future internet

MDPI

Article

IoT Based Smart City Bus Stops

Miraal Kamal, Manal Atif, Hafsa Mujahid, Tamer Shanableh, A. R. Al-Ali * and Ahmad Al Nabulsi

Department of Computer Science and Engineering, American University of Sharjah, Sharjah P.O. Box 26666, UAE; g00058026@alumni.aus.edu (M.K.); g00057426@alumni.aus.edu (M.A.); g00058470@alumni.aus.edu (H.M.); tshanableh@aus.edu (T.S.); aalnabulsi@aus.edu (A.A.N.)

* Correspondence: aali@aus.edu

Received: 26 September 2019; Accepted: 24 October 2019; Published: 26 October 2019

Abstract: The advent of smart sensors, single system-on-chip computing devices, Internet of Things (IoT), and cloud computing is facilitating the design and development of smart devices and services. These include smart meters, smart street lightings, smart gas stations, smart parking lots, and smart bus stops. Countries in the Gulf region have hot and humid weather around 6–7 months of the year, which might lead to uncomfortable conditions for public commuters. Transportation authorities have made some major enhancements to existing bus stops by installing air-conditioning units, but without any remote monitoring and control features. This paper proposes a smart IoT-based environmentally - friendly enhanced design for existing bus stop services in the United Arab Emirates. The objective of the proposed design was to optimize energy consumption through estimating bus stop occupancy, remotely monitor air conditioning and lights, automatically report utility breakdowns, and measure the air pollution around the area. In order to accomplish this, bus stops will be equipped with a WiFi-Based standalone microcontroller connected to sensors and actuators. The microcontroller transmits the sensor readings to a real-time database hosted in the cloud and incorporates a mobile app that notifies operators or maintenance personnel in the case of abnormal readings or breakdowns. The mobile app encompasses a map interface enabling operators to remotely monitor the conditions of bus stops such as the temperature, humidity, estimated occupancy, and air pollution levels. In addition to presenting the system's architecture and detailed design, a system prototype is built to test and validate the proposed solution.

Keywords: Internet of Things; intelligent transportation systems; air pollution; energy efficiency

1. Introduction

This paper proposes an Internet of Things (IoT) based solution to improve the services of smart bus stop operations and maintenance in the United Arab Emirates (UAE). In recent years, the transport authority of Dubai has built around 100 air-conditioned smart bus-shelters [1]; however, the "smart" aspect of these bus stops does not provide a means of remotely monitoring bus stop utilities and malfunctions. Authorized operators periodically check the bus stop's status as well as ask bus stop users to report faulty air conditioners (ACs) by calling a helpline [2]. An ideal solution is to minimize response time by automatically detecting malfunctions and instantly reporting them to the authorities. Additionally, officials recognized the need for energy-efficient bus stops and planned to install solar panels to power them.

It would be ideal to utilize smart technology to contribute to this initiative of conserving energy by regulating the ACs and the lights' energy consumption based on real-time decisions made by analyzing the readings acquired from status and environmental monitoring sensors.

In an attempt to enhance existing bus stops, our proposed system's objective was to develop an efficient sensor array unit to collect, process, and distribute data related to the bus stop operational and

environmental parameters in order to operate the bus stop efficiently. The proposed system uses a mobile app that enables users to view current bus stop conditions on Google Map. An added benefit of the proposed system is that by keeping an estimated track of the number of people at the bus stop, the system can turn off the lights and AC during hours of vacancy, thus leading to significant energy conservation.

In the past, maintenance used to be considered difficult to conduct, however, with recent technological advances, it is no longer an issue [3]. An example of such an advancement includes the concept of IoT technology where people, processes, and things are connected [4]. Similarly, our proposed system reflects the aspect of 'things' by making use of sensors and actuators. The 'process' aspect of the proposed system is processing the generated and collected data in order to make appropriate decisions to command actuators accordingly. Whereas, the 'people' aspect refers to the maintenance personnel and the general public.

The rest of the paper is organized as follows. The literature review is presented in Section 2. Section 3 discusses the proposed hardware architecture. Section 4 discusses the proposed software architecture. Section 5 presents the implementation and testing of the prototype. The last section concludes the paper and presents ideas for future work.

2. Literature Review

A smart community architecture platform utilizing the IoT concept was proposed to provide neighborhood watch and pervasive healthcare as well as some additional added values such as smart metering management and social network applications [5]. Recently, IoT applications have been extended to smart city services. Smart transportation is a pillar in smart city applications. In the literature, an IoT-based school bus tracking with arrival time prediction was reported in [6]. The bus was equipped with a microcontroller, Radio Frequency Identification (RFID), and Global Positioning System (GPS) locators, Global System for Mobile (GSM) and General Packet Radio Services (GPRS) for communications. Another IoT-based school bus monitoring system was reported in [7], where the system integrated speed sensors, GSM/GPRS/GPS/RFID technologies, and a single chip microcontroller. The system had a lightweight machine-to-machine communication protocol that utilized the publish–subscribe message queuing telemetry transport (MQTT) connectivity protocol. Messages were used to allow parents, schools, and other authorized entities to track and locate the bus location and obtain an estimated arrival time to school or home. Another RFID smart application is a parking management system [8]. This system was designed to improve energy consumption and operational efficiency. A recent vehicle wireless monitoring and tracking system based on Bluetooth was deployed in fixed road stations along the road for the collection of data [9]. Vehicle start and end stations, real-time location, and delays were transmitted via the on-vehicle Bluetooth to the stations along the road. A traffic analysis was then performed in urban areas to predict the arrival time based on the real-time traffic flow.

In addition to monitoring and tracking, some transportation vehicles are utilized to monitor air pollution. In [10], a system that integrates a single-chip microcontroller, several air pollution sensors (CO, NO_2, SO_2), GPRS-Modem, and a GPSmodule was proposed. The system utilizes the wireless mobile public networks and integrates a mobile and wireless data acquisition unit that can be placed on top of any public transportation vehicle. The NO_2 and CO detection sensors provide good relative readings to determine air quality. In [11], a wireless sensor system was developed for real-time monitoring of toxic environmental volatile organic compounds. Another wireless sensor network system was developed in [12] to monitor indoor air quality, while in [13], an outdoor air pollution monitoring system was developed using ZigBee networks for ubiquitous-cities. This outdoor air pollution monitoring system assimilated a wireless sensor board that consisted of temperature, humidity, CO_2, and dust sensors. In [14] and [15], IoT-Bluetooth based vehicle tracking, road monitoring, and air pollution systems were proposed. The system's on–board controller collected and stored the

vehicle speed and location, exhaust pipe emission gas, and wind velocity. At a traffic light, the stored data are downloaded to a fixed computing platform via Bluetooth for further processing.

Vision-based systems have also been used in smart traffic applications, for instance [16,17] described camera-based vehicle collisions avoidance systems based on cameras and signal processing algorithms.

Most of the above systems are used to monitor and track buses and mobile air pollution using a sensing unit installed on top of the bus or along the road side to measure air pollutants. However, none of the above systems considered monitoring and enhancing the condition of the bus stops by using the recent advances in technologies. Recent attempts in smart cities, as evident in Dubai, U.A.E., have resulted in constructing smart bus stops equipped with air-conditioned shelters with comfortable seating arrangements and time schedules showing the approximate bus arrival time.

This proposed work presents an enhanced bus stop monitoring and operation system. It is designed to optimize energy consumption through estimating bus stop occupancy, remotely monitor air conditioning and lights, automatically report utility breakdowns, and measure the air pollution around the area. In order to accomplish this, bus stops will be equipped with Internet enabled standalone microcontrollers connected to sensors and actuators. The microcontrollers transmit sensor readings to a real-time database hosted in the cloud. A mobile app then notifies operators or maintenance personnel in the case of breakdowns. The mobile app encompasses a map interface enabling operators to remotely monitor the bus stop conditions such as the temperature, humidity, estimated occupancy, and air pollution levels in and around bus stops. In addition to presenting the system's architecture and detailed design, a system prototype was built to test and validate the proposed design.

It is worth highlighting that the major contribution and novelty of this system is that none of the existing works have monitored the status of bus stops in terms of the number of occupants, air conditioning, lighting, and air pollutant levels.

3. Proposed System Hardware Architecture

The objective of the proposed IoT-based bus stop was to capture the bus stop operational and surrounding environment parameters using a set of sensors that interfaced with a microcontroller. The microcontroller continuously reads and transmits the most recent sensor values to the remote database. In turn, a remote database pushes the most recent values to a mobile app available to the operators or maintenance personnel. Notifications are generated as soon as a utility abnormality or breakdown is detected at a bus stop.

For example, a temperature of above 23 degrees Celsius with an occupancy > 0 is considered "abnormal". CO levels above 70 ppm and LPG levels above 1000 ppm are considered abnormal. Additionally, a light is considered as "not working" if it is dark while there are people at the bus stop, and so forth.

In order to satisfy such requirements, the hardware architectural model chosen for the proposed IoT-based bus stop was a three-layered architecture as illustrated in Figure 1. The layers consist of the edge device layer, cloud layer, and the application layer.

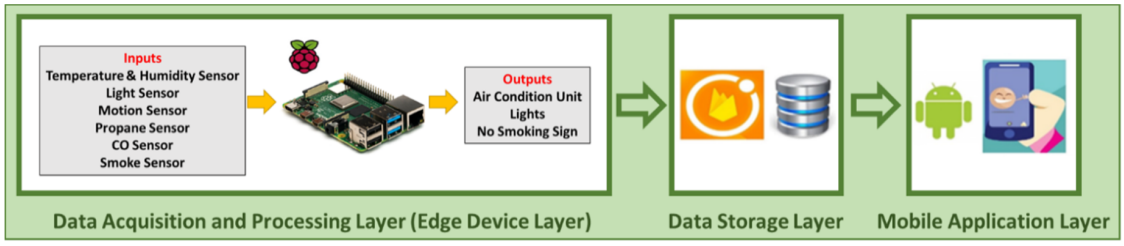


Figure 1. The hardware layers of the proposed system.

The edge device layer consists of a set of sensors, single board microcomputer with WiFi access point, and a set of relays. The sensors are temperature, humidity, light, motion, and air pollution sensors. The sensors communicate with the controller using the inter-integrated circuit communication protocol (I2C).

The relays are used to interface the lights, the air-conditioner, and non-smoking sign to the RPi output ports. Two different single-board edge device alternatives are compared in Table 1. The comparison is based on several criteria such as cost, wireless remote access, I/O pin availability, power consumption, built-in memory size, clock speed, weight, and size.

Table 1. Edge devices alternative comparison.

	Arduino		**RPi**		**Justification**
	Score	**WS** [1]	**Score**	**WS**	
Cost (15%)	8	120	6	90	Arduino UNO: 29$, RPi3: 35$
Memory (7%)	3	21	8	56	Arduino UNO: 2KB SRAM, RPi3: 1GB SRAM
Remote Access (15%)	3	45	10	150	Arduino UNO: Needs Extra WiFi Shield, RPi3: Available
I/O Pin Availability (16%)	5	80	10	160	Arduino UNO: 14 I/O, RPi3: 26 I/O
Power Consumption (18%)	7	126	3	54	Idle Arduino UNO: 232.5 mW, Idle RPi3: 1.9 W
Clock Speed (10%)	1	10	10	100	Arduino UNO: 16 MHz, RPi3: 1.2 GHz
Weight & Size (15%)	5	75	7	105	Arduino: Size and weight depends on added shields, RPi3: Credit card compact size microcomputer
Total Weighted Score		**477**		**715**	

[1] WS = Weighted Score.

In Table 1, the Cost factor was allocated as 15% as the implementation costs of large-scale systems of distributed smart bus stops must be kept to a minimum. The Memory factor, which refers to the static and dynamic RAMs of the microcontroller was allocated 7%. The Remote Access factor was allocated 15%. This factor refers to the Internet connection services that a microcontroller provides. The I/O Pin Availability factor was allocated 16% as the number of sensors and actuators that can be interfaced to the microcontroller board needs to be considered. A shortage of I/O pins at the microcontroller would restrict the application of the system. The Power Consumption factor was allocated 18% as the system aims to be environmentally friendly, and therefore, the power consumption must be kept to a minimum. The Clock Speed factor was allocated 10% as the microcontroller processing speed will affect the response time of the entire system. Finally, the Weight and Size factor was allocated 15% as portability needed to be considered.

Based on the comparison, the RPi outperformed Arduino in many aspects including the built-in wireless access point, speed, I/Os, and memory size. Additionally, the RPi can be configured as a standalone server and therefore in conclusion, the RPi was selected as the physical layer computing platform in this work.

The second layer is the cloud layer, which is used to store the collected bus stop status data. In this layer, the data are stored in JSON format in a Firebase real-time NoSQL database, which is cloud-hosted.

The third and last layer is the application layer. In this layer, the mobile phones of the operators and users are used to access physical edge devices through the cloud layer.

4. Proposed System Software Architecture

The system's software architecture is divided into three layers: data acquisition and processing layer, storage layer, and mobile app layer.

As shown in Figure 1, the data acquisition and processing layer is responsible for acquiring the real-time data from sensors that reflect the status of the bus stop, which is installed in a RPi at the bus stop. The RPi reads the status and values of a bus stop light sensor, temperature sensor, humidity sensor, motion sensor, smoke sensor, and air pollution sensor. The program was developed using

Python and is divided into several functions. At the start of the program, a function is called to configure the I/O ports of the RPi. The program then returns the temperature and humidity sensor values in °C and RH%, respectively. It communicates with the light sensor to read a value relative to the light intensity in the surroundings. It detects the motion of commuters entering and existing at a bus stop. It also outputs a value relative to the levels of smoke in the vicinity of the sensor and communicates with the air pollution sensors to read air pollutant levels.

The program reads light levels reported by the light sensor and compares them to a threshold value to determine if the indoor bus stop lights need to be turned on. It also compares the temperature to a threshold value to determine if the air conditioning needs to be adjusted. Finally, the program transmits data collected from all sensors to the remote database, which is the storage layer.

The flowchart of the data acquisition and processing layer is shown in Figure 2.

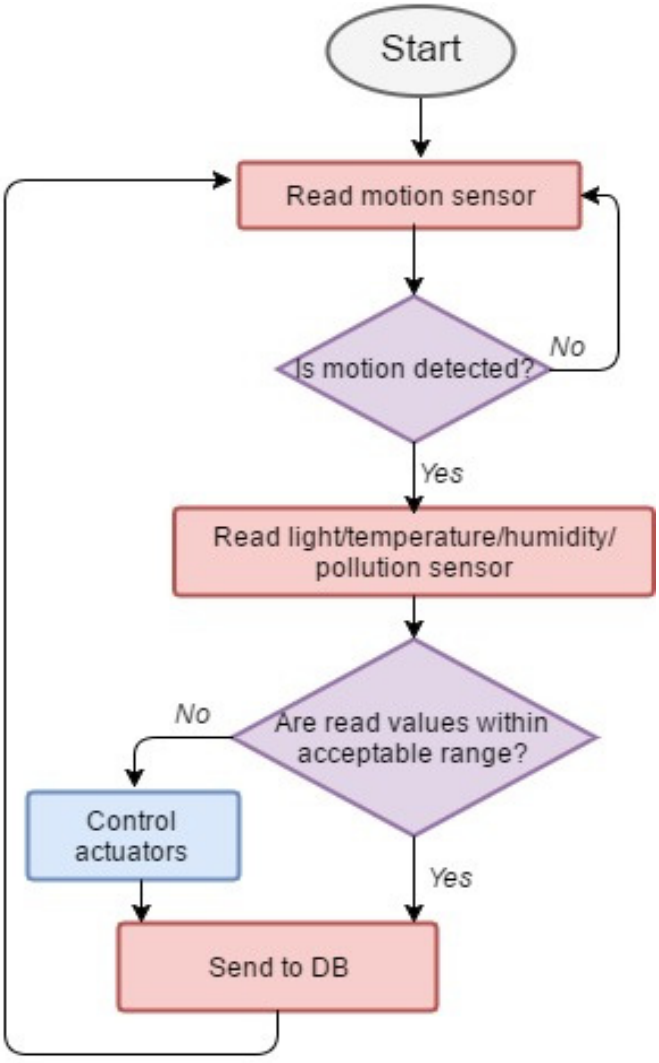


Figure 2. Flowchart of the control and processing layer.

The second software layer is the data storage layer. This layer consists of a cloud hosted remote database called Firebase. The Firebase platform provides a real-time database that we used to store the bus stop sensor data. Data are stored in JSON format in the Firebase real-time NoSQL database. When Firebase detects a change or insertion of new data in the database, it triggers an event that pushes the most recent values to the third layer, which is the mobile app layer.

The third and final layer is the mobile app layer, which is the top most layer that the end-user interacts with. It is responsible for enabling the visualization of information to the mobile app users (i.e., the bus stop maintenance personnel) and runs a background service for notifications. In addition to Firebase, the mobile app uses an SQLite database local to the device to store daily and monthly averages of temperature and pollution levels. This allows the app users to view the statistics for each bus stop. The software of this layer was developed using the Android Studio IDE.

The app's implementation consists of Graphical User Interfaces (GUI) for signing in, viewing the map, and for viewing the bus stop status and statistical graphs. The app runs a background service to receive sensor readings from Firebase and to send notifications to the app users.

The mobile app is not concerned with the processing logic used at the data acquisition and processing layer. This provides for system scalability, where the processing can happen on distributed nodes, then pass the information to the remote database. Consequently, the app can communicate with the database to retrieve the desired values. The summary of the averages of the daily parameters for bus stops are saved locally on the operators' mobile devices for faster access to the daily statistics. This is to avoid storing history on the Firebase real-time database due to the costs associated with large amounts of data storage. Having a separate layer for data storage can also allow for future scaling of the system, where the database is implemented in a distributed architecture, allowing for the storage of big data generated by the proposed IoT system.

The proposed system layers communicate with each other using interfaces. The data acquisition and processing layer communicates using an interface between Python and Firebase. It enables values to be posted directly to the Firebase real-time database. The application layer is interfaced to Firebase through the Firebase database interface. This allows the end-users to view current bus stop conditions. The deployment diagram of the system is shown in Figure 3.

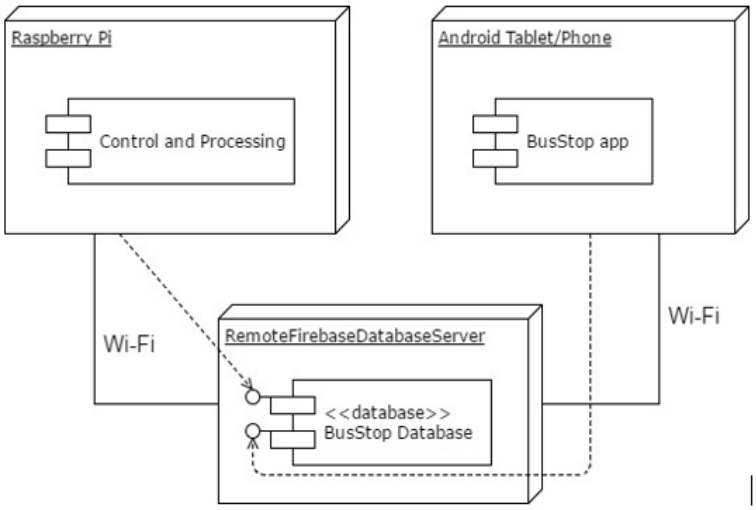


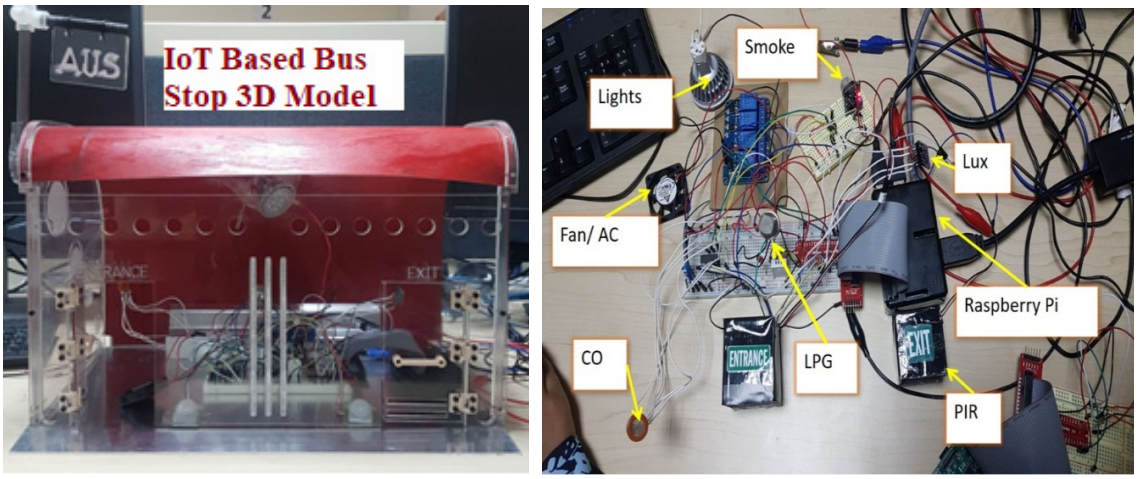
Figure 3. System deployment diagram.

5. System Prototype Implementation and Testing

In light of the aforementioned hardware and software architectures, the objective of the design proposed was to optimize energy consumption through estimating the bus stop occupancy, remotely monitoring, and automatically controlling the air conditioning and lights. The system also reports utility breakdowns and measures the air pollution around the area.

As such, the outcome should be a standalone remote monitoring and control unit that can be installed in existing bus stops to enhance their operation in terms of efficient energy consumption and user satisfaction.

To validate the proposed system's hardware and software architectures, a prototype was built and tested. The prototype is shown in Figure 4.



(**a**) (**b**)

Figure 4. (**a**) IoT based bus stop 3D model; (**b**) System hardware prototype.

The sensors used were DHT22 for temperature and humidity, TSL2561 for light, MQ7 for carbon monoxide, MQ6 for LPG, and MQ135 for smoke detection. Table 2 shows the specifications of the used sensors.

Table 2. Sensor specifications.

	Humidity	**Temp**	**CO**	**Smoke**	**LPG**
Sensor Name	DHT22	DHT22	MQ7	MQ135	MQ6
Operating Range	0–100% RH	−40–80 °C	20–2000 ppm	10–300 ppm	200–10000 ppm
Sensor Resolution	0.1% RH	0.1 °C	< 0.5 ppm	< 0.6 ppm	< 0.6 ppm

Verification of our system included tests that enabled us to check if the code and procedures executed by the hardware and software were correct. Test cases were developed based on certain use cases and tested to verify if they matched the expected output. According to the proposed system's requirements, the marker attribute on the app's map must turn green if all the following conditions are met: the bus stop is not empty, the temperature is less than or equal to 23 °C, the LPG is less than or equal to 65 ppm, the CO is less than or equal to 39 ppm, and the lights are working.

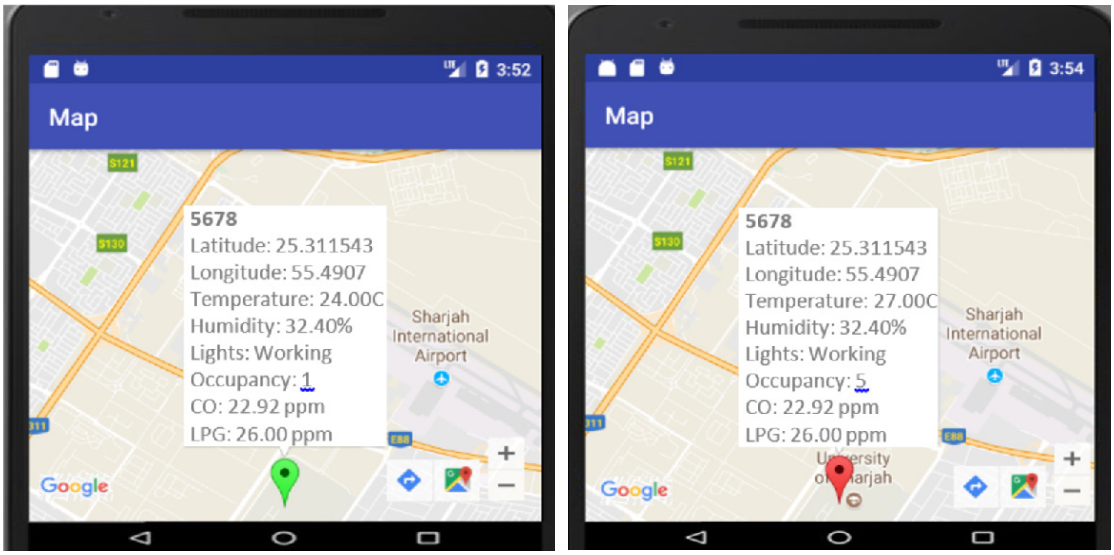
The marker attribute on the application's map must turn red if any of the following conditions are met. The bus stop is not empty, the temperature is greater than 23 °C, the LPG is greater than 65 ppm, the CO greater than 39 ppm, or the lights are not working.

The thresholds used for CO and LPG were reduced for testing purposes as it was not possible to obtain hazardous CO and LPG levels.

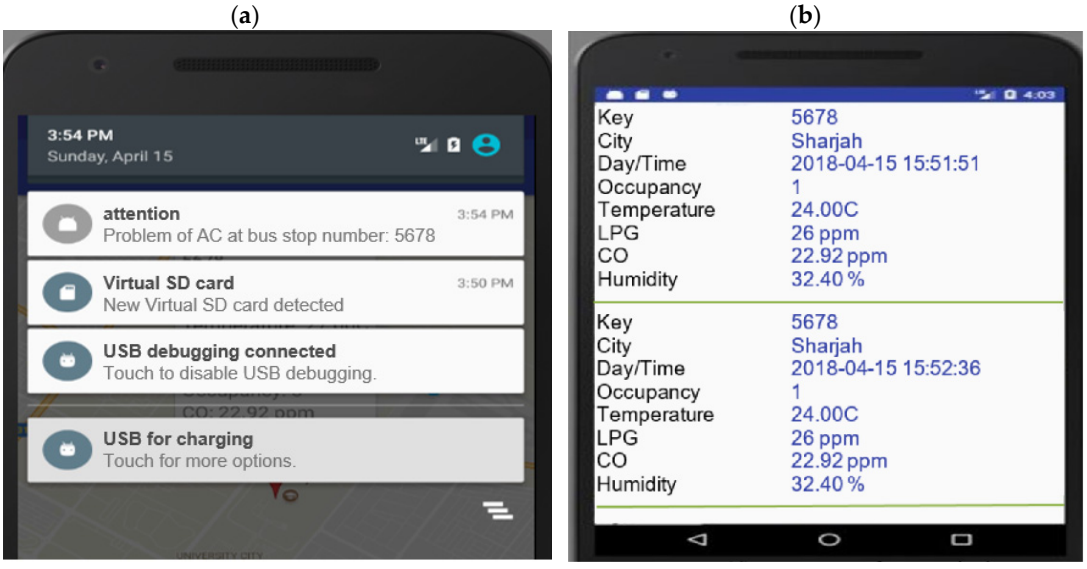
The test conditions above were used to check the correctness of the hardware for controlling the lights and the air-conditioning and report them as broken if needed. This was achieved by adjusting the temperature, occupancy, LPG, and light levels. The actual outcomes were the same as the expected output for all of the above test cases. For example, in test case 3, the AC was turned ON as expected because the bus stop occupancy was 1 and the temperature was 24 °C (map indicator was green, as shown in Figure 5a). On the other hand, in test case 6, the AC did not work although the temperature was 27 °C with an occupancy of 5 (map indicator was red, as shown in Figure 5b). This indicates that the AC was malfunctioning, which confirms that the system was able to detect such abnormality. More test cases results are tabulated in Table 3.

Table 3. Test cases.

Sr. No.	Test Case	Excepted Condition at the Bus Stop Site
1.	Occupancy (6): Temperature (23 °C)	AC turned OFF.
2.	Occupancy (0):	AC turned OFF.
3.	Occupancy (1): Temperature (24 °C)	Turn ON AC.
4.	Occupancy (1): Light Level 5	Turn ON lights.
5.	Occupancy (1): Light Level 1	Lights not working.
6.	Occupancy (5): Temperature (27 °C)	AC not working.
7.	Occupancy 3, LPG 72 ppm	Gas level high at bus stop.
8.	Occupancy 0, LPG 72 ppm	Gas level high at bus stop.
9.	Occupancy 1, CO 40 ppm	CO level high at bus stop.
10.	Occupancy 0, CO 40 ppm	CO level high at bus stop.

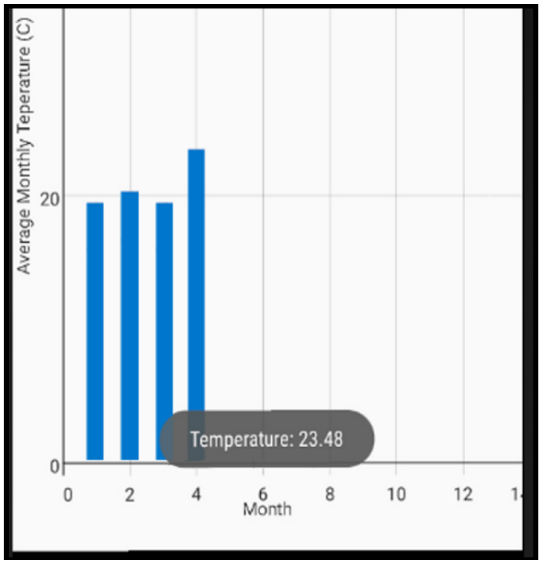


(**a**) (**b**)



(**c**) (**d**)

Figure 5. *Cont.*



(**e**)

Figure 5. (**a**) Normal bus stop conditions. (**b**) Abnormal bus stop conditions. (**c**) Notification of Air Condition (AC) not working. (**d**) Day status for a particular stop; (**e**) Example of monthly average sensor readings.

Figure 6 shows the transmitted data frame format that contains the bus stop real-time status. Figure 5 shows some screenshots from the mobile app.

Bus Stop ID 5678	Latitude (°)	Longitude (°)	Temperature (°C)	Humidity (%RH)
	25.311543	55.4907	24.00	32.40
	Light	Occupancy	CO (PPM)	LPG (PPM)
	Working	1	22.92	26.00

Figure 6. Bus stop data format.

The scalability of the system was tested by simulating over 90 bus stops and adding them to the Firebase database, as shown in Figure 7. The app was able to correctly show the bus stop conditions and had a quick response time of less than three seconds in reporting any malfunctioning utilities at the bus stop. It is worth noting that we were limited in this testing as the free Firebase Realtime Database plan used in our prototype is limited to 100 database instances.

The system hardware cost is $100 for the Raspberry Pi and Sensors. It was assumed that the central server could host up to 50 bus stops. The server cost is about $1500 and can be divided between the 50 bus stops (i.e., $30 per bus top, which added up to $130 per bus stop). The Internet service provider charge is about $20 per month.

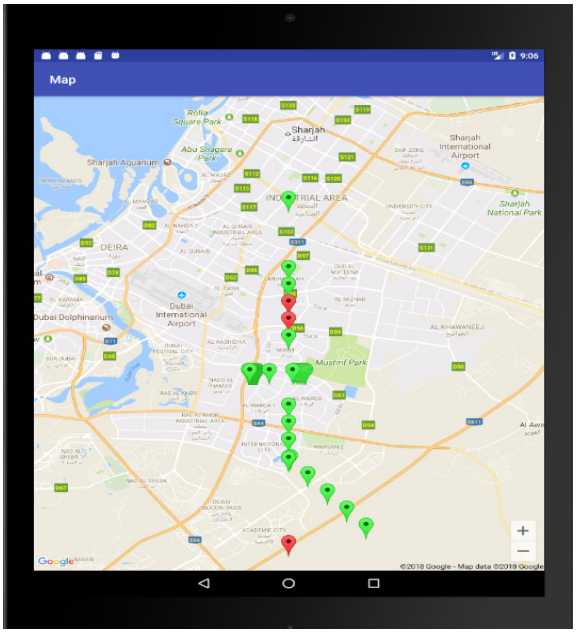


Figure 7. Testing scalability: around 99 bus stops.

6. Conclusions

This paper introduced a novel IoT-based bus stop that provided smart monitoring and maintenance solutions to reduce energy consumption and increase the satisfaction of commuters. The system consisted of layers corresponding to data acquisition, processing, storage, and presentation. The system monitored the bus stop's occupancy and sensors so that based on the sensor readings, the lights and air-conditioning could operate more efficiently. The air pollution in and around the bus stop vicinity was also monitored. The bus stop operation status and air pollution levels were then sent to a cloud-based server. A mobile app was developed to enable operation engineers to monitor the air conditioning and lights remotely. Utilizing Google Maps, a bus stop operation status was displayed using different colored attributes. In order to meet the system objectives, test cases were conducted to ensure that the system performance was aligned with the goals. The results were conclusive to determine that this project can be scaled to multiple bus stops. The proposed system cuts down on power consumption by controlling the bus stop's utilities based on its occupancy. By keeping constant track of the bus stop's occupancy, the system can control the lights and the air conditioning. This helps save energy, as compared to traditional bus stops where the utilities remain running even during hours of no occupancy, leading to a waste of resources. The exact amount of saved energy will be reported in future work as we plan to add a solar energy system to supply power to the bus stop. Another aspect than can be addressed in the future is the communication between the vehicle and the infrastructure (Bus Stop). Additionally, the versatility of the system can also expand its application to various settings such as schools, weather stations, hospitals, and the like.

Author Contributions: Conceptualization, M.K., M.A., H.M., T.S., A.R.A.-A., and A.A.N.; Methodology, M.K., M.A., H.M., T.S., A.R.A.-A., and A.A.N.; Software, M.K., M.A., and H.M.; Validation, X M.K., M.A., and H.M.; Formal analysis, M.K., M.A., and H.M.; Investigation, M.K., M.A., and H.M.; Resources, M.K., M.A., H.M., and A.A.N.; Data curation, M.K., M.A., and H.M.; Writing—original draft preparation, M.K., M.A., and H.M.; Writing—review and editing, M.K., M.A., H.M., T.S., A.R.A.-A., and A.A.N.; Visualization, M.K., M.A., and H.M.; Supervision, T.S., A.R.A.-A., and A.A.N.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Reporter, S. Dubai Gets 100 Air-Conditioned Smart Bus Shelters. Khaleejtimes.com. 2017. Available online: https://www.khaleejtimes.com/rta-constructs-100-smart-ac-bus-shelters (accessed on 20 September 2019).
2. 100 AC Bus Shelters Not Working in Dubai. Emirates 24|7. 2017. Available online: http://www.emirates247.com/news/emirates/100-ac-bus-shelters-not-working-in-dubai-2012-07-09-1.466390 (accessed on 20 September 2019).
3. Alexandru, A.M.; Fiasché, M.; Pinna, C.; Taisch, M.; Fasanotti, L.; Grasseni, P. Building a smart maintenance architecture using smart devices: A web 2.0 based approach. In Proceedings of the 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow (RTSI), Bologna, Italy, 7–9 September 2016; pp. 1–6.
4. Islam, T.; Mukhopadhyay, S.C.; Suryadevara, N.K. Smart sensors and internet of things: A postgraduate paper. *IEEE Sens. J.* **2017**, *17*, 577–584. [CrossRef]
5. Li, X.; Lu, R.; Liang, X.; Shen, X.; Chen, J.; Lin, X. Smart community: An internet of things application. *IEEE Commun. Mag.* **2011**, *49*, 68–75. [CrossRef]
6. Jisha, R.C.; Jyothindranath, A.; Kumary, L.S. IoT based school bus tracking and arrival time prediction. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics, Udupi, India, 13–16 September 2017; pp. 509–514.
7. Zambada, J.; Quintero, R.; Isijara, R.; Galeana, R.; Santillan, L. An IoT based scholar bus monitoring system. In Proceedings of the IEEE First International Smart Cities Conference, Guadalajara, Mexico, 25–28 October 2015; pp. 1–6.
8. Tsiropoulou, E.; Baras, J.S.; Papavassiliou, S.; Sinha, S. RFID-based smart parking management system. *Cyber Phys. Syst.* **2017**, *3*, 22–41. [CrossRef]
9. Fernández-Ares, A.J.; Mora, A.M.; Odeh, S.M.; García-Sánchez, P.; Arenas, M.G. Wireless monitoring and tracking system for vehicles: A study case in an urban scenario. *Simul. Model. Pract. Theory* **2017**, *73*, 22–42. [CrossRef]
10. Al-Ali, A.R.; Zualkernan, I.; Aloul, F. A mobile GPRS-sensors array for air pollution monitoring. *IEEE Sens. J.* **2010**, *10*, 1666–1671. [CrossRef]
11. Tsow, F.; Forzani, E.; Rai, A.; Wang, R.; Tsui, R.; Mastroianni, S.; Knobbe, C.; Gandolfi, A.J.; Tao, N.J. A wearable and wireless sensor system for real-time monitoring of toxic environmental volatile organic compounds. *IEEE Sens. J.* **2009**, *9*, 1734–1740. [CrossRef]
12. Chung, W.; Yang, C.H. Remote monitoring system with wireless sensors module for room environment. *Sens. Actuators B* **2009**, *113*, 35–42. [CrossRef]
13. Kwon, J.W.; Park, Y.M.; Koo, S.J.; Kim, H. Design of air pollution monitoring system using ZigBee networks for ubiquitous - city. In Proceedings of the 2007 International Conference on Convergence Information Technology, Gyeongju, Korea, 21–23 November 2007; pp. 1024–1031.
14. Jagasia, A.; Advani, S.; Prakash, A.; Kulkarni, C.; Ghadge, V.; Shete, A. IoT based vehicle monitoring system using bluetooth technology. *Int. J. Innov. Res. Sci. Eng. Technol.* **2017**, *6*, 1–8.
15. Lewandowski, M.; Płaczek, B.; Bernas, M.; Szymała, P. Road traffic monitoring system based on mobile devices and bluetooth low energy beacons. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 3251598. [CrossRef]
16. Agarwal, Y.; Jain, K.; Karabasoglu, O. Smart vehicle monitoring and assistance using cloud computing in vehicular Ad Hoc networks. *Int. J. Transp. Sci. Technol.* **2018**, *7*, 60–73. [CrossRef]
17. Jain, N.K.; Saini, R.K.; Mittal, P. A review on traffic monitoring system techniques. In *Soft Computing: Theories and Applications*; Ray, K., Sharma, T., Rawat, S., Saini, R., Bandyopadhyay, A., Eds.; Springer: Singapore, 2019; Volume 742.