Smart Water: a Prototype for Monitoring Water Consumption

Eduardo Hoeltgebaum Pereira
Department of Systems and Computing
Regional University of Blumenau
Blumenau, SC, Brazil
pereiramateduardo@gmail.com

Aurélio Faustino Hoppe
Department of Systems and Computing
Regional University of Blumenau
Blumenau, SC, Brazil
aureliof@furb.br

Miguel Alexandre Wisintainer
Department of Systems and Computing
Regional University of Blumenau
Blumenau, SC, Brazil
maw@furb.br

ABSTRACT

The traditional metering system of water meters, through human readers, although still widely used by companies that provide water service, tends to become an increasingly unviable process over the years, due to urban growth. With the objective of finding a solution to this question, this paper presents the development of a prototype to monitor water consumption and an application that allows the end user to visualize his consumption. For the prototype, it was used the NodeMCU module, because of it being a low-cost device, along with a Wisol WSSFM10R2 Breakout module, which allows communication through the Sigfox network, considered an alternative network for IoT communications, using simple AT commands, besides the Sigfox company provide all the architecture for the developer. The present work also discusses about how the Sigfox hardware and network works, explaining the pulse conversion processes emitted by the flow and pressure sensors, as well as the use of the NodeMCU module for control and sending of messages through the Sigfox network. In general, the prototype obtained a satisfactory result in relation to the calculation of water consumption, reaching accuracy rates above 90% in tests that used the values returned by both sensors in constant flows and an average accuracy rate of around 99% for tests with varied flows, where it has been proven that the use of the pressure sensor optimizes the consumption calculation.

KEYWORDS
Water consumption, Monitoring, Sigfox

1 INTRODUCTION

Water is an essential resource for humans, being necessary from the absorption of food to the elimination of excrements. Every living being must keep its water supply close to normal, otherwise it dies. A man can live without solid food for more than a month, but without water he can only live for two or three days. For water to reach the population, being suitable for consumption, it is required a whole process of obtaining, treatment and finally, enabling the distribution. Firstly, water must be captured in water springs, which may be lakes, rivers or underground water, and soon after, a treatment process is performed, in order to become pure and fit for consumption [1].

Once the water is available to the population, it is necessary to control the consumption of each residence attended by the distributors, through the information contained in the water meters. Currently, it is a manual process, requiring the presence of a person responsible for visiting the houses and registering the value contained in the hydrometer [2]. Although this process can be acceptable nowadays, it will become more complex and slower, due to the urban growth. Therefore, the use of microcontrollers can propitiate the development of solutions that seek the connection of devices in network, managing industrial and residential processes. To make it possible the read of water meters in a short time, remote metering systems were developed, reducing the reading time and without manpower cost. In Brazil, there are already suppliers for this new measurement system, although the cost of implementation is more expensive than the traditional system [3].

Also, it would be interesting the development of a tool able to display the water consumption of each end user, thus allowing him to control unnecessary use of water and avoiding the waste of this resource [4].

As related works to the current project, three researches were used. Gokilapriya and Bhuvaneswari [5] developed a low-cost system for monitoring water consumption, alerting consumers to detection of substandard consumption. For the solution, they use NodeMCU and the Ubidots tool for data storage. Data is captured through a flow sensor and sent to the cloud. In the event of an anomaly in consumption, an alert is sent to the consumer via SMS, Telegram or email. According to the authors, the results were satisfactory, with the difference between the actual value and the captured value being small.

Mudumbe and Abu-Mahfouz [6] also developed a system for monitoring water consumption, enabling the visualization of water consumption in real time. Data capture is performed through an analog Reed switch and is sent to a gateway, which forwards the information to a server. For the capture prototype, they used a Dizic module, flash memory and a power system, containing a solar panel and a battery. Communication between the prototype and the gateway occurs through the 6LoWPAN protocol, while the communication between the gateway and the server is via the IPv6 protocol. The authors considered the results satisfactory due to the identification of a constant hydraulic leak near the company, which, according to them, would go unnoticed for a long period.

Schwabe and Van Laerhoven [7] sought to present water consumption in real time, making consumers aware of their water consumption, also called eco-feedback. For the prototype, they use the NodeMCU module, as well as the BMX055 magnetometer and the VCNL4010 infrared sensor to capture the data. After capture, the data is sent to a server using Wi-Fi networks. In order to process the values returned by the sensors, it was concluded that using the Zero-crossing technique would be the best alternative, due to the low memory consumption. Regarding the results, the authors considered it reasonable, where the prototype lost some rotations and, consequently, presented a considerable variance in relation to the real value.

Given the above, this paper presents the development of a prototype able to monitor water consumption in residences. It is believed that the process of data collection will be facilitated and will allow the consumer to monitor his water consumption.

2 SIGFOX

Sigfox is a telecommunications operator, responsible for the development of wireless networks, in order to connect low-consumption devices to the Internet or to each other [8]. Currently, it is available in 32 countries, including Brazil [9]. The Sigfox network is considered a Low Power Wide Area Network (LPWAN), due to the use of ISM (industrial, scientific and medical) bands of 868 MHz in Europe and 902 MHz in the United States and Latin America, which allows to overcome solid objects and cover vast areas [10]. By using these radio bands, also called Ultra Narrow Band (UNB), Sigfox guarantees a very low level of noise during data transmission and consequently results in a low power consumption in the modules that communicate through it [11]. The low level of noise in the network is due to the signal having a very small bandwidth, so the receiving stations will filter the messages through this low bandwidth, removing much of the noise that may occur throughout the transmission process. The message range at this signal frequency can exceed 30 km in rural locations [8].

Among the technology principles used by the Sigfox network are: 1) random access to the channel; 2) cooperative reception; 3) sending small messages. Random access to the channel serves to enable the quality of service, in which the device sends a message at a random frequency and then sends two more replicates at different frequencies, in order to try to ensure reception at the base station. Cooperative reception allows a device not to be linked to only one station. The message sent by the device, called uplink, is received by a base station next to it. On average, there are 3 base stations close to the device. The principle of small messages is used to ensure the battery autonomy and consequently, low the cost of the devices [9]. The size of a message of type uplink varies from 0 to 12 bytes of payload, in addition to 14 bytes of overhead, which contains information regarding module, timestamp, authentication and error detection. The maximum number of uplink messages that can be sent is 140 messages per day [8].

There are also messages sent from the base stations to the devices, which are called downlinks. These downlinks only occur when a data transmission has been done and are limited to 4 messages per day, for each device. The maximum payload length of a downlink message is 8 bytes. Due to this limited number of downlinks, the Sigfox network uses the principle of random access to the channel, as mentioned earlier [11]. In order to activate the downlink messages, it must be indicated in the payload of an uplink that is expected to receive the message, besides activating the option of sending downlinks in the backend of Sigfox.

Regarding the network architecture, there are two main layers: 1) Network Equipment, which corresponds to the layer of the base stations that receive messages from the devices and send them to the next layer of the network; 2) Sigfox Support Systems, in which the messages are processed, stored and sent to client systems through call-backs, in addition to modules and features that ensure network deployment, operation and monitoring, ensuring operation of the network. The communication between these layers is through a Virtual Private Network (VPN) [9]. The Figure 1 depicts the Sigfox network architecture.

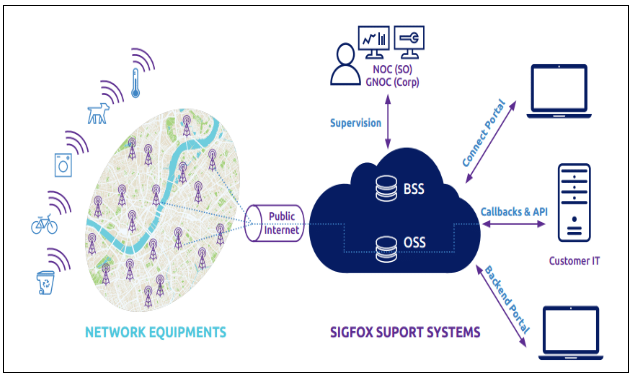

Figure 1. Sigfox network architecture

To retrieve the messages stored by Sigfox, there are two ways: 1) through the Representational State Transfer (REST) API provided by Sigfox; 2) configuring a call-back, so that as soon as the network receives a message, it sends the information to a URL or an email address. The best way is to use the call-back option, because through the REST API, it would be necessary to constantly send HTTP requests, checking if a new message has been received by the network. Therefore, it is recommended to use the API to retrieve geographic information and to add or remove devices on the network. Besides sending messages to the client, the call-back service also can send errors occurred during the communication process and information about the device, such as battery level, identifier and temperature.

3 IMPLEMENTATION

The architecture of the project, which is illustrated in the Figure 2, has three major elements: 1) the prototype to collect the consumption information, using flow and pressure sensors; 2) the backend, which consists of a server running on the Heroku platform, processing and saving data in a Firebase database; 3) an interface, developed with the Angular framework, allowing the end-user to visualize his water consumption.

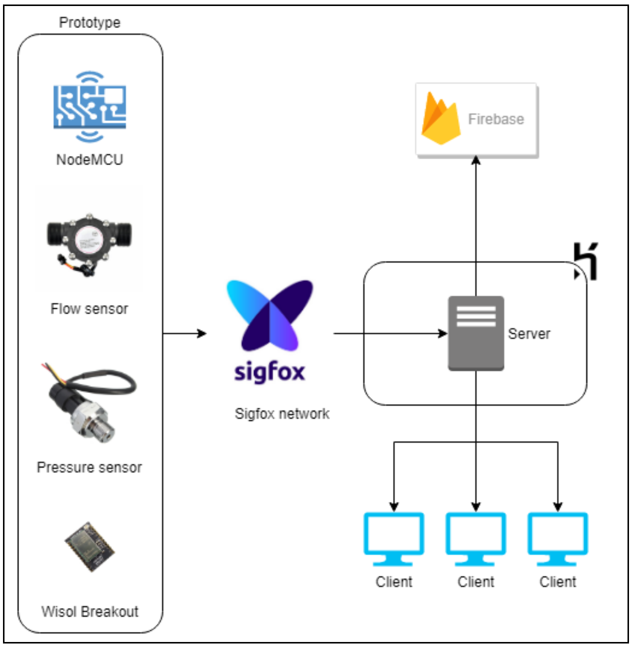

Figure 2. Application architecture

3.1 Prototype

For the prototype, it was used the NodeMCU module [12], because it is small and a low-cost device. The process of capturing the water consumption is done through the YF-S201 water flow sensor [13] and the SKU237545 water pressure sensor [14].The communication with the Sigfox network is performed by the Wisol WSSFM10R2 Breakout module. Regarding the energy, it was used a MB102 power supply, developed by the YwRobot company, providing 5 V voltages, necessary for the sensors existing in the project.

The working principle of the flow sensor is simple, sending digital pulses every time a rotation is performed by the gear that exists inside him. Regarding the water pressure sensor, his working principle consists of a diaphragm that is deformed based on the pressure applied over him, sending the output value in analog pulses.

This way, the calibration of the flow sensor must be performed, in order to convert the amount of pulses emitted by it in the amount of water consumed. For this, the FlowMeter library was used, where properties must be defined for the sensor, as well as the pin in which it is connected to the NodeMCU. The required properties are: 1) the maximum flow rate supported by the sensor; 2) a factor K, which corresponds to the frequency of pulses generated by water flow, in l / min; 3) a list of ten M factors, which correspond to correction factors to be applied on the factor K, in order to make the final value accurate. For the definition of the values of the properties, some manual calibration tests were performed. At the end of the process, it was used the value 8.8 for the K factor, and the values 0.9, 1 and 1.05 for the M factors.

After setting the properties, an interrupt routine must be created, that only increments the pulse count. To perform the conversion, a normalization process is realized, using the K and M factors configured. The Figure 3 shows the steps to convert the digital pulses.

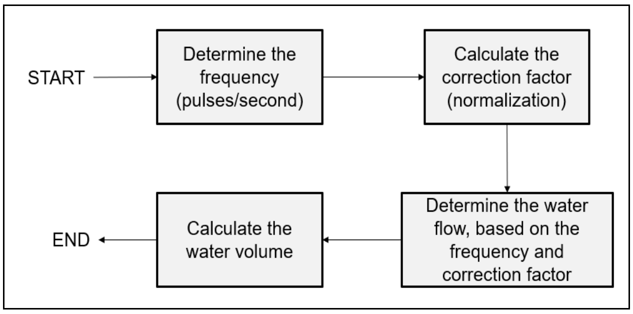

Figure 3. Process of conversion pulses into water volume

For the communication part of the hardware developed, the Wisol WSSFM10R2 Breakout module was used, which is marketed by the company SeaSlugLabs. The module sends the information to the Sigfox network through AT commands. To transmit the data to the application backend, a serial communication channel is created between the NodeMCU and the Wisol module, using the SoftwareSerial library. Creating it requires only the number of the GPIOs used as RX and TX on the NodeMCU.

To send the data via Sigfox, a test command is performed, to ensure that the Wisol module can pass the information. After this step, the transmission process begins. First, it must verify that the channel used by the Wisol module is configured correctly. For this, the command AT$GI? is used, returning two values (X and Y). Afterwards, the return is validated and, if the channel is unconfigured, the AT$RC command is executed to configure it. Once it is ready, the data is sent through the AT$SF command, and the parameter 1 is added at the end of the payload, so that in addition to sending the message, the module receives the send confirmation response from the server. The Figure 4 illustrates the process explained above.

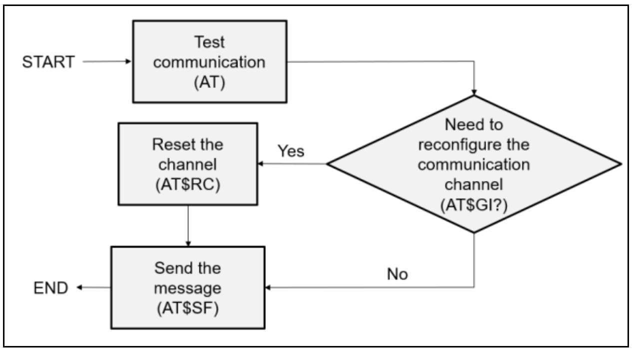

Figure 4. Process of transmission via Sigfox

3.2 Backend

The backend of the application consists of a Node.js server [15] running on the Heroku platform [16], besides the Firebase database [17]. To allow the integration between the developed server and the data received by the Sigfox network, a callback was configured on the Sigfox network to send a request to the server, storing the information in the database subsequently. The callback was defined as data sending (DATA), which is a bidirectional communication (BIDIR). The message is sent through a HTTP request, defining the URL in the Url pattern field. The request has an authorization header, using a JWT token, and in the body are sent the device ID information, the data received by the network (date), as well as the time sent, in JSON format.

With the callback created, an uplink route was configured in the server, to handle the request sent by Sigfox. First, it validates if the request is correctly authenticated. After the validation, the contents of the request are stored, returning a response with status 200, in addition to the payload of the downlink message that will be sent to the Wisol module, confirming the reception of the message.

In order to query the data, two routes were created: 1) a yearly messages, to return water consumption over a year; 2) a monthly messages, which gets water consumption over a month. These routes first ensure that the requesting user is authenticated and after verification, perform a database query, returning the water consumption values according to the calling user.

User authentication is performed through the credentials route, which checks the database for any users with the given login and password. If so, it returns the customer a valid 3-hour JWT token, plus the Wisol module ID, so they can only verify their water consumption data.

4 RESULTS

To prove the effectiveness of the proposed solution with regarding the calculation of water consumption, three experiments were performed. The experiments consisted of connecting the prototype to a conventional water tap and allowing water to pass through it using a silicone hose. At the end, the water filled a 10-liter graduated bottle. For the first experiment, only the value returned by the prototype flow sensor was considered, performing low, medium and high flow tests, using volumes of 3, 5 and 10 liters. Table 1 presents the results obtained.

Volume (liters)	Flow kind	Average value read (liters)	Accuracy rate (%)
3	High	3.03	99%
3	Medium	2.92	97%
3	Low	2.05	68%
5	High	5.28	95%
5	Medium	4.90	98%
5	Low	3.76	75%
10	High	9.97	99%
10	Medium	9.43	94%
10	Low	7.97	80%

Table 1. Results using only the flow sensor value

As can be seen, the prototype, considering only the values of the flow sensor, performed well for the high and medium flow tests, obtaining an average accuracy of 97%. However, for low flow tests, the accuracy rate was low, especially for tests with volumes of 3 and 5 liters, reaching an average of 74% accuracy. This low level of accuracy is believed to be related to the FlowMeter library conversion calculation, being optimized only for high and medium flow rates, omitting readings at low flow rates.

Thus, in the second experiment, besides the flow sensor value, the output value generated by the pressure sensor was also used. For this, an adjustment factor was calculated, establishing a ratio between the water flow and the pressure sensor output. The adjustment factor was 2.5. To validate this calculation, the same tests were performed as the first experiment, and the results are shown in Table 2.

Volume (liters)	Flow kind	Average value read (liters)	Accuracy rate (%)
3	High	2.93	98%
3	Medium	2.85	95%
3	Low	3.09	97%
5	High	5.14	97%
5	Medium	4.93	99%
5	Low	5.05	79%
10	High	9.81	98%
10	Medium	9.63	96%
10	Low	10.15	98%

Table 2. Results using both sensors values

As shown, the results at high and medium flow rates remained somewhat accurate compared to the previous experiment, while the results at low flow rates obtained a significant improvement, increasing the accuracy rate by up to 19%. This increase in accuracy was due to the fit calculation to optimize the results at low flows, where the flow-to-pressure ratio is low and, consequently, increasing the consumption value at each iteration of pulse to water conversion.

Finally, in the third experiment, intermittent tests were performed to verify how the prototype would respond when the water flow was altered along the filling of the graduated bottle. For these tests, the water flow was changed every 10 seconds, ranging from low, high, medium and no flow, exactly in this order. In addition, volumes of 3, 5 and 10 liters were also used. Table 3 presents the results at the end of the experiment.

Volume (liters)	Average value read, using only the flow sensor (liters)	Accuracy rate (%)	Average value read, using both sensors values (liters)	Accuracy rate (%)
3	2.81	94%	2.99	100%
5	4.69	94%	5.03	99%
10	9.29	93%	9.94	99%

Table 3. Results of intermittent tests

It can be seen from Table 3 that although the results with the flow sensor alone were reasonable, achieving an average accuracy of 94%, the use of the pressure sensor with the fit calculation was able to improve the results, obtaining an average accuracy of 99%.

Thus, it is possible to assume that the use of the pressure sensor allowed the optimization for low water flow calculations, due to the limitation in relation to the FlowMeter library, proven throughout the experiments performed.

5 CONCLUSIONS

Through the development of this work, it was possible to verify the feasibility of the implementation of smart residential water meters, which take the readings and send the consumption data to a server, where the user can consult them. The final cost of the system was estimated at around R$ 200. The solution presented has a good accuracy rate in relation to the actual consumed value of water, using a somewhat reliable and long-range communication network. However, the present project does not address the energy expenditure issue of the developed prototype, which is considered a relevant and necessary aspect to ensure that the developed solution is economically viable.

The Sigfox network proved to be a technology, at first, ideal for the project in question, as it is an alternative network that allows the communication of IoT devices, using only AT commands, considered simple and easy to understand, besides providing all the network architecture to the developer, requiring only the purchase of a module that communicates through it and that the developer is properly registered with the Sigfox backend service. However, some of the limiting aspects of this include the maximum number of downlink messages, where the developer should be stuck with only 4 daily uplink messages if all messages sent must have a server commit response. Another limiting point is in relation to the coverage area, because as it is still a technology in expansion, in certain points of Blumenau, there is no signal, making it impossible to use the present work throughout the municipality. Thus, it would be interesting to verify the use of other alternative networks, such as LoRaWan or even NB-IoT, in order to increase the reliability regarding the confirmation of data sending and that it has a larger coverage area, seeking to make the project into a commercial product.

In addition to the use of alternative networks for sending data, the possibility of using communication over Wi-Fi networks can be verified, due to the ESP8266 microcontroller, present in the NodeMCU module. Thus, if sending data through the Sigfox network was malfunctioning, so that there was no confirmation that the server had received the information, the data could be sent over a Wi-Fi network by taking advantage of this feature in prototype.

Regarding the prototype calibration calculations using flow and pressure sensors, in addition to the FlowMeter library, a high accuracy rate was verified through the experiments, making the solution reliable with respect to the water consumption measurement process. However, it would be interesting to study the possibility of changing the calibration process without the need to use the FlowMeter library, due to its limitation in low water flows, through a proper calculation for the flow sensor, without the need to use the pressure sensor, thus lowing the cost of the developed solution.

In addition to the elaborate application, the present work was relevant for presenting the operation of the Sigfox network, as well as its use, as it is a relatively new and still expanding technology, seeking to contribute to future work in the area of Internet of Things, who will use it as a communication base for the development of their applications.

REFERENCES

[1] Bruni, J. C. A água e a vida, Tempo Social, vol. 5, no. 1-2, pp. 53-65, Dec. 1993. Accessed on Sep. 25, 2018. [Online]. Available: dx.doi.org/10.1590/ts.v5i1/2.84942

[2] Pureza, S. A. Pereira, E. C. and Machado. C. M., Modelo Heurístico para Otimização de Rotas nos Serviços de Leitura de Hidrômetros, in Congr. Nac. de Matemática Aplicada e Computacional, 2014, pp. 1-7. doi: 10.5540/03.2015.003.01.0331.

[3] Minucci, A. S. and Lima, F. A. S. Automatização no Processo de Leitura de Hidrômetros, B.S. thesis, Curso de Tec. em Automação Industrial, Univ. Tec. Federal do Paraná, Ponta Grossa, PR, 2013. [Online]. Available: http://repositorio.roca.utfpr.edu.br/jspui/handle/1/9713

[4] Lima, C. A. F. and Navas, J. R. P. Automação e medições inteligentes para incentivar o consumo consciente de energia e água, in Simp. de Automação de Sistemas Elétricos, 2011, pp. 1-8. [Online]. Available: https://www.knbs.net.br/site/wp-content/uploads/2012/10/Artigo-IX-Simpase-Sistema-Automacao-Medicoes-final.pdf

[5] Gokilapriya, V. and Bhuvaneswari, P. T. V. Design and development of SoC based residential water meter monitoring system, in Int. Conf. on Wireless Communications, Signal Processing and Networking, 2017, pp. 2163-2166. doi: 10.1109/WiSPNET.2017.8300143

[6] Mudumbe, M. J. and Abu-Mahfouz, A. M., Smart water meter system for user-centric consumption measurement, in Int. Conf. on Industrial Informatics, 2015, pp. 993-998. doi: 10.1109/INDIN.2015.7281870

[7] Schwabe, N., Scholl, P. M. and Van Laerhoven, K. An Ad-Hoc Capture System for Augmenting Non-Digital Water Meters, in Int. Conf. on the Internet of Things, 2016, pp. 25-33. doi: 10.1145/2991561.2991576

[8] Barbosa , R. M. R. Sistema de monitorização do consumo de água utilizando tecnologia Sigfox, M.S. thesis, Curso de Engenharia, Fac. de Engenharia da Univ. do Porto, Porto, 2017. [Online]. Available: https://repositorio-aberto.up.pt/handle/10216/106909

[9] WND Brasil, Uma visão técnica da Rede Sigfox, Aug. 2017. Accessed on Sep. 29, 2018. [Online]. Available: https://www.embarcados.com.br/uma-visao-tecnica-da-rede-sigfox

[10] Fagundes, E. Sigfox uma plataforma para IoT, Sep. 2017. Accessed on Sep. 29, 2018. [Online]. Available: http://efagundes.com/blog/sigfox-uma-plataforma-para-iot

[11] Mekki, K. et al. A comparative study of LPWAN technologies for large-scale IoT deployment, Ict Express, vol. 5, no. 1, pp. 1-7, Mar. 2019. Acessed on: Jun. 26, 2019. [Online]. Available: doi.org/10.1016/j.icte.2017.12.005

[12] Handson Technology, User Manual V1.2 ESP8266 NodeMCU WiFi Devkit, n.d. Accessed on Feb. 11, 2020. [Online]. Available: https://www.handsontec.com/pdf_learn/esp8266-V10.pdf

[13] Mantech Electronics, YF-S201 datasheet, n.d. Accessed on Feb. 11, 2020. [Online]. Available: https://www.mantech.co.za/Datasheets/Products/YF-S201_SEA.pdf

[14] Rev Robotics, Analog Pressure Sensor Data Sheet, n.d. Accessed on Feb. 11, 2020. [Online]. Available: https://www.revrobotics.com/content/docs/REV-11-1107-DS.pdf

[15] Michael Henderson, What is NodeJS and Why You need to learn it, May 2019. Accessed on Feb. 12, 2020. [Online]. Available: https://medium.com/@michaelhenderson/what-is-nodejs-and-why-you-need-to-learn-it-f0760ba9a76a

[16] Heroku, The Heroku Platform, n.d. Accessed on Feb. 12, 2020. [Online]. Available: https://www.heroku.com/platform

[17] Chris Esplin, What is Firebase?, Oct. 2016. Accessed on Feb. 12, 2020. [Online]. Available: https://howtofirebase.com/what-is-firebase-fcb8614ba442