Android Application Based Gas Leakage Notifier

Aishwarya R[1], Akshath R Hegde[1], Ashwini N[1], Mohammed Thavaf A R[1], Anitha C[2]

[1] *UG Student, Department of Computer Science and Engineering, The National Institute of Engineering, Karnataka, India*
[2] *Assistant Professor, Department of Computer Science and Engineering, The National Institute of Engineering, Karnataka, India*

ABSTRACT

Gas leakage is a phenomenon which can possibly dispense significant harms to lives and property. Detection of gas leakage and informing the concerned individuals is frequently the first and the most imperative stride in keeping away from any casualties. Already accessible products in the market neglect to convey the notice to clients who are situated elsewhere, and, offer next to zero flexibility. The best answer for this is to use the capacity of the idea called "Internet of Things", which means whereby remote sensors, cloud storage, and convenient terminal gadgets can be utilized to alarm the client with respect to a gas leakage. This paper introduces such a method, wherein the client is kept mindful of the leakage status through an Android application and cellular network based SMS. The proposed solution is highly flexible and cost effective.

Keyword : - *Gas leakage , Internet of Things, Android application, Raspberry PI and MQ6 sensor*

---

1. INTRODUCTION

Gas leakage refers to a situation wherein potentially dangerous gases are leaked from source and spread out in the environment or some limited space such as home. This can lead to deaths and destruction of properties based on the nature of the leaked gas in question. Some gases are toxic while some are highly inflammable. Gases like, Hydrogen Sulphide inhibits oxygen exchange on the cellular level; Carbon monoxide inhibits the body’s ability to transport oxygen to all parts of the body. Such gases come under the category of toxic gases. Liquid Petroleum Gas (LPG) used for domestic cooking purpose is a mixture of two gases: propane and butane is highly flammable. In this paper we focus on the detection of leakage of LPG.

The only recourse in these situations is to stop the leakage and move out people into safe zones. But that is possible only when the leakage is detected. Most of the times people who are in the vicinity of the gas leakage, are not even aware of the leak in the first place. Many gases are odorless and are poisonous too (Carbon monoxide, for example). If such gases are spread out in the environment, they can’t be detected by smell. The most common situation is leakage of combustible cooking gas (LPG) at home. These gases are added with odor for the sole purpose of enabling detection by leakage. But if the owner of the house is away during leakage, chances of damage are high. The whole scenario is even more challenging for people with disabilities. They might not be able to notice any leakages, thus putting their lives under greater peril.

Effective actions can be taken if the concerned people are informed immediately in case of leakage. With the latest available technology, doing so is even the easier. This paper proposes a system that notifies the user of gas leakage and contains Sensors that can detect presence of gases and gateway devices like Raspberry Pi. This is combined with the omnipresence of portable terminal devices that are able to spread the information quickly.

2. EXISTING SOLUTIONS

There a couple of solutions proposed for notifying the gas leakage that is considered in this paper. In [1], a solution is discussed where the users are alerted by audio-visual indications. The system proposed in [1] is a typical hardware system which comprises of an MQ-3 sensor, a speaker alarm and an LED light. Whenever there is a gas leakage and if the concentration exceeds 1000 ppm (parts per million), it is detected by the MQ-3 sensor. The sensor's output is connected to the speaker alarm and the LED. This approach has three drawbacks. Firstly, it assumes that the user always present nearby when the leakage occurs. If the user is away from his home, then the system fails to make an impact. Secondly, it does not provide a facility for specially-abled people. They might not be able to hear the alarm or see the LED switched on. And finally such a system is neither portable nor finds an extension.

In [2], a solution is proposed for vehicle tracking with gas leakage and temperature detection. It's an IOT application, which monitors and tracks the vehicle. The system proposed in [2] comprises of an Embedded Linux board (Raspberry - Pi) with GPS and GSM modules for vehicle tracking, MQ-6 sensor for gas leakage detection and DS18B20 as temperature sensor. The system alerts the user through an SMS on gas leakage. This system has some shortcomings. It virtually gives no control to the user to take action after alerting of leakage. The user cannot obtain the real time readings. Also the numbers to which the SMS is to be sent is hardcoded and can't be changed or added. The system also uses the digital output of the sensor. So, fine control over the threshold value is not possible. To summarize, no modifications or up gradations are possible once the system is deployed.

3. OVERVIEW OF PROPOSED SOLUTION

In this section we briefly summarize the proposed solution – 'Android application based Gas Leakage Notifier (AAGLN)'.

The proposed solution makes use of sensors, gateway devices, lightweight messaging protocols and portable terminal devices to alert the user about the gas leakage. The solution is very generic and is applicable regardless of the nature of the gas or the site of deployment of sensor. The premises in which the proposed system is tested, is a confined room with an LPG gas source. The solution also assumes the availability of a Wi-Fi connection in the vicinity of the gas source (usually home).

First, an area that has the potential to be affected by the leakage is identified within the house. The sensor is deployed at a suitable place depending on the user's sensitivity. The sensor is connected to the Raspberry Pi (Rasp.Pi) [3, 9] device by wired means. The Rasp.Pi is connected through Wi-Fi to the user. It is responsible for sending application (app) alerts and SMSes.

Every user is given a unique ID. The user has to install our Android app and register with the system using this unique ID. The user will have an option to enter as many as 4 phone numbers to which the alerts will be sent. It is preferable for the user to have a data connection as the alerts can be sent through the app, but it is not mandatory. The system makes sure that the user and the contacts referred are alerted even in the absence of a data connection by sending SMS alerts. However, it is assumed that the user is connected to a cellular network at all the times.

4. IMPLEMENTATION

In this section, the implementation of the proposed solution is discussed in the following sequence. First, the circuit details are discussed followed by the description of the app's working and sending SMS alerts. Next, server-related details are provided. Finally, the functioning of the android application is discussed.

Fig -1: System Architecture of AAGLN.

Fig. 1 shows the 3–tier client-server system architecture of AAGLN in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate tiers.

Fig. 2 and 3 show the different users interacting with the proposed AAGLN system. The functionalities provided to the users can be broadly categorized into Login/Sign Up function and Leakage Detection function.

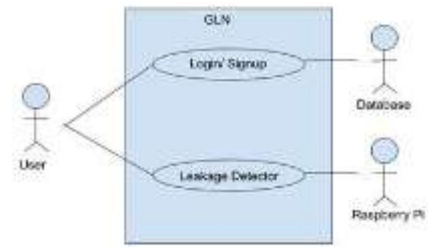


Fig -2: Use Case Diagram Level – 1.

The action taken after Leakage Detection functionality can be further categorized based on the type of notification as either SMS notification or APP notification (as depicted in Fig. 3). The corresponding logic layer intermediate servers involved for notification is also depicted.

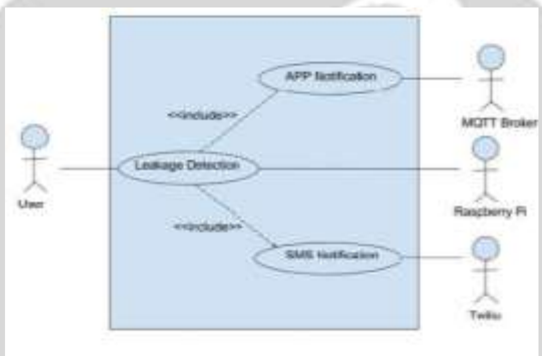


Fig -3: Use Case Diagram Level – 2.

4.1 Circuit Connections

The solution has been built and tested for leakage of LPG (fig. 4). The sensor used for detecting gas leakage is MQ - 6.The system has been tested on the MQ - 6 sensor module manufactured by eProLabs [3]. The sensor requires a power of 5V. In the absence of LPG or the individual gases (propane and butane) that constitute the LPG, the resistance of the surface of the sensor will be high which limits the analog output voltage. Upon exposure to these gases, the resistance falls and the analog output voltage increases accordingly. The analog output voltage varies between 0V and 5V where values near to 0V indicate absence of leakage and values close to or equal to 5V indicate widespread leakage. More information about how the surface is made or the inner circuit details can be obtained in the data sheets for this sensor [3].

The sensor has 4 pins. One of them, indicated as Vcc is the input voltage, equal to 5V. Another pin, indicated as GND, must be connected to the ground of the Rasp.Pi. The remaining 2 pins are output pins, one analog and one digital. As mentioned earlier the analog output pin varies from 0V to 5V.The digital pin outputs either a high or a low. The threshold for this pin can be set by rotating the knob on the back of the sensor.

The system has been tested by using the analog output pin only as the analog pin gives better precision as it indicates the exact voltage value. On the other hand, the digital pin is capable of outputting either a high or a low, based on a threshold value. This leads to loss of precision.

The system uses Raspberry Pi 3 Model B (Rasp.Pi) as the gateway device. This model of Pi comes with a built-in microprocessor, a Wi-Fi module etc. The Rasp.Pi has 40 General Purpose Input Output (GPIO) pins which allow interfacing with sensors and other devices. However, all these are digital pins. So the output of the sensors must be passed through an Analog to Digital Converter (ADC) before connecting to the Rasp.Pi. In addition, the Rasp.Pi's

input pins can accept a maximum of 3.3V. Thus a voltage divider circuit must be used to step down the voltage level ranging (0 to 5V) to (0 to 3.3V) [10].
The system has been tested with the MCP3008 ADC from ADAFRUIT [4]. The ADC is capable of working with 8 simultaneous input channels. The stepped down analog voltage from the sensor is connected to one of the 8 input channels.
The Vdd pin is the power supply for the ADC and is connected to 3.3V pin of the Rasp.Pi. The reference voltage pin Vref of the ADC must be connected to 3.3V from Rasp.Pi. AGND and DGND refer to analog and digital ground respectively and must be connected to the ground pin of Rasp.Pi. The remaining pins can be connected to any 4 GPIO pins of Rasp.Pi when using software Serial Peripheral Interface [4].

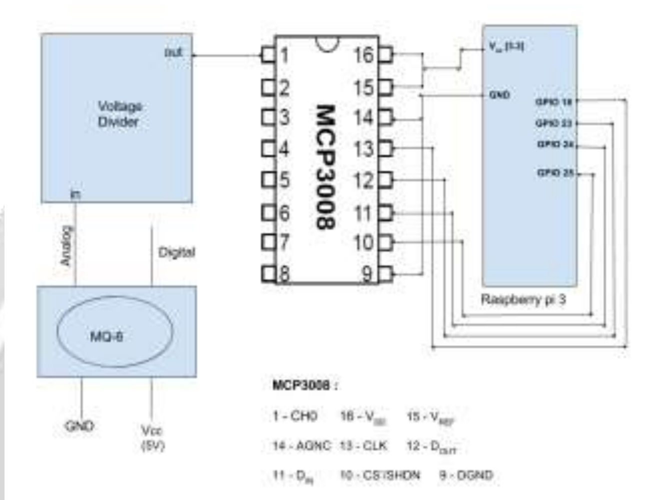


Fig -4: Circuit Connections for AAGLN

4.2 Sending Alerts and SMSes

The Rasp.Pi deployed to the user contains a program that is responsible for reading the values from the ADC (range 0 to 1023) and also sending the alerts and SMS. But, the MQ-6 sensor works in the range of 200 to 10000. This required a mapping of corresponding values in the ADC range. For this purpose, a simple algorithm was used [3]. Let us consider two ranges:

$$X[X_{min}, X_{max}] \text{ and } Y[Y_{min}, Y_{max}]$$

Given X_n *corresponding* Y_n *needs to be found*

$$X_{percent} = \frac{X_n - X_{min}}{X_{max} - X_{min}} \quad (1)$$

$$Y_n = X_{percent} * (Y_{max} - Y_{min}) \quad (2)$$

According to eProLabs datasheet [3], as per the standard detection condition range, Y=[200,10000] and X=[0,1024].
From equation (1) and (2),

$$X_{percent} = \frac{X_n}{1024} \quad (3)$$

$$Y_n = X_{percent} * 9800 + 200 \quad (4)$$

In order to detect and alert for gas leakage, a threshold value had to be hard coded into the program depending on the user's sensitivity requirements. The system was tested with the threshold of 500 as it is reasonable. If the value reaches this threshold then the app sends a notification to the user and also an SMS is sent to the user himself or to the primary number specified. Then the system waits for a specific amount of time, here considered as 6 seconds. If the measured value is again above the threshold, then SMSes are sent to the user's secondary contacts. This might be numbers of fire stations/agencies or his neighbours, for example. This feature is provided so that the user could trust someone with the task of handling the situation when he is unable to do so. The algorithm repeats if the leakage has not subsided.

Note that, regardless of whether the value is above or below the threshold, the readings can always be obtained in the app. The numbers' and user's details are stored on a server maintained by us, the developers. The Rasp.Pi fetches the numbers from the server every time the SMS is to be sent. Fig. 5 shows the details involved in the proposed algorithm for gas leakage detection and notification.

4.3 The server

The system will have a server maintained by the developers which is responsible for storing the user credentials [8]. The testing was done on a PC and the program was implemented in Python [6]. The server maintains 2 tables. One of them contains the set of valid unique IDs. Whenever a new user is registered, to this set, a new ID is added. The other table consists of user credentials such as username, password and the primary and secondary phone numbers. A new entry is added to this table when a user successfully registers on the Android App [5, 7]. However, the user can update the credentials anytime. The server interacts with both the Rasp.Pi and the Android App. The Rasp.Pi fetches the numbers from the server. The Android App fetches the unique ID from the server in the event of a login.

Algorithm Name : Notification and SMS
Input : Sensor Readings
Steps : 1.state_is_ok = true state_is_emergency = false last_time = 0 2.loop: 2.1 val=read_value() 2.2 if val>**threshold** and state_is_ok notify() send_sms_primary() state_is_ok = false state_is_emergency = true last_time = get_currenttime()

2.3 else if val>**threshold** and state_is_emergency = true and get_currenttime() - last_time > **T** send_sms_all() state_is_ok = true state_is_emergency = false last_time = 0
Output : Notification is sent to user SMS is sent to user

Fig -5: Notification and SMS Algorithm.

4.4 The Android Application

A user of the system is assisted with an Android App. This app allows the user to
- Register his product
- Add and modify phone numbers to which the SMSes will be sent
- View readings from the sensor in real time
- Get notifications upon leakage

The App has a user-friendly and a simple interface. Upon opening the App, the user is directed to login/Sign Up page. A first time user must register his product by mentioning the unique ID given to him. He can set the username and password and enter up to 4 phone numbers which will receive the SMSes. Then he is redirected to a page where he can choose among one of these options: view readings, modify phone numbers and logout. Upon selecting "view readings", the readings from the sensor in real-time digital format are presented to the user. Upon selecting "modify", he will be allowed to change the user credentials.

In case of a gas leakage, the user will be alerted by the App notification and vibration. The system will also send an SMS, to be twice sure that the information reaches the user.

5. RESULTS

The proposed AAGLN system was tested in real time and the results obtained are depicted as screen shots in this section.

When there is a gas leakage detected, the system simultaneously sends SMSes to the user and all secondary contacts and also alerts the user through the Android App (Fig. 6 to 8).

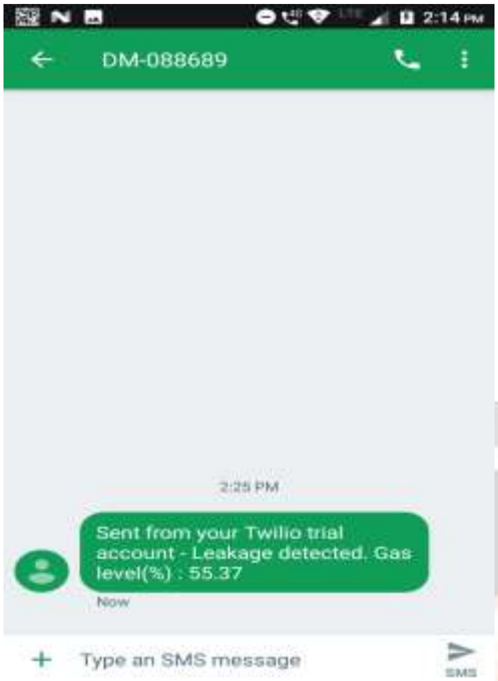
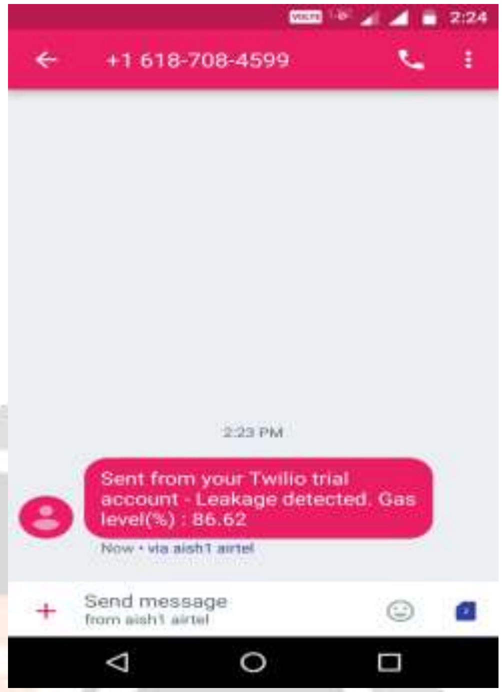


Fig -6: SMS Notification to the user (primary contact)



.**Fig -7**: SMS Notification to first secondary contact.

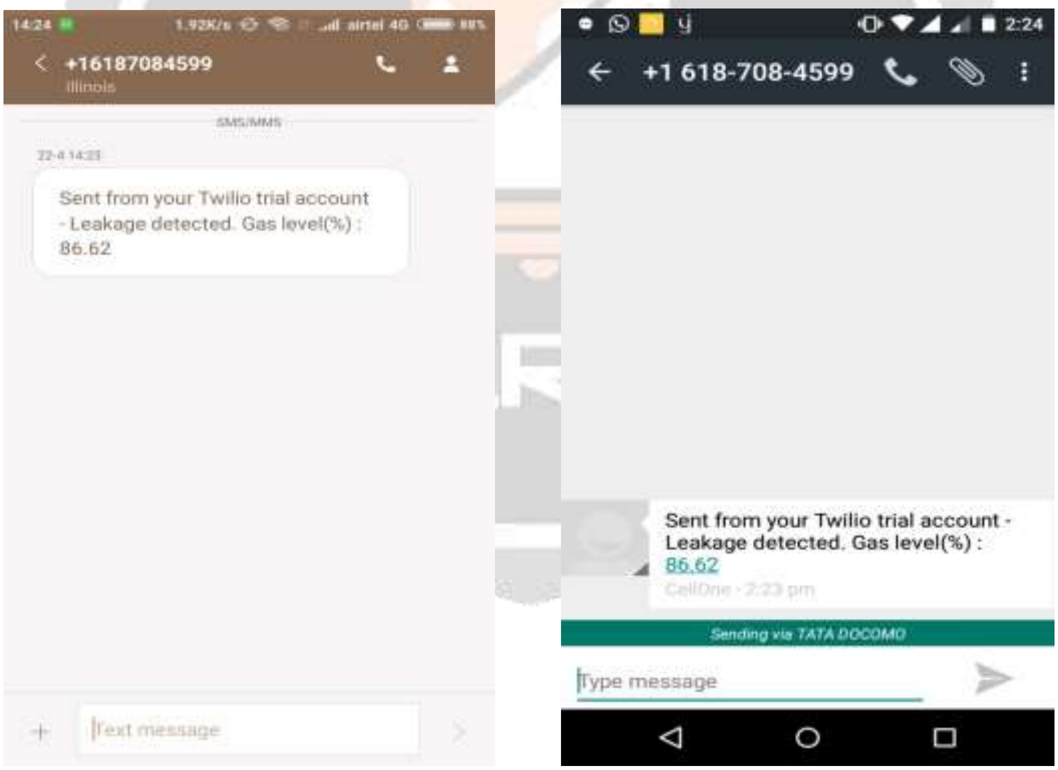


Fig -8: SMS Notification to the remaining secondary contacts.

As soon as the user logs into the Android App, sensor's readings will be displayed (0 to 1023). A simple color coding is used so that the non-leakage readings can be differentiated from the leakage readings. Figures 9 and 10 show readings displayed by the Android App. The threshold value was set for 500 for testing purpose.

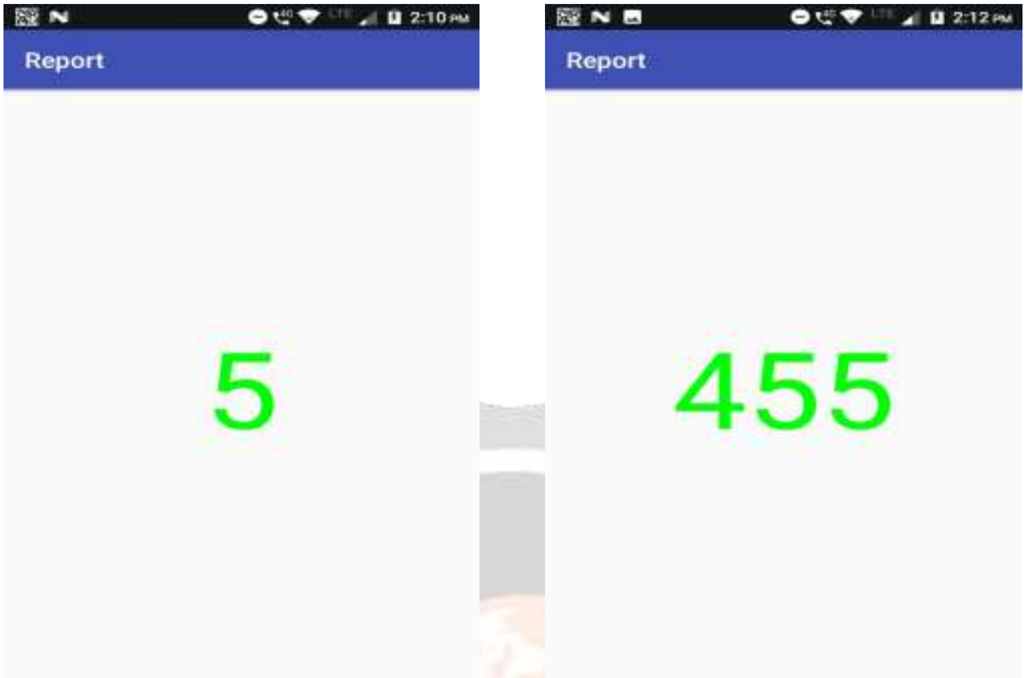


Fig -9: App Notifications indicating no leakage.

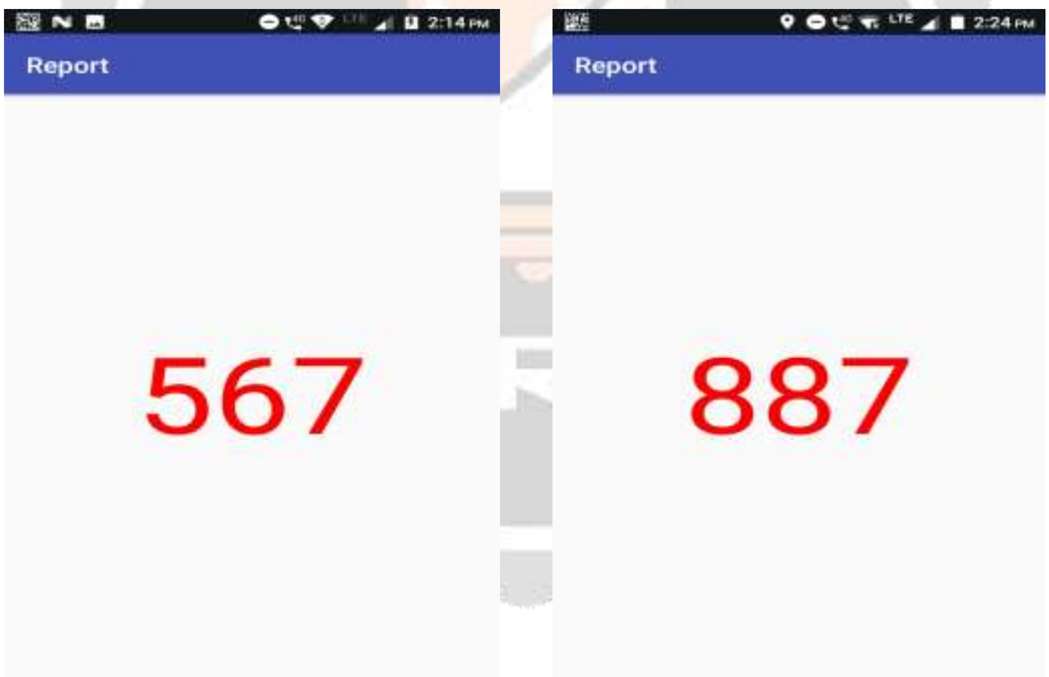


Fig -10: App notifications indicating gas leakage.

6. CONCLUSION

The proposed system provides solutions to the questions raised in the existing systems [1, 2]. It has all the functionality of these systems. Our system alerts the user wherever he is, through both notifications with vibration and SMS. No additional alarming devices required. This addresses the drawback of [1]. It allows users to customize the system as well as allow future enhancements .This is made possible by taking input from the user through the terminal device application and having a server that acts as the bridge between the Rasp.Pi and the Android App. This solves the drawback of the second existing system [2].

The system can be very helpful in this modern life characterized by its busy pace. It is so very common that people have less and less time to attend to household details. But we must remember that leakage of gas is not some trivial issue. We have made use of the current technologies to create a solution that mitigates the issue to a large extent.

5. FUTURE ENHANCEMENTS

The proposed system focuses on alerting the user about the LPG gas leakage. But the system can be upgraded to actively stop the leakage. This is possible by making Rasp. Pi to activate a motor, for example, to turn off the gas source knob. This can be done without hampering other functionalities of the current system. The system can be equipped with cameras so that photos of the site can be sent to the user during gas leakage. In case of multiple gas sources, the photos sent to the user will give a clarification to the user regarding which among the sources is responsible for the leakage. The system can be added with smart water sprayers for fire safety.

6. ACKNOWLEDGEMENT

The authors are grateful to The Principal, Dr. G L Shekar of The National Institute of Engineering, Mysuru for funding this project under TEQIP – II. The authors also extend gratitude to the Head of the Department and technical staff of Computer Science and Engineering Department for providing the lab facility during the project development.The authors express their gratitude for Ms Anitha C ,who provided invaluable guidance throughout the duration of the project and has been a source of constant inspiration.

7. REFERENCES

[1]. Gas Leakage System, D Hari Priya, Lalith Babu, International Journal of Scientific and Research Publications, Volume 4, Issue 9, September 2014, ISSN 2250-3153.
[2]. Advanced Vehicle Monitoring and Tracking System, Prashanth A Shinde, Y B Mane, IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO), 2015.
[3]. Available:https://www.eprolabs.com/wpcontent/uploads/2016/07/SEN-0046.pdf
[4]. Available: https://learn.adafruit.com/raspberry-Pi-analog-to-digital-converters/mcp3008
[5]. Available:https://developer.android.com/guide/components/services.html
[6]. Available: https://www.twilio.com/docs/libraries/python
[7]. Available: http://loopj.com/android-async-http/
[8]. Available:https://developer.android.com/training/basics/data-storage/shared-preferences.html
[9]. Available: https://www.raspberryPi.org/resources/learn/
[10]. Available: https://learn.sparkfun.com/tutorials/voltage-dividers