Proceedings of the Informing Science + Information Technology Education Conference

An Official Publication of the Informing Science Institute
InformingScience.org

InformingScience.org/Publications

June 30 – July 4, 2019, Jerusalem, Israel

USER-GENERATED GEOSPATIAL METEOROLOGY MAP PROTOTYPE

William Miranda-Hill	Computer Science Department, University of La Verne, La Verne, CA, USA	william.miranda-hill@laverne.edu
Jozef Goetz*	Computer Science Department, University of La Verne, La Verne, CA, USA	jgoetz@laverne.edu

* Corresponding author

ABSTRACT

Aim/Purpose	This project aims to prototype the functionality of a user-generated geospatial meteorology map. This includes the design and implementation of a database driven website with a public and a password protected admin component, in addition to database, web server and hardware components.
Background	Previous research described and assessed the feasibility of a system in which end-users generate environmental data and examined the quality of the data provided. We sought to distill the minimum essential use-cases to achieve the required functionality, based on preexisting and original theorization, and then implement them in a functional prototype.
Contribution	The possible value of this potential information system, both as a dataset for metrology, climatology, ecology, as well as other fields of research, and also as an end-user web service for highly accurate weather reports, has been noted by previous researchers. The specific contribution of this project is to, by the implementation of a functional prototype, establish that a smart device can remotely generate geopositioned weather reports, which can be accepted by a central server and displayed on a public world map.
Findings	Through the implementation of the project, we were able to assess the quality of the use-cases outlined. We found the project was a functional information system, with each public server-side and hardware competent interfacing correctly, most limitations resulting from the scope of the project.

Accepted by Executive Review by Editor: Eli Cohen | Received: March 9. 2019 | Revised: March 13, 2019 | Accepted: March 14, 2019.

Cite as: Hill, W. M., & Goetz, J. (2019). User-generated geospatial meteorology map prototype. *Proceedings of the Informing Science and Information Technology Education Conference*, *Jerusalem, Israel,* pp. 257-269. Santa Rosa, CA: Informing Science Institute. https://doi.org/10.28945/4257

Impact on Society	This would unlock the possibility of the next step towards the full realization of the theoretical information system: a limited real world rollout.
Future Research	As this project outlines and implements the minimum use-cases required to this system, future research would ideally involve the limited real-world rollout of the system and asses the quality of the data provided. Further research could also be conducted on data quality assurance strategies, both at the point end-user device sensors and broad-scale error correction.
Keywords	GIS, weather, crowdsourcing, Apache, MySQL, PHP, Arduino, CSS, JavaScript

INTRODUCTION

The most fruitful avenue to improving human life through technological innovation seems, as of the last few years, not to be found in radical developments in technology, but in the novel exploitation of established technology adoption. Currently, environmental sensing is mainly accomplished via few highly accurate weather stations, only achieving report precision on the level of counties. It seems, with only a minor evolution of current technology, wide smart device adoption could be leveraged to supplement existing meteorological information systems with great benefit to the granularity of data. This would be of particular benefit in densely populated areas such as major cities (Kanhere 2011).

Some researchers have theorized a future system in which mobile phones and other smart devices are equipped with small integrated environmental sensor units (such as the 2.5mm BME280 sensor used in this project) which could sense the environment around them and pair this data with a GPS position and a timestamp, send this data to a central service, which would in turn be able to display these reports to end users, positioned on a map. This would be a function similar to the mobile app *Waze* (https://www.waze.com), but for weather instead of traffic. While each unit would achieve fidelity to reality inferior to that of a weather station, by its increased precision this future system could obtain greatly improved report resolution (perhaps on the level of city blocks rather than counties). If the system achieved wide adoption, it's data could also go to supplement weather prediction based on traditional weather stations, and in turn display these predictions on the same service. Previous research has acknowledged as of yet unproven potential of such a system to provide, as described by Muller et al. (2015), a "valuable source of high temporal and spatial resolution, real-time data" and thereby add "value to science, technology and society."

BACKGROUND

Previous research has proposed the possibility of the system this paper describes, but has not attempted to develop a prototype. We have found that while many similar designs have been designed and implemented, it is consensus that this specific potential new system has broad potential, but is unproven. Many existing web services display weather information on a map, either only from major weather stations, or including amateur weather stations. However, in both cases these weather stations are stationary and few, thus they do not meet the criteria of this concept. These amateur weather stations however, in the view of Chapman Bell, and Bell (2017) represent a step towards a crowdsourced meteorological paradigm.

Muller et al. (2015) offers a comprehensive review of the specific concept, and future potential, and can thus serve as a useful guideline for the requirements of a potential prototype of such a system. Beyond a review of crowdsourcing projects and the categorization thereof, they approached the topic from the perspective of a potential source of data and examined it's "value within the climate and atmospheric science disciplines," and its connection to citizen science. This is somewhat aside from our focus, which while generating a dataset which can be used in this way, mainly focused on its potential role as an end-user service. While this involves a form of what Cuff, Hansen, and Kang (2008) categorized as 'passive' crowdsourcing where devices "silently collect, exchange and process information," it nonetheless requires motivating end user initiation, thus its utility to end users must be

established to achieve adoption. The data science aspect is thus contingent on the end user competent strategy. Muller et al. (2015) talk extensively about citizen science. For decades, weather data has been generated by sources other than official weather stations, by citizen scientists with amateur weather stations. Being static, few, and not a fully integrated system for end users, this does not fully meet the specifications of the system they are concerned with. However, they did also review several projects with an end user strategy in their overview of projects and techniques related to crowdsourced weather reports, such as the user-generated environmental data sources: UKSnowMap, PressureNet, and WeatherSignal.

Projections of the future involving the minor evolution of smart devices, such as the inclusion of small integrated environmental sensor units, are clearly limited by the realities of commercial smart device production. Overeem et al. (2013) however, showed that novel techniques such as utilizing smartphone battery thermometers to generate temperature data might be a viable approach to crowdsourcing air temperature. They also emphasized along with Muller et al. (2015) the additional benefits that would result from this systems deployment to specifically densely populated urban areas, as was explored by Kanhere (2011). While this smartphone battery approach may not be ideal, it does demonstrate the possibility of limited deployment without any prerequisite evolution in consumer smart device features. When a "straightforward heat transfer model is employed to estimate daily mean air temperatures" they write that they can obtain "relatively accurate air temperature information for the urban canopy layer" and report that their "results demonstrate the enormous potential of this crowdsourcing application for real-time temperature monitoring in densely populated areas."

Purpose of the study

The information system thus far described differs from conventional means of collecting metrological data, thus its functionality needs to be prototyped. A prototyping phase such as this could assist in theorizing about the feasibility of the system and convincing others that it is worth consideration. It also allows us to qualitatively analyze and test changes to the system on a smaller scale. However, both to implement the prototype and to assess its merit after the fact, first requires the outlining of certain requirements, from which design specifications can be extrapolated.

We established a list of minimum use-case scenarios necessary for a functional information system which meets the criteria. The primary objective of this study is to develop a framework for collecting metrological data. To accomplish this purpose the following conditions should be satisfied:

- Design and implement a database driven website with a public and a password protected admin component on the web server side
- Design and implement a MySQL database which handles both reports and users
- Build a hardware weather station capable of communicating with a web server
- Design and implement a web server of accepting weather reports submitted manually through a public website or via direct remote SQL connection
- Display weather reports in their location on an interactive world map.

The purpose of this project therefore is the practical demonstration of a system that meets these standards and its status as functional, such that this work may be used in future evolutions of the concept. For instance: these standards, as well as the technical solution implemented here, might be used in a limited rollout of the information system to a particular highly populated area.

DESIGN

Design specifications were outlined at the proposal stage of the project. These specifications described the minimum objectives that the solution must necessarily archive to constitute a full implementation of each outlined use-case, as well as an adequate user experience, and accordance with web design best practices.

The most straightforward implementation of the public and administrator functionality was determined to be a public website, which can be accessed on a computer browser or on a mobile browser. While a more mature roll-out would unavoidably require the production of iOS and Android native apps, a universal and responsive website was determined to be the best use of development time (special care was taken to ensure touchscreen usability of the live map—see Live Map Design). To complete its function, this project has to consist of a web server able to accept and process constant incoming reports (of temperature, humidity and barometric pressure at a given location) and display this information on a public webpage both in the form of a list of reports and a live map with a visual indication of each report at its location. From the public perspective, a user must be able to view the live map without logging in, or register for an account and be allowed to submit a report manually and view or edit their past reporting. From an administrator perspective, an admin with full privileges must also be able to login to the website and view and moderate all reports, including editing and deleting, as well as modifying the list of supported regions. Each use case is outlined in the functional diagram (Figure 1). Each path (which can start from the colored ovals) in the diagram represents a single use case.

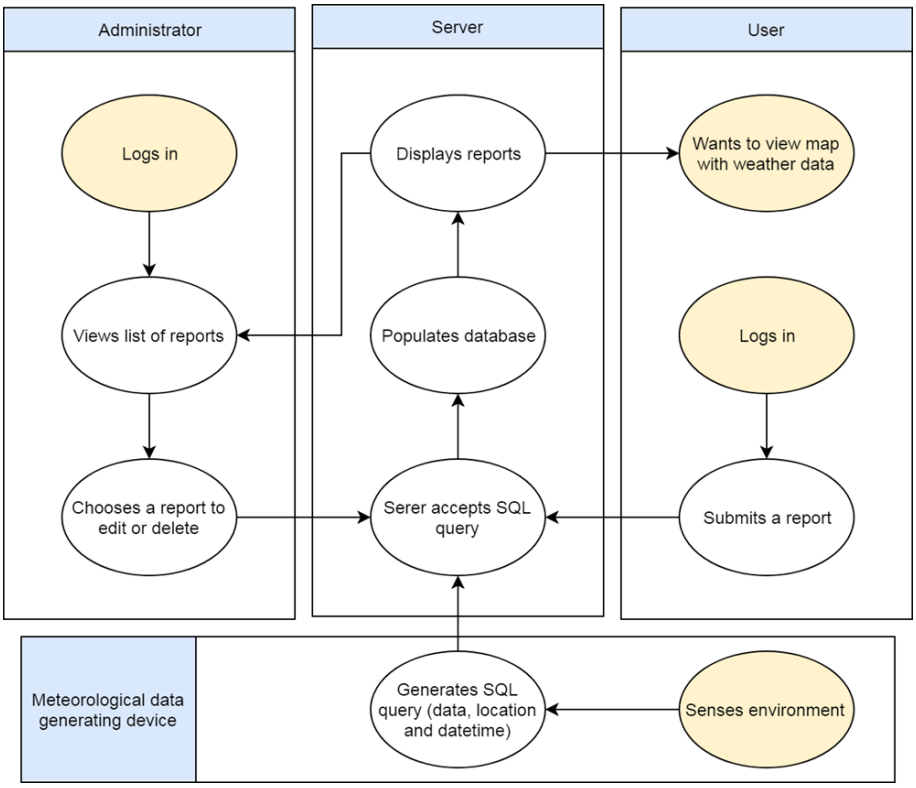

Figure 1. Functional Diagram

Website

The website (Figure 2) was designed to both meet the project specifications, providing for each use case outlined in Figure 1, but also ensure that the conditions of the web design best practices are met.

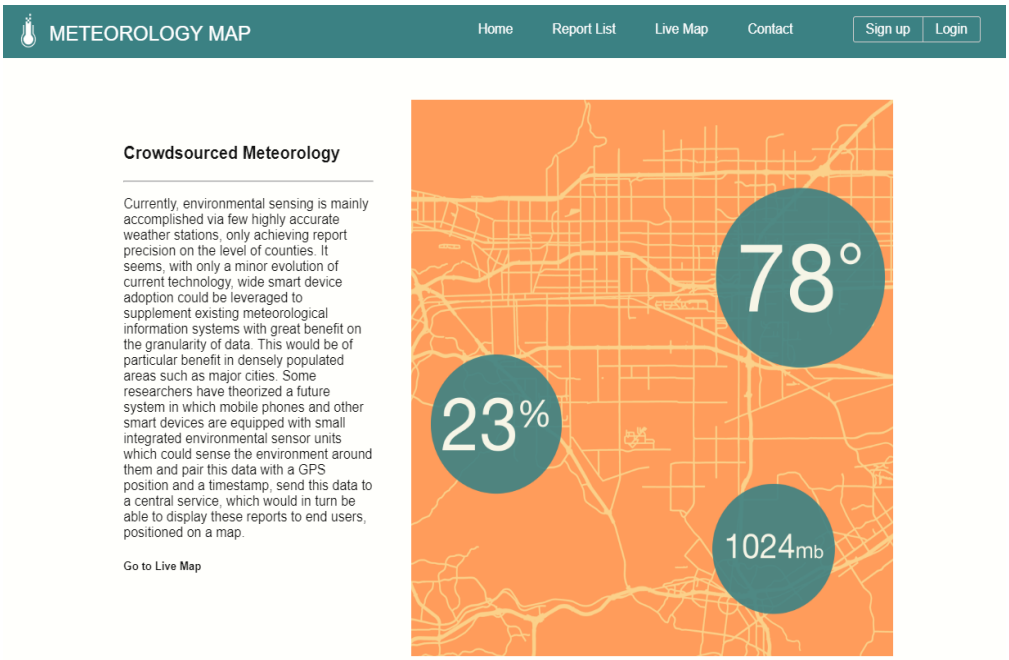

Figure 2. Website Design

Web Page Layout

Standard web design best practices are followed for the website layout to ensure usability. PHP is used to make the design of the header, navigation and footer reusable, for a modular approach to building websites. The index.php file invokes layout.html.php then, based on the route, and generates another PHP file with the content for the page. The webpage displays a navigation header, a main content area which consists on every page of a CSS Grid content layout, and a footer (Figure 2).

Site Navigation Flow

The website navigation is designed with the three perspectives of a new user, a logged in user, and an administrator in mind. The top navigation changes depending on the permission level of the user. To a new user the following navigation options are available:

- Home
- Reports list
- Live map
- Contact
- Login
- Sign up

When logged in either as a normal returning user or as an administrator the last two options change to 'Dashboard' and 'Log out.' Administrator only pages are available in the administrator dashboard (see Figure 9). The dashboard layout changes dynamically on accessing the page based on the user's permissions (from 0 to 63).

Live Map Design

The problem of displaying weather information on a map has been widely studied and iterated on by many different designers. The prototyping utility of this project would have been highly limited by simple displaying weather reports as a list, therefore a live map solution had to be developed. Of course, UX (User Experience) testing to arrive at the ideal method of displaying this information was outside the scope of this project, and the minimal functional design is implemented. In order to be functional the website has to display recent reports in a particular region in the correct position on the map, display additional information with user interaction, and provide basic sorting capability. The final design can be seen in Figure 3.

The live map uses OpenStreetMap, MapBox, and Leaflet APIs to show and control an interactive live map. OpenStreetMap provides the world map data, MapBox provides the in browser display technology, and client-side webpage control of the map is achieved through Leaflet. The client side Leaflet API code written in JavaScript interfaces with the sever database to display reports, in their proper location. Each report icon shows temperature and humidity, and clicking or taping on a report shows all available information. The color of the marker icon corresponds to the temperature. The user is able to sort the reports by report time and report region.

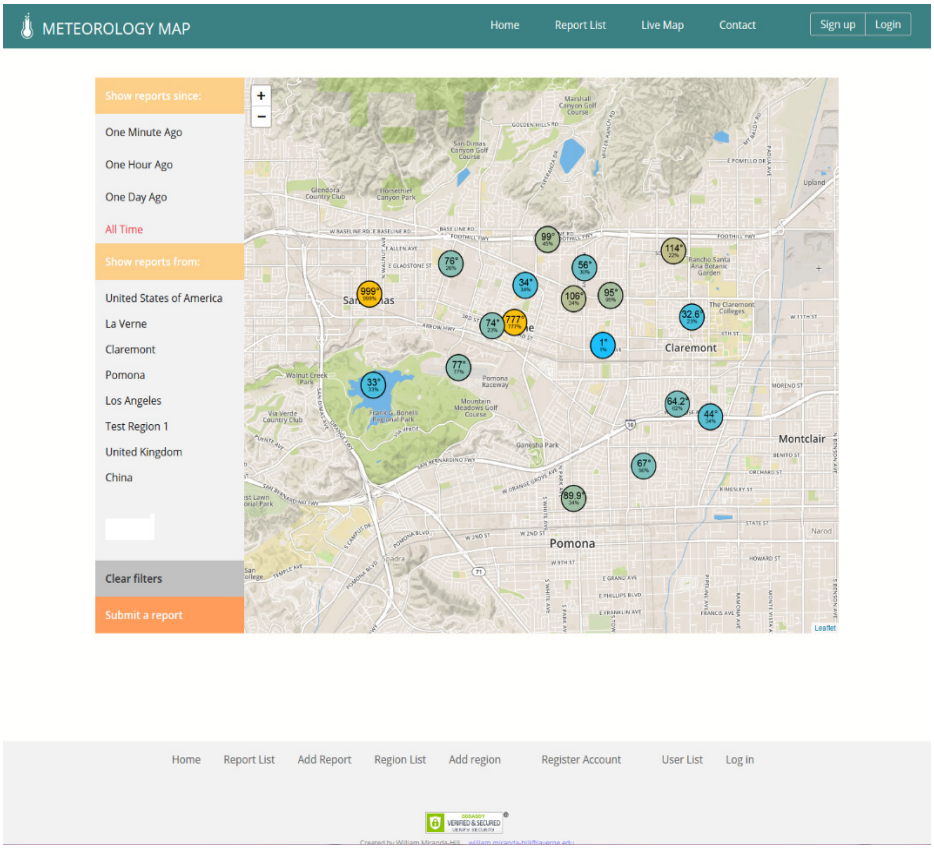

Figure 3. Live Map Design

Server Design

Both HTTP requests for public and admin website pages and remote SQL queries from weather stations has to be fulfilled by this server to meet its design specifications. The server thus must contain a relational database both for the purpose of public and admin users and for collecting reports.

Ninja Framework

This project uses the server-side PHP Ninja framework, documented in the book *PHP & MySQL: Novice to Ninja* by Yank and Butler (2017). It provides a basic framework for modular page layout, routes, querying with a database and displaying results on a website, and a user registration, login and permission system. Expansions and modifications to the function of the framework were required to achieve the desired functionality, mostly concerning the query system.

Database Design

The table relationship with data type for each field is illustrated in Figure 4. The relational database handles reports, users and report regions. Regions are in this prototype defined manually by administrators. Reports are linked to report regions in a look-up table, which are joined to the reports when we query the database. User passwords are stored in hashed form, rather than as plain text.

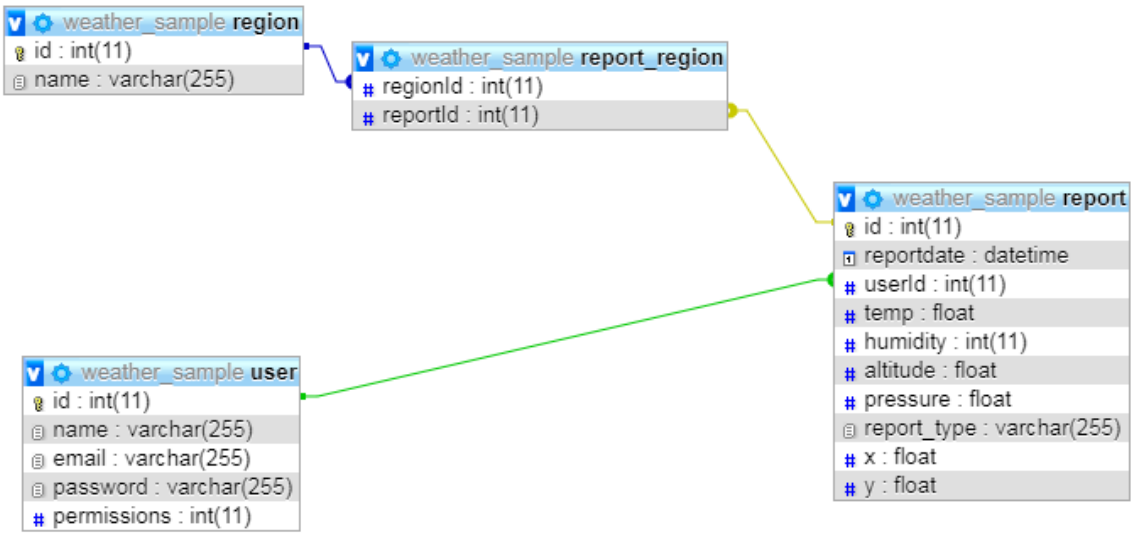

Figure 4. Database Design

Hardware Design

This system is meant to accept live, real-world, geo-positioned, time-stamped environmental data from smart devices. Constraints on the smart device are minimal, aside from some means of generating environmental data, such as the addition of small integrated environmental sensor. It would also need GPS positioning or some other means of supplying longitude and latitude information as part of its reports, an Internet connection, and minimal processing power. Smartphones, tablets, cars, and many IOT devices would be able to fit these constraints. In the case of smartphones the accelerometer might be used to only take measurements when the phone is first removed from a pocket or bag, in order to reduce false temperature reports from body heat (testing the techniques surrounding the prevention of low quality reports from handheld devices was outside the scope of this project).

Prototype Hardware Requirements

For the purposes of testing this prototype, all additional features of the smart devices would be unnecessary. Because of this, a custom build microcontroller system was decided upon. Between the most popular programmable microcontrollers, the Raspberry Pi and the Arduino, the Arduino was selected, as the operating system and additional features of the Raspberry Pi were deemed unnecessary for meeting the design specifications outlined. It should be noted that Grove connection to Raspberry Pi is possible and the Raspberry Pi is capable of remote SQL connection, thus if the pro-

gram was remade in Python, instead of C, the same components could be used with a Raspberry Pi to exactly the same effect.

The Arduino program was developed in C. It uses a publicly available Arduino remote SQL library to establish a connection to a remote SQL server. The final design used in testing (Figure 5) connects to the Internet over Ethernet, however the same library is equally capable of connecting over Wi-Fi. Submission of the query was tested as usually requiring around 10 milliseconds, and an artificial delay of 1 minute was added after every submission. It was decided that a period of one minute was sufficient temporal resolution to account adequately for the speed of natural environmental changes. This delay is completely arbitrary and could be changed dynamically based on environmental conditions as to minimally waste sever I/O resources.

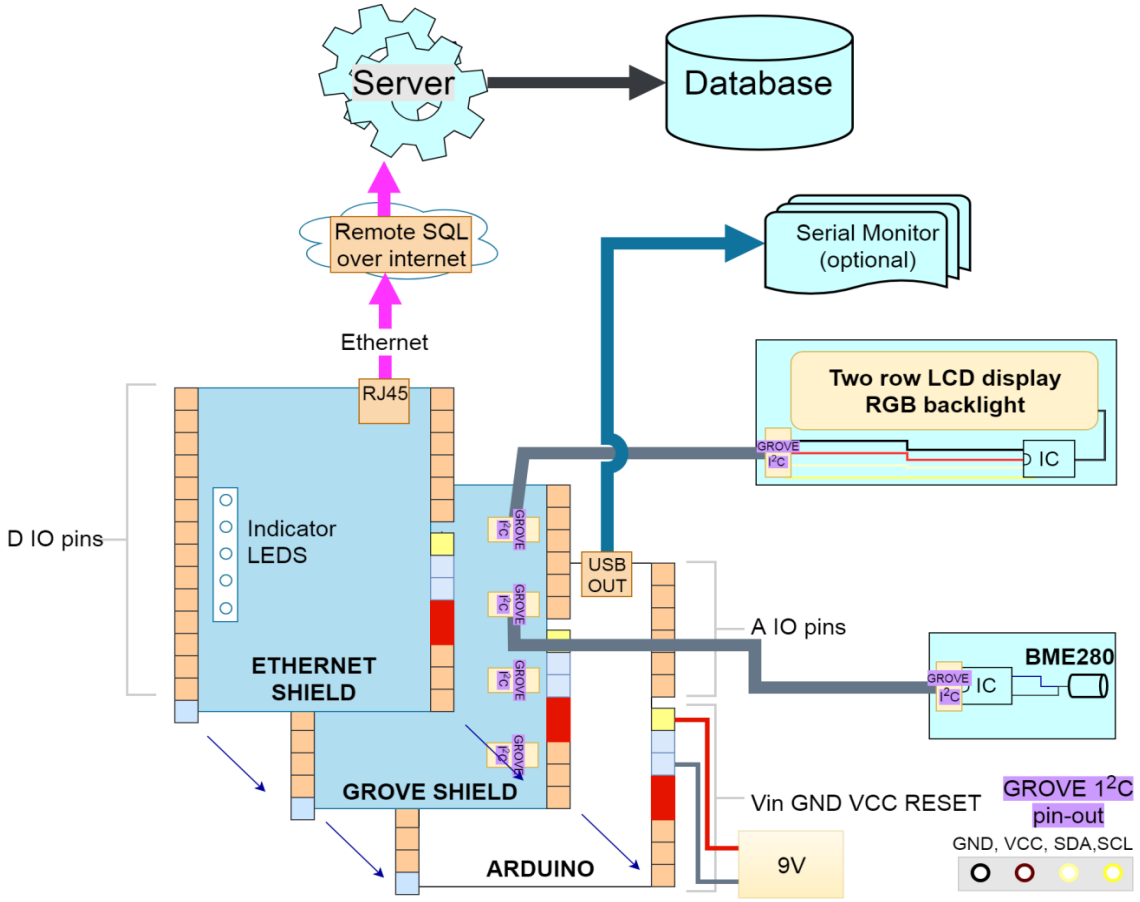

Figure 5. Hardware Diagram

Final Design Explanation

A diagram of the final hardware component used in testing is seen in Figure 5. The Arduino microcontroller board constitutes the "brain" of the stand-in smart device. To meet the hardware requirements of, the component needed a way to connect to the Internet over USB, a way to interface with a LCD display (for immediate feedback when not connected to the serial monitor) and a way to interface with an environmental sensor unit. Both the LCD display and the environmental sensor unit are governed by their own ICs and required use of the I2C interface protocol. The I2C protocol requires ground, VCC, serial data, and serial clock connections to interface two ICs. Rather than to wire them directly to the I/O pins of the Arduino, the most elegant solution is to use the Grove connector standard and a Grove connector Arduino shield.

After the Arduino is powered and initiates its program, the data flow starts at the BME280, which senses the environment, generating multiple floating point numbers. The data is sent to the Arduino, which uses it to populate the LCD display and to generate an SQL query. The Arduino upon first being initiated attempts to connect to the Internet through its Ethernet shield. After the integrity of the connection is established the Arduino attempts to connect to the remote SQL server with a username and password. If the login is successful, it will continually accept more data from the environmental sensor, generate SQL queries, and submit them to the remote SQL server, indefinitely.

IMPLEMENTATION

The project is published publicly on a personal domain, only intended for limited testing. This implementation lacks the resources to handle the many concurrent connections that would be required even for a limited rollout of the system. Public users can visit the website and create an account, view the live map, and submit reports manually. It currently does not accept ad-hoc environmental sensor connections. While the website could theoretically accept automatic weather reports from any internet connection in the world, there is no way to submit this data without being added to the remote SQL whitelist at this time. The current system interfaces directly with the remote SQL connection of the XAMPP server, and thus this system is only acceptable for the small number of prototype devices created as part of the project. We found this implementation to be adequate for basic testing. Figure 6 shows the sample of serial monitor output of Arduino. Each minute, it outputs the generated SQL query to the serial monitor at the same time it submits the query to the remote SQL server.

Time Stamp	Serial Output
04:07:08.966	INSERT INTO weather_sample.report (pressure,altitude,temp,humidity,x,y,report_type)VALUES(981.920,264.126,69.1,63,34.0843,-117.723,'arduino')
04:08:09.016	INSERT INTO weather_sample.report (pressure,altitude,temp,humidity,x,y,report_type)VALUES(981.920,264.126,69.1,63,34.0843,-117.723,'arduino')
04:09:10.020	INSERT INTO weather_sample.report (pressure,altitude,temp,humidity,x,y,report_type)VALUES(981.910,264.211,69.2,63,34.0843,-117.723,'arduino')
04:10:11.021	INSERT INTO weather_sample.report (pressure,altitude,temp,humidity,x,y,report_type)VALUES(981.880,264.467,69.1,63,34.0843,-117.723,'arduino')

Figure 6. Sample of the Arduino serial monitor output

TESTING

The project was tested from the perspective of public users, administrators and an autonomous datasource. Each use-case was tested for functionality and proper user experience

WEBSITE

By the nature of this service and its exploitation of smart device adoption, a full deployment would clearly use mobile apps. These apps would be for IOS and Android, as well as designed to integrate with other smart devices, with a native app on the device or interfacing through an IOS or Android device as it common with IOT devices. We found each webpage completed its function; however, iterations on the original design specifications had to be made to assure passable user experience (see Solution Analysis below).

Server

This web server used Apache and MySQL to handle requests and store the information. It was deployed on a public LAMPP server. The server's ability to generate each webpage correctly, manage accounts, and accept reports from both users and remote SQL connections, was tested as completely functional.

Hardware

The Arduino assembly is implemented and tested as working. It is able to generated accurate live reports of temperature, humidity and barometric pressure, couple them with a timestamp and longitude and latitude point, and generate an SQL query based on this data. It is then able to connect the remote SQL server and submit the query, which in turn properly updates the database within milliseconds. Whenever it is turned on with power and an Ethernet connection it proceeds to send reports to the database automatically (if the IP of the network the Arduino is connected to is on the Remote SQL whitelist). In the final version, the USB connection to a computer is optional and is only used to trace the device's actions over the serial monitor.

Solution Analysis

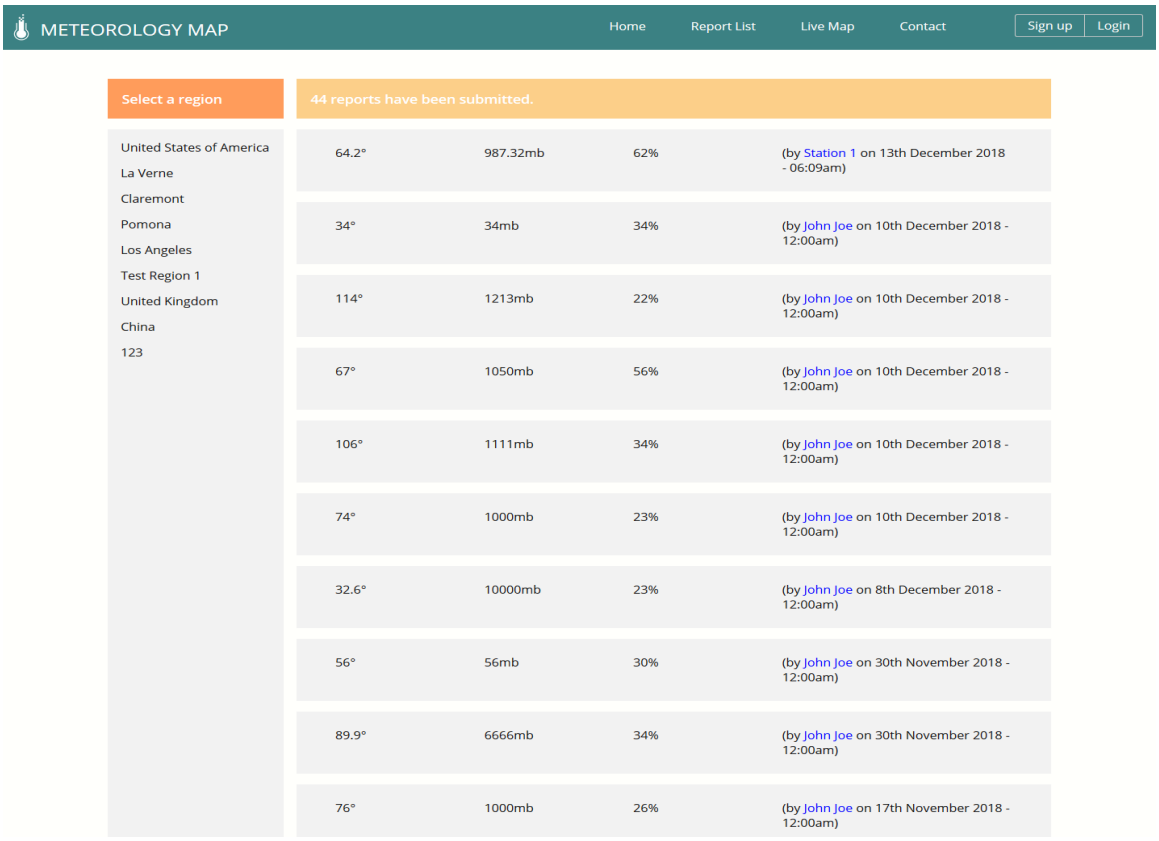

Figure 7. Plain-text report list

The stated objective for this project is to prototype the basic functionality and use cases of a theoretical future user-generated meteorological information system. From testing each use case, even at this small scale, the utility of a plain-text list of reports, shown in Figure 7, seems to fall below that of the live map. This user-interface is certainly useful for viewing one's own contributions, and for the purpose of administration. We suggest that such an interface only be provided for those two use-cases, and that reports from all local sources need only be visible on the live map.

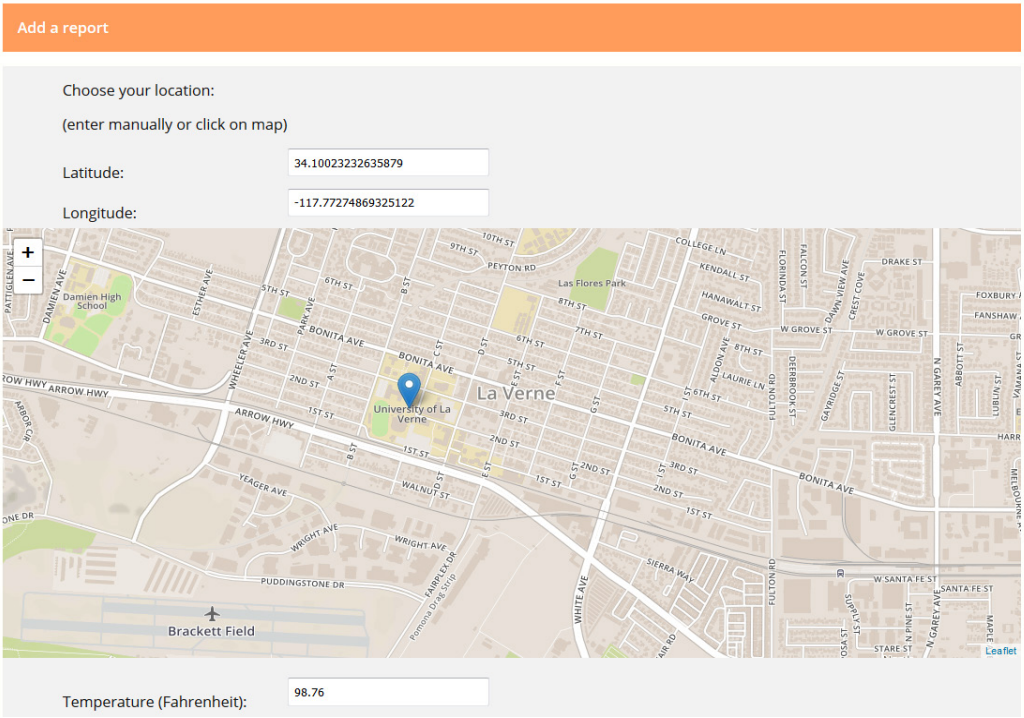

Figure 8. Selection for manual report (Map data © OpenStreetMap contributors)

We speculate that a fully manual report function would only be useful if utilizing an adoption strategy that involves 'citizen scientists,' as are described in Muller et al. (2015). A more mature system would likely only require the 'semiautomatic' and 'fully automatic' models described by Muller et al. and Cuff et al., in which the environmental sensor in one's smart device uploads its data upon command, or automatically. However, we found that a public interface for such a system (Figure 8) must, for acceptable user experience include a click-on-map solution for generating longitude and latitude. Due to the quantity of options available to logged-in users, we also found it a benefit to user experience to include functions such as adding reports not as submenus in the main pages, but rather to implement a consolidated dashboard for user actions (Figure 9).

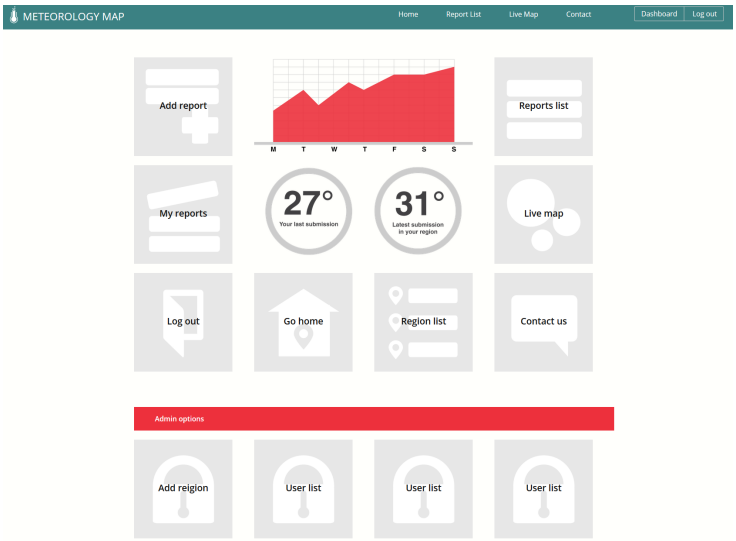

Figure 9. Consolidated dashboard (Admin permissions)

CONCLUSION

Previous research theorized a metrological information system based on geopositioned user reports from smart devices, mainly from the perspective of value to scientific research and quality of data, however, they did not attempt to identify a set of minimum uses-cases it would require, nor design and implement these use-cases in a functional prototype. Each element of the prototype, the public website, administration, server, and hardware components, were implemented successfully and in accordance with the outlined design specifications. Each use-case the design specifications were meant to allow for was tested on a basis of functionality and user experience. Each use-case was functional, and few interactions on the original specifications had to be made to ensure acceptable user experience. Though the project lacks the architectural scalability required for public release, we nonetheless believe this project severed the stated purpose and acted as a proof of concept for the previously theorized meteorological information system. Thus, as outlined in the abstract, this project goes to establish practically the possibility of this system and the validity and functionality of the basic use-cases we identified for a crowdsourced meteorology solution. It also allowed us to test each aspect of the system at a smaller scale. Through the implementation of the project we were able to assess the quality of the uses-cases outlined. We found the project was a functional information system, with each public server-side and hardware component interfacing correctly and most limitations resulted from the scope of the project. This result in our eyes unlocks the possibility of the next step towards the full realization of the theoretical information system: a real-world rollout, perhaps limited to a particular densely populated area.

REFERENCES

Chapman, L., Bell, C., & Bell, S. (2017). Can the crowdsourcing data paradigm take atmospheric science to a new level? A case study of the urban heat island of London quantified using Netatmo weather stations. *International Journal of Climatology, 37*(9), 3597-3605. https://doi.org/10.1002/joc.4940

Cuff, D., Hansen, M., & Kang, J. (2008). Urban sensing. *Communications of the ACM, 51*(3), 24-33. https://doi.org/10.1145/1325555.1325562

Kanhere, S. S. (2011). Participatory sensing: crowdsourcing Data from mobile smartphones in urban spaces. *IEEE 12th International Conference on Mobile Data Management, Lulea*, pp. 3-6. https://doi.org/10.1109/mdm.2011.16

Muller, C., Chapman, L., Johnston, S., Kidd, C., Illingworth, S., Foody, G., Overeem, A., & Leigh, R. (2015). Crowdsourcing for climate and atmospheric sciences: current status and future potential. *International Journal of Climatology, 35*, 3185-3203. https://doi.org/10.1002/joc.4210

Overeem, A., Robinson, J. C. R., Leijnse, H., Steeneveld, G. J., Horn, B. K. P., &. Uijlenhoet, R. (2013). Crowdsourcing urban air temperatures from smartphone battery temperatures. *Geophysical Research Letters, 40*, 4081–4085. https://doi.org/10.1002/grl.50786

Yank, K., & Butler, T. (2017). *PHP & MySQL: Novice to Ninja* (6th ed.). SitePoint.

BIOGRAPHIES

William Miranda-Hill is majoring in Computer Science, with a concentration in Internet Programming, at the University of La Verne in La Verne, California. His interests include full-stack web development, rich web applications, software development in the .NET framework, robotics, and machine learning.

Jozef Goetz received his M.S. degree in Computer Science, minored in Electronics from Wroclaw University of Technology, Poland. He received his Ph.D. in Computer Science with Honors from Cybernetics Institute of Wroclaw University of Technology. He is currently a professor in Department of Computer Science, University of La Verne, US. Jozef Goetz has a balanced academic experience as well as technical, hands-on, 12-year experience in industry as a senior software developer engineer at Fujitsu. His broad academic experience consists of teaching and research first at the Wroclaw Polytechnic University, Poland, then the California State University, Fullerton, US, finally the University of La Verne. He teaches seven Internet Programming classes. His interests include WEB and mobile app development, website design, software development, design best practices, e-commerce, Web analytics, programming and modeling systems using Petri Nets.