19/18
Status Report 2	1 day	Fri 2/2/18	Fri 2/2/18
Test Plan and Procedure	**20 days**	**Tue 1/23/18**	**Mon 2/19/18**
Status Report 3	1 day	Fri 2/16/18	Fri 2/16/18
Maintenance Manual	**35 days**	**Tue 2/13/18**	**Mon 4/2/18**
Lecture	1 day	Tue 2/13/18	Tue 2/13/18
Maintenance Manual Plan	1 day	Thu 2/15/18	Thu 2/15/18
Maintenance Manual Draft	**15 days**	**Thu 2/15/18**	**Wed 3/7/18**
Document Compilation	3 days	Sat 3/3/18	Tue 3/6/18
Document Editing	1 day	Wed 3/7/18	Wed 3/7/18
Maintenance Manual Final Draft	**3 days**	**Wed 3/7/18**	**Fri 3/9/18**
Necessary Revisions	3 days	Wed 3/7/18	Fri 3/9/18
Maintenance Manual Report	1 day	Mon 3/12/18	Mon 3/12/18
User Manual	**33 days**	**Thu 2/15/18**	**Mon 4/2/18**
Lecture	1 day	Thu 2/15/18	Thu 2/15/18
User Manual Plan	1 day	Fri 2/16/18	Fri 2/16/18
User Manual Draft	**29 days**	**Sat 2/17/18**	**Wed 3/28/18**
Document Compilation	3 days	Sat 3/24/18	Tue 3/27/18
Document Editing	1 day	Wed 3/28/18	Wed 3/28/18
User Manual Final Draft	**3 days**	**Wed 3/28/18**	**Fri 3/30/18**
Necessary Revisions	3 days	Wed 3/28/18	Fri 3/30/18
User Manual Report	1 day	Mon 4/2/18	Mon 4/2/18
Application Prototype	24 days	Tue 1/30/18	Fri 3/2/18
Status Report 4	1 day	Fri 3/2/18	Fri 3/2/18
Testing	**22 days**	**Sat 3/3/18**	**Tue 4/3/18**
Status Report 5	1 day	Mon 3/19/18	Mon 3/19/18
Test Readiness Presentation	1 day	Thu 3/22/18	Thu 3/22/18
Status Report 6	1 day	Fri 3/30/18	Fri 3/30/18
Status Report 7	1 day	Fri 4/13/18	Fri 4/13/18
Design Day Poster	15 days	Tue 4/3/18	Mon 4/23/18
Design Day	1 day	Fri 4/27/18	Fri 4/27/18
Status Report 8	1 day	Fri 4/27/18	Fri 4/27/18
Final Acceptance Test Procedure and Results	1 day	Tue 5/1/18	Tue 5/1/18
Final Report	1 day	Tue 5/1/18	Tue 5/1/18
Final Project Notebooks	1 day	Thu 5/3/18	Thu 5/3/18