Group 26 - Qualtic

Anni Campbell - 0029078839
Deep Mehta - 0029703648
Edison Proaño - 0030044402
Karthik Sharma - 0028851090
Sarah Whitcomb - 0029571886
Shivane Jetley - 0029669215
Yuhang Zhang - 0029939131

We certify that the submitted work does not violate any academic misconduct rules, and that it is solely our own work. By listing our names and student IDs we acknowledge that any misconduct will result in appropriate consequences. Moreover, we have **read and understood the assignment instructions**.

"As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together - we are Purdue."

Group Login Information

iTap
Username: g1109698
Password: 5U6sWPwX

SQL Account
Username: g1109698
Password: SEVERELY26

Web URLs

Company Website Home Page: https://web.ics.purdue.edu/ g1109698/Map/home
Company Website About Page: https://web.ics.purdue.edu/ g1109698/Map/about
Company Website Contact Page: https://web.ics.purdue.edu/ g1109698/Map/contact
Company Website Products Page: https://web.ics.purdue.edu/ g1109698/Map/products
Company Website Create a New System Page: https://web.ics.purdue.edu/ g1109698/Map/system

Company Twitter Page: https://twitter.com/Qualtic_
.csv Youtube Tutorial: https://youtu.be/S9ifOO9Ggp4

Bonus Point Considerations

1. Phase 1 and Final Report formatted in Latex
2. Company Social Media Accounts (Instagram, LinkedIn, Twitter)
3. Google Maps Interface and API
4. Implementing visuals to alert users of AQI ratings (smiley faces in AQI ratings)
5. Utilizing JavaScript to create graphs for the analytics dashboard
6. Interactive graphs with real-time information about the parameter
7. Prior to our system design phase, we discussed our company's values, vision, and philosophies. We then aligned our project to fit these.
8. Self-created noise function
9. Link to team-created Youtube video with instructions on how to create .csv file for website

Contents

Executive Summary

Measuring air quality to ensure suitable living conditions is paramount in ensuring a healthy quality of life for people all over the world. Research has linked high levels of air pollution and particulate matter (PM) to lung diseases, heart diseases, and other related heart problems (Environmental Protection Agency). Furthermore, ambient air pollution accounts for 24% of all deaths from stroke, 25% of all deaths and diseases from ischaemic heart disease, and 29% of all deaths and diseases from lung cancer (World Health Organization). Due to the harmful impacts that poor air quality can have on the body, Qualtic has embraced the mission to provide optimal air sensor network solutions that can be implemented all over the world. This implementation of an advanced air quality monitoring network becomes the first step in improving air quality for people from all walks of life.

Why Qualtic? We consider the user in everything we do. Our fully-functioning, visually appealing, and easy-to-use web-interface considers all of the users needs and constraints. Our designs are flexible; every system is customized for the region it is designed for. In order to determine a solution that best fulfills each client's needs, we consider their unique budget, location preferences, and zoning information. We aim to maximize knowledge of areas with high PM values, and therefore optimizing the functionality of the implemented air sensing network. This is accomplished through use of machine learning to teach optimal mobile sensor movement and an optimization algorithm which optimizes placement of the air quality sensors.

Qualtic has created a product that allows the public to easily access the air quality information for their area through interaction with the sensor placement network on the website. Users are able to look at current historical trends, thus enabling all users to make more informed decisions about their day-to-day activities. After implementation, the Qualtic air sensing system can be a benefit to all peoples; families, neighbors, and even city officials and governments looking to improve air quality in their area through accessing the current and historical trends available through the Qualtic web interface. Government officials can use the air sensor network to identify problematic air quality areas in their community, and sensitive health groups will be informed on when to limit their time outside.

Company Description

Here at Qualtic, our mission is to provide the best way to monitor air quality to fit the diverse needs of each individual. We strive to implement cutting edge technologies to design the best dynamic air sensor network solutions around the world. We abide by our philosophy of providing our users with the knowledge of locations with the highest Air Quality Index (AQI). We provide quality products and services that align with our core company values:

- *Integrity* - We pride ourselves on performing honest work.

- *Commitment to Diversity* - Our team is very diverse. In this same way, we are determined to provide a diverse set of solutions that fit each individual's unique needs.

- *Passion* - At Qualtic, we are passionate about what we do. We strive to provide our users with high quality solutions.

Air quality is a huge problem that our world faces today. For this reason, we provide functional and user-friendly solutions that can be used by anyone, anywhere. Our products can be used by anyone with access to the internet, and can be widely implemented by city and county governments. Our goal is to provide you, the user, with the most optimal dynamic air sensor network to fit your diverse needs.

As technologies grow and change, our vision remains the same: *Better air, everywhere.*

Products & Services

Dynamic Sensor System

Qualtic works hard to provide the client with the best sensor network to identify the areas in each city with poor air quality. Our Mobile and Fixed Sensors collect Air Quality, Temperature, Humidity, and Air Pressure data, which are easily accessible through our user-friendly website. This website allows anyone to access real-time weather information in the area and highlights areas with poor air quality so the appropriate measures can be taken to improve the overall quality of life.

Qualtic's website prompts the user to enter a range of budgets. Using information provided by the user about their city, we then formulate a few recommended system solutions. It is up to the user to then choose their preferred solutions. Once a system has been chosen, we handle the implementation of the sensors, ensuring full-functionality and ready-to-use availability. Our website provides a user-friendly interface for any user to access live-weather data, as well as view forecasts of data in the area. We highlight areas with high pollution levels for citizens to avoid. These highlighted regions also act as visualization of problem areas. Local governments can direct their implementation of creating a cleaner city by targeting their efforts at these respective areas. Qualtic hopes to implement this product in cities across the globe to bring attention to the growing pollution levels and by doing this, aim to truly have *"Better air, everywhere."*

Web Interface

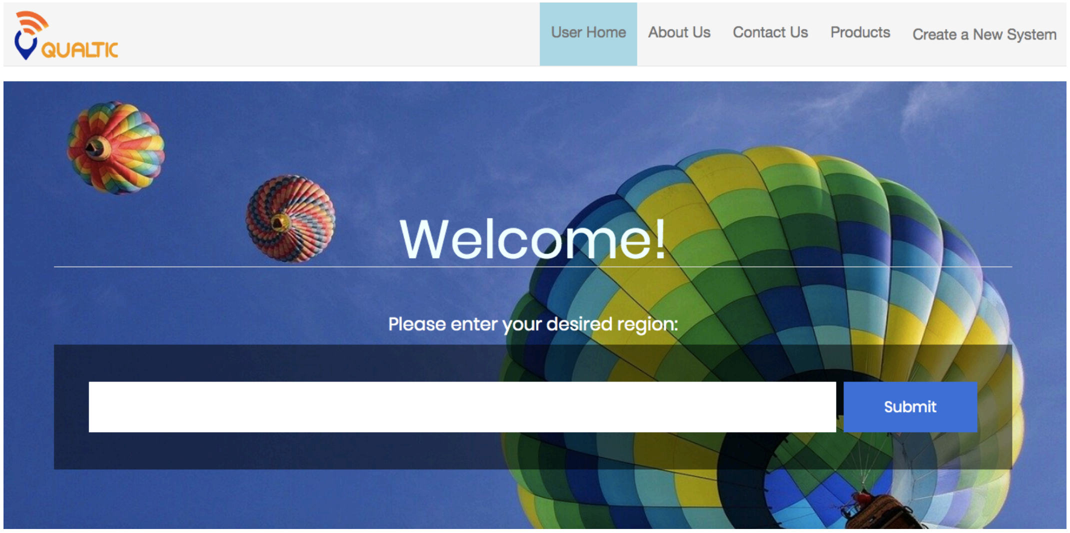


Figure 1: Welcome Page

The website begins with a simple, clean, and user-friendly homepage. We ask our users to pick a region where the system has been created already. They can do this by selecting their region from the dropdown menu, or by typing the region manually into the textbox.

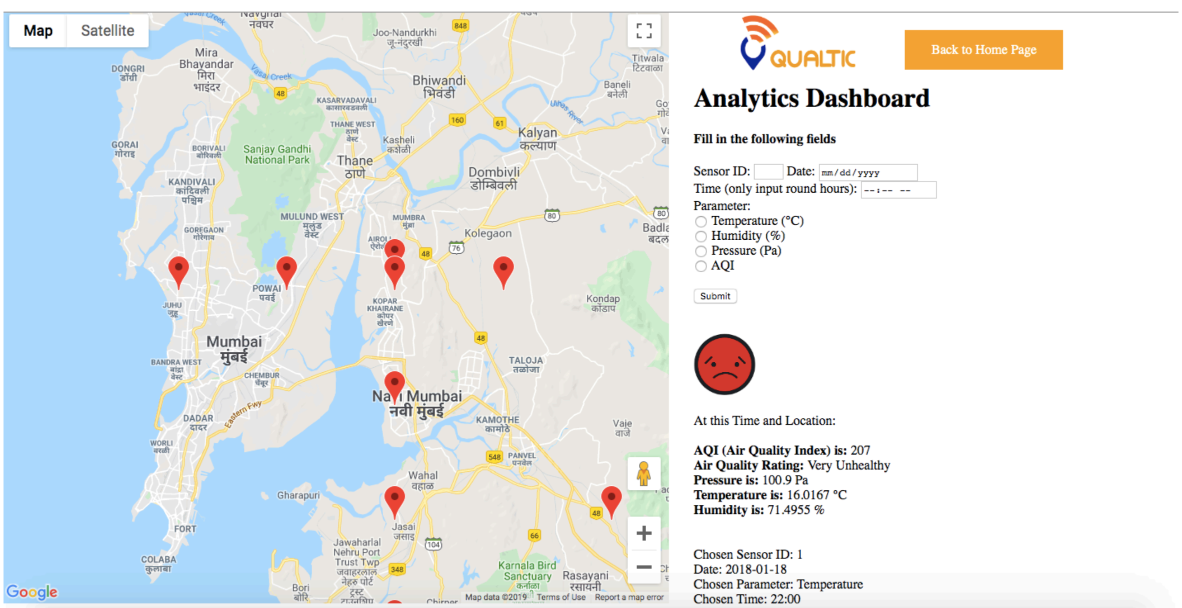
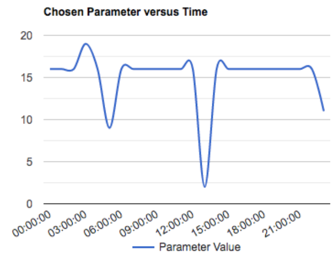


Figure 2: Map and AQI Readings

After submitting the region, the user will be taken to a new page with a map of the region scattered with viable sensors. All we ask of the user is for them to submit a sensor ID, desired date, desired time, and parameter of temperature, humidity, pressure, or PM (particulate matter) 2.5. Upon clicking submit, a pop-up next to the map will display the Air Quality Index (AQI), air quality rating, pressure, temperature, and humidity for their selected sensor and parameters.



Air Quality Index - Particulate Matter	
301 – 500	Hazardous
201 – 300	Very Unhealthy
151 – 200	Unhealthy
101 – 150	Unhealthy for Sensitive Groups
51 – 100	Moderate
0 – 50	Good

Figure 3: Map and AQI Readings Continued

After viewing the AQI and Air Quality Rating, the user can scroll down to see a graph of their chosen parameter vs. time. From this graph, the user can see historical trends from the past, and are able to use the graph as a means of predicting future possible readings. Below the graph is a legend for determining the Air Quality Index based on particulate matter (PM) values. The colors on the legend will correspond with the colored "smiley face" icon. This playful and user-friendly feature makes air quality readings understandable to any user.

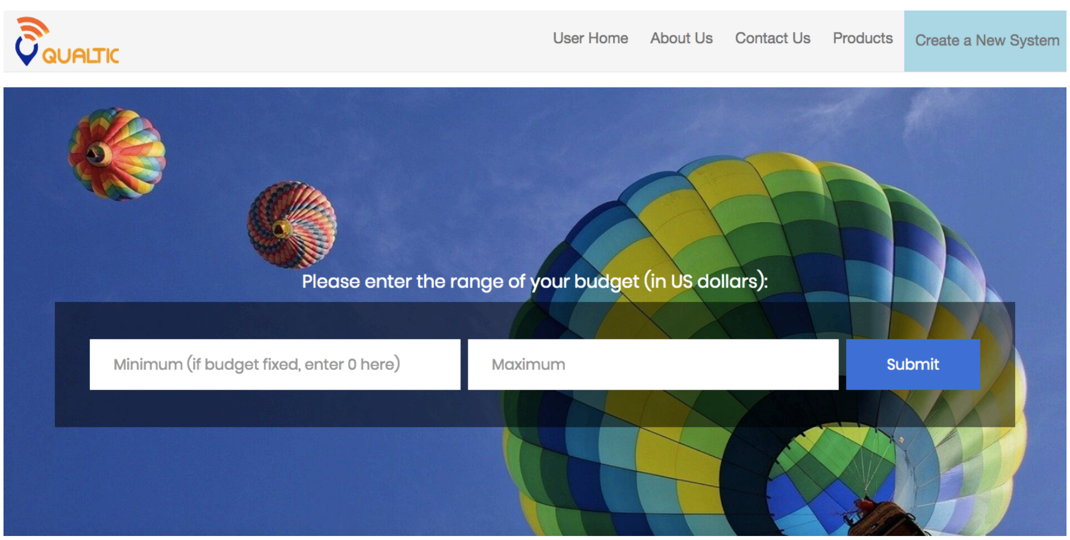


Figure 4: Budget Input Page

Upon clicking the "Create a New System" tab in the top right-hand corner, the user will be taken to a new page that asks them to entire their minimum and maximum budget. It is important to us that we keep the user's ideal budget in mind while creating the optimal solution.

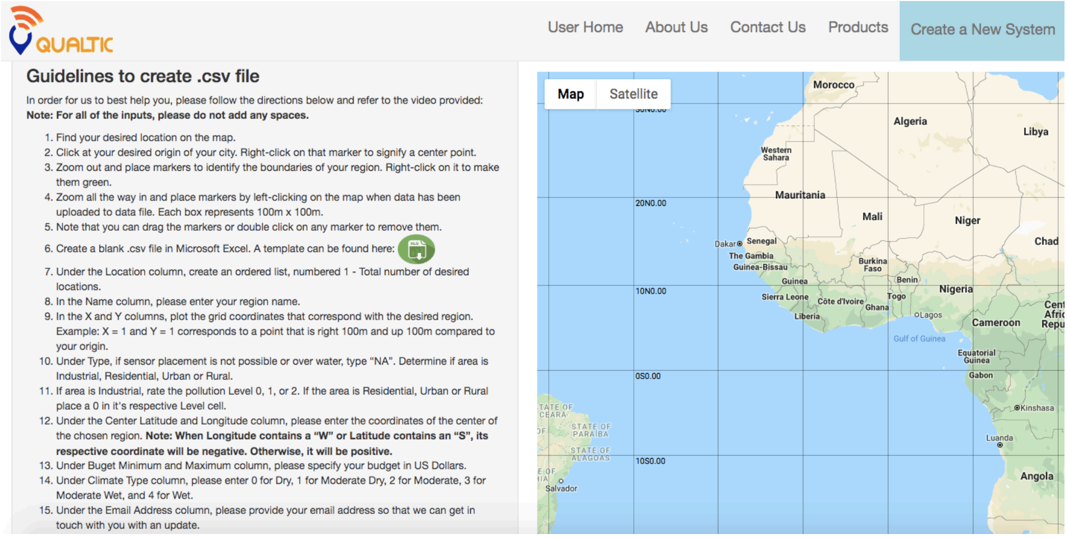


Figure 5: Create .csv Instructions

Upon clicking the "Create New System" tab, the potential client is provided step-by-step instructions on how to upload a .csv file containing data for their selected region. Alongside step-by-step instructions, Qualtic has provided a link on the website to a visual tutorial on how to perform these guidelines. Within a week, we will email the client a set of potential solutions with various budget requirements, sensor placements, and quantity of fixed and mobile sensors.

Company Future

Future Initiatives

- **Live maps:** We plan to implement real-time readings as part of the map so the user is easily able to access live-weather data.
- **Automated algorithm**: We plan to optimize the response time by creating a fully-automated and integrated algorithm. By doing this, we eliminate the wait time and plan to bring potential solutions to the client within moments of them clicking the submit button.
- **More accurate predicted data**: By integrating machine learning into the simulation algorithm to learn from collected data, Qualtic hopes to improve the accuracy of the projected data. The collected data will be stored and used to create more accurate regression equations.
- **Client login**: By creating a separate portal for the client to login to, Qualtic will be able to store personalized data. This client portal will display data from the .csv file and will allow the clients to make edits to any of that information to account for changes in their city.

Conclusion

Qualtic is a company made up of a diverse, open-minded team that strives to put the user first. With a mission statement that aims to provide the best way to monitor air quality to fit the diverse needs of each user, Qualtic aims to call attention to the growing air pollution epidemic that is causing premature deaths and health issues around the world. The vision for this product is to highlight the areas with dangerous levels of air pollution in order to improve efforts of improving air quality. Through the user-friendly web interface, Qualtic is able to provide historical data, weather trends, and interactive sensor maps. We strive to be at the forefront of new cutting-edge technologies to improve air quality around the world. Qualtic looks forward to working with you. *Better air, everywhere*; not just a promise, but a guarantee.

References

- (n.d.). Retrieved from https://www.w3schools.com/php/php_file_upload.asp.
- (n.d.). Retrieved from https://developers.google.com/maps/documentation.
- (n.d.). Retrieved from https://developers.google.com/maps/documentation/javascript/mysql-to-mapsloading-the-xml-file.
- (n.d.). Retrieved from http://www-das.uwyo.edu/ geerts/cwx/notes/chap16/geo_clim.html.
- Ambient air pollution: Health impacts. (2018, September 25). Retrieved from https://www.who.int/airpollution/ambient/health-impacts/en.
- Form Handling, H. T. M. L. F. advanced. (n.d.). How to switch the 'action' field in an HTML form dynamically. Retrieved from http://javascript-coder.com/html-form/html-form-action.phtml.
- Haponiuk, B. (2018, October 24). Air Pressure at Altitude Calculator. Retrieved from https://www.omnicalculator.com /physics/air-pressure-at-altitudehow-to-calculate-air-pressure-at-altitude.
- Hylton, M. (2016, July 5). The AQI Equation. Retrieved from https://forum.airnowtech.org/t/the-aqi-equation/169.
- In this document we have discussed JavaScript Form Validation using a sample registration form. (n.d.). Retrieved from https://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php.
- Johnson, J. J. J. (1965, October 1). Set date input field's max date to today. Retrieved from https://stackoverflow.com/questions/32378590/set-date-input-fields-max-date-to-today.
- JSFiddle. (n.d.). Retrieved from http://jsfiddle.net/VC88F/.
- k388mmoo4v. (n.d.). Retrieved from https://codesandbox.io/s/k388mmoo4v.
- Live Animated Air Quality Map (AQI, PM2.5...). (n.d.). Retrieved from https://www.airvisual.com/air-quality-map?lat=11.89735lng=0.852629zoomLevel=2.
- PHP Connect to MySQL. (n.d.). Retrieved from https://www.w3schools.com/php/php_mysql_connect.asp.
- Research on Health and Environmental Effects of Air Quality. (2019, May 15). Retrieved from https://www.epa.gov /air-research/research-health-and-environmental-effects-air-quality.
- Rokr. (2019, November 30). 99 Free HTML Website Templates 2019. Retrieved from https://colorlib.com/wp/free-html-website-templates/free-html-templates.
- Shapes : Maps JavaScript API : Google Developers. (n.d.). Retrieved from https://developers-dot-devsite-v2-prod.appspot.com/maps/documentation/javascript/shapesuser-editable.
- Sohcot, S. (2019, January 20). Simple PHP Contact Form Tutorial (Part 1 of 2). Retrieved from https://medium.com /@stevesohcot/simple-php-contact-form-tutorial-part-1-of-2-6cdb2e383b23.
- WunderMap®: Interactive Weather Map and Radar. (n.d.). Retrieved from https://www.wunderground.com/wundermap?lat=41.79lon=-87.74.

Appendices

Appendices Preview

- Overall Data Flow
- Database Design
- Simulation Model
- Drone Movement
- Sensor Placement Optimization

Overall Data Flow

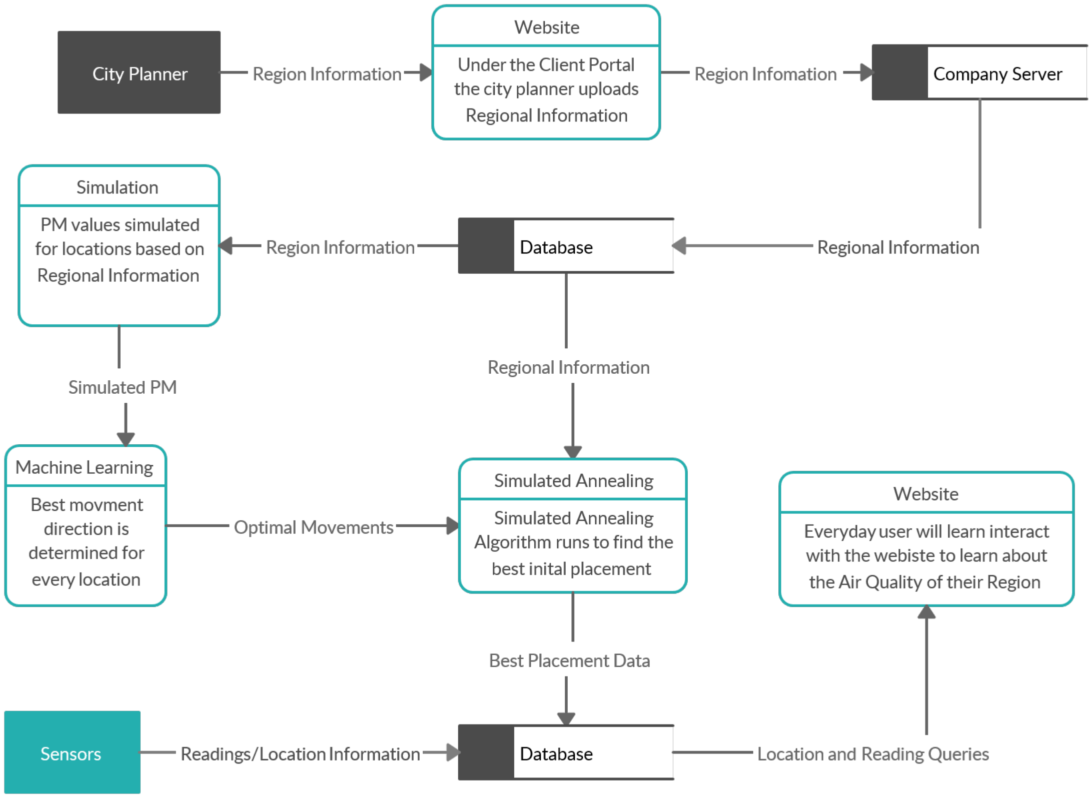


Figure 6: Data Flow

Database Design

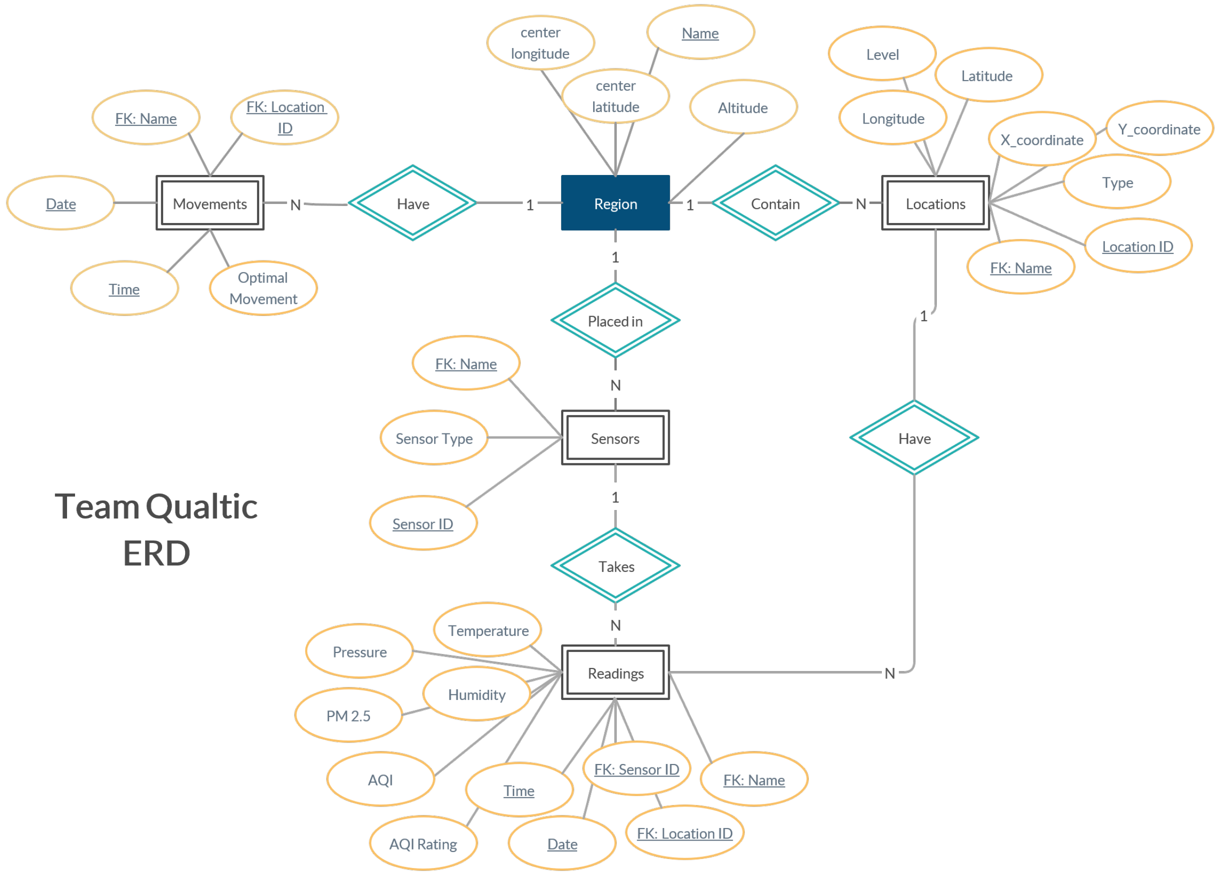


Figure 7: Database Design ERD

Simulation Model

We used the given Krakow temperature, humidity, pressure and AQI to track trends for those type of values. We expect different region types to follow similar trends, but have different coefficient values. In order to find graphical trends, we used R to plot the data as a function of time. In order to test our regression equations, we create them using 75% of the data (randomly selected), and then tested our regression on the remaining 25%. Data from the .csv file is read into forecast well for distinct regions.

Temperature
We expect region type (Industrial, Residential, Rural, and Urban) to impact the PM2.5 values, so we split up the data in order to identify region trends and coefficient values. We clean daily data in order to account for sensor failure, then average hourly data to get daily averages. Once we determined the regression equation for each region type, we were able to forecast data for our given region. The regression equations follow this trend: y = $\beta_0 + \beta_1 \frac{1}{x} + \beta_2 x + \beta_3 x^2$

Derived regression equations for region type:

Industrial:
PM 2.5 level $= 32.93 + \frac{214.6}{date} - (0.1085)(date) - (0.0009763)date^2$

Rural:
PM 2.5 level $= 37.39 + \frac{189.2}{date} - (0.105)(date) - (0.001408)date^2$

Urban:
PM 2.5 level $= 53.74 + \frac{149.6}{date} - (0.5689)(date) - (0.001777)date^2$

Residential:
PM 2.5 level $= 36.92 + \frac{202.3}{date} - (0.1071)(date) - (0.001281)date^2$

We then read in the .csv file submitted by the client. Based on the region type, the respective regression equation is used to forecast data. If the region type is Industrial, there will be a ranking of how polluted that region is. This rank dictates how much to scale the data (0 – no change, 1 – add 10%, 2 – add 20%). Based on the country/continent of the city and average PM2.5 data, we have set ranges that the respective region has historically fluctuated between. By doing this we are able to closely forecast what the PM2.5 value will be for a given time of year. This value is converted into an AQI value, so it is easily interpreted by the user.

Temperature, Humidity, and Pressure all follow this basic trend with varying β values:
$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta x^3 + \beta x^4$

Temperature
Using this regression equations, we were able to get predicted data for the given region. After research we found that weather follows a pattern with respect to latitude. This trend is followed by this equation
Temperature $= \frac{-1}{130.645}(lat)^2 + N$

N being the amount the average temperature is shifted by given time of year and location on the map. This N is vector is calculated using given Krakow temperature data. We are able to predict the Temperature values for any given region as long as we have an input of latitude. We first input the Krakow temperature and latitude in order to calculate Degree Celsius/latitude degree. Using this N vector and the Latitude value pulled from the .csv file, we are able to forecast temperature data.

Humidity
After graphing the given humidity data, we noticed that there was no considerable trend in any of the data. Our plots show that humidity tends to be incredibly variant and it's yearly trend relies heavily on type of climate. We ask the user to input their climate type as a number (0 – Dry, 1 – Moderate Dry, 2 – Moderate, 3 - Moderate Wet, 4 – Wet). These inputs correspond to if-statements with set ranges for humidity. A value is randomly chosen between this range and noise is added to show variability.

Climate Type	Value Entered	Humidity Range
Dry	0	0% - 30%
Moderate Dry	1	10% - 30%
Moderate	2	30% - 70%
Moderate Wet	3	70% - 100%
Wet	4	60% - 100%

Air Pressure

In order to calculate accurate forecasts, we decided to use the forecasted temperature rather than air pressure regression trends and calculated air pressure accordingly.
$P = P_0 * e^{\frac{-g*M*h}{R*T}}$
$g = 9.80665\frac{m}{s^2}$ $M = .0289644\frac{kg}{mol}$ $R = 8.31432\frac{N*m}{mol*K}$
All ANOVA tables had a p values less than 5% (meaning predictors are statically significant) and have R^2 values between 38.23% and 70.79%.

In order to account for variability in weather data, we created our own function to add "noise". This function adds a random value between 0 and 1σ to 10% of the data. In the winter to account for even larger variability the upper bound is 2σ.

Top Challenges

- Challenge 1: Unable to predict daily data accurately
  There were difficulties in trying to figure out a balance between daily and yearly regression for collected weather data. Currently the yearly regression runs accurately, but daily data is not accurately predicted. Yearly trend is followed accurately but this was achieved at the sacrifice of accurate daily predictions.

  Solution: We randomly simulated points in order to get usable data.

- Challenge 2: No trend found for humidity data
  Given our plotted Krakow Humidity values, we noticed no particular trend in the data. Humidity is dependent on various factors which we were not able to account for.

  Solution: Using a Climate Type input from the client and researched average Humidity ranges we randomly pull a mean and add noise to the data in order simulate variability

R Code

- Challenge 1: STACK.R, Line 16
- Challenge 2: humidity.R, Line 57

Drone Movement

Description/Justification

The drone movement algorithm is written in the main.R file (using functions from the Reinforcement Learning Package) (Lines 30-47) and uses the machine_learning.R file (using functions from the stringr package). Reinforcement learning based machine learning techniques are used to determine the optimal movement pattern for the drone.

The process of creating the reinforcement learning begins by defining possible actions and states in main.R. Actions is a vector denoting the types of movement that the drone can make. The actions are hard coded and standard across all systems. States will vary by region and is determined by the variable n (which is the side length of the nxn grid that the user has defined at their desired sensing region). Next, the reinforcementLearning function requires a user defined environment, containing rewards, as an input. The environment function is defined as gridbyn in machine_learning.R. Taking inputs state and action, gridbyn enforces outside boundaries using the global variable, n, to change the parameters of the if statements creating the environment. Rewards for each state-transition combination are assigned at the end of the gridbyn function. The reward value assigned is the PM value at each space taken from the simulated data. For spaces where a sensor can not go there is a NA value for the simulated value and the reward assigned for the NA is -150.

The reinforcementLearning function, outputs a model which contains the policy of the best action to take for any spot in the nxn environment. The policy is a matrix, each row denotes a location, and column an action. The value for each location action combination tells how desirable that action is at that location. The rows are sorted numerically, columns sorted alphabetically, and stored in a 3-dimensional array called rec_policy.

This process was decided to be the best because it simply and effectively trained the robot brain to look for the locations with highest PM values, which aligns with the company philosophy for how to best monitor air quality.

Top Challenges

- Challenge 1: Formatting the output of the Machine Learning
  The issue was that the output of the model from the reinforcementLearning function was a randomly ordered list with no apparent logic to the ordering of the returned table. The row names were a character type; simply using sort was not an option. Then there was the problem of putting these values which were dependant state, action, and hour in a usable data structure.

  Solution: convert the row names to integers and then employ the sort function, and the column names were alphabetized using the sort function. The data structure problem was solved using a 3-dimensional array.

- Challenge 2: Determining how much to let the machine learning itself
  This issue was that within the environment function. The initial attempt included a way of restricting the movement of the drone so it could not enter NA spaces, and spaces that already had sensors at that instance in time.

  Solution: We implemented a very large negative reward if it entered NA spaces so the drone would learn to never move there, and realized the Machine Learning should learn not to end up in a space with another sensor anyway because it would only return one value.

R Code

- Challenge 1: main.R, Lines 49-53
- Challenge 2: ”movement.R”, Lines 35-44

Sensor Placement Optimization

The optimization algorithm uses a method of simulated annealing to determine the optimal initial placement and combination of fixed and mobile sensors. The simulated annealing uses files Simulated Annealing.R, create_initial.R neighbor.R, and obective_val.R. Inputs into the simulated annealing function, SA, are budget, num_hours, user_data, simulated_data, rec_policy, temperature, maxit, and just_values.

To start, the create_initial function outputs an initial solution of randomly placed fixed sensors that is scored using the score_layout function (found in objective_val.R).

Objective function : max $\sum pmij$,where the pm value read by a sensor at location i at date and time j

In causing the objective function to maximize the PM values that we are reading in our final layout, the algorithm should determine the best way to do so is to place sensors in regions that have high simulated PM values, and thus aligns with the Qualtic philosophy of monitoring air quality by finding the locations with the highest AQI values. Furthermore, this objective function accomplishes this goal in a concise manner with few lines of code. This conciseness reduces the runtime of our algorithm.

One of the major strengths of the algorithm is the simplified objective function. Since the objective function is a double summation over the measured PM values, the goal is accomplished in a concise manner with few lines of code, thus reducing the runtime of our algorithm. Another major strength is the algorithm's natural tendency to find areas with poor air quality, since the nature of the simulated annealing algorithm is that it will only keep later iterations of the initial placement if the air quality is poorer. This is indicated by a larger neighbor objective value.

One of the weaknesses of the algorithm is in regards to determining the failure rate of sensors. The current algorithm makes the sensor network using 90% of the budget input in an attempt to account for the need to potentially replace sensors. More research could have been done into the lifetime of fixed and mobile sensors in order to develop an effective stochastic model for estimating when sensors may need to be replaced. Another weakness is in regards to code efficiency. Some functions in the algorithm could have been written more efficiently, using fewer lines of code and leading to a decrease of the overall run time of the algorithm.

Pseudocode

Create an initial solution using budget, num_hours, and user data, that places as many fixed sensors as possible.

Score initial solution using the objective function store in variable

Store as current best solution

For (1 to maxit){
Create a neighbor solution
Score neighbor solution
Do comparison of score for neighbor solution and current best solution.
If the neighbor is better then assign it to best solution Else If the neighbor is worse cause choose to probabilistically accepts it as the new best
}

Return placement
End

Create neighbor function will either shift the locations of some subset of sensors or change the number of fixed and mobile sensors

Top Challenges

- Challenges: The major challenges can be found in lines 14 - 54 in feasibility_checks.R and 49 - 61 in neighbor.R. There were issues in correctly identifying user defined NA spots and subsequently removing them from the list of new locations that sensors could move to. This caused further issues when checking if a potential new location for a sensor was open. This is especially true for line 41 of the feasibility_checks file because if placement[i] equaled NA, the conditional would return a NA instead of a true or false, subsequently resulting in an error and the termination of the script.

  Solution: In order to guarantee that all of the NA spots were removed from the list of spots that a fixed sensor could be moved to, all of the NA spots were found in the space using the *which function*. Next, the intersect of that vector was saved along with the vector of locations where fixed sensors were trying to move to in order to make a new placement. These points were then removed to the neighbor function, and appropriately removed the NA locations from the vector of potential points for new sensor locations.

R Code

- Challenge 1: feasibility_checks.R, Lines 14-54
- Challenge 2: neighbor.R, 49-61