Smart Phone based Cost Effective Visitor Management System for Smart Offices

https://doi.org/10.3991/ijim.v12i6.9476

Santhosha Rao(✉), Smitha A, Kunal Kulkarni
Manipal Academy of Higher Education, Manipal, Karnataka, India
santhosha.rao@manipal.edu

Abstract—In recent years, the proliferation of the Internet of Things (IoT) has kick started the home and office automation in a very rapid manner. The paper demonstrates a cost effective implementation of an IoT system for managing the visitors in an office environment. The automation system comprises low cost NodeMCU based Wireless Transmitter, NodeMCU based Wireless Display Unit and an android mobile phone. The mobile phone also serves as Wireless Access Point to which the Wireless Transmitter and the Wireless Display Units are wirelessly connected for exchanging the messages using UDP protocol. The Wireless Transmitter and the Wireless Display Units are kept in the visitor's waiting area. The consulting person possesses an android mobile phone in which the automation software is installed. The visitor enters a message using the Wireless Transmitter and notes down the acknowledgement token number sent by the automation software. When the consulting person checks this message, the same token number is sent to the Wireless Display Unit signaling the visitor to consult the person.

Keywords—Internet of Things (IoT), NodeMCU, Office automation

1 Introduction

Whenever a visitor enters into the office area, the first person to greet him is the receptionist. The role of a receptionist is manifold. A receptionist handles varieties of tasks such as greeting the people, phone answering, data entry, package delivery, providing concierge services etc. Receptionists are the brand ambassadors of the company and they shape the first impressions that the people have on the company. Due to the advancement in technology and availability of automated visitor management systems, automated phone answering, online appointment scheduling software etc. some people in the business argue that the role of the receptionist is becoming obsolete. But this should not be the case. The technology should facilitate the receptionist to spend less time on the administration and more time on creating a positive impression among the visitors towards the business. The technological assistance provided to the receptionist should be cost effective so that the business head never makes up his mind to replace the receptionist totally with the automated systems. Nevertheless, it should be user friendly and powerful enough to be used in the absence of receptionist. The automated systems should be designed by considering these facts.

Owing to the advancement in the mobile internet technology, the number of smart phone users are increasing considerably day by day. These devices started influencing the day to day activities of almost every human being. The operating system in majority of these smart phones permit the users to install varieties of third party applications available in the application store. Nonetheless, user can also develop and use any custom application of his choice by using open source application development tools. The presence of powerful hardware and software on par with the laptops and desktops also increased the number of smart phone users considerably for their day to day tasks. The cost effective technological solutions could be provided to the receptionist by exploiting these capabilities of the smart phones along with the low cost IoT solutions available in the market. One such device which is extensively used in designing low cost applications is the nodeMCU [1]. This is an open source development board and firmware based on the ESP8266 -12E WiFi module [2] which can be programmed by LUA programming language or Arduino IDE. The ESP8266-12E module supports complete TCP/IP protocol stack. The board can be directly flashed from an USB port. It combines the features of a microcontroller and wireless access point/station. It can even be used to host a web server or connect to internet to fetch or upload data. With GPIOs, PWM, UART, I2C, 1-Wire and ADC modules integrated, the board provides the best platform for IoT applications with low cost. ESP8266 with NodeMCU Firmware can be configured as an Access Point, Wifi Client or both as Client and Access Point concurrently. It has capability to work with 802.11a, 802.11b and 802.11n wireless networks. Owing to its low cost and powerful hardware it has been used by the IoT system developers over the last few years.

2 Literature Review

Several works were reported in the literature based on ESP8266 WiFi module. In this section, a brief summary of the works are presented.

An indoor air quality monitoring system based on IoT has been proposed in [3]. The system uses Arduino Uno to send the data from sensors to ESP8266 which in turn forwards the data wirelessly to the Thigspeak platform.

A robust home automation system to deal with the complexity, multiple incompatible standards, and the overall costs in the existing systems has been proposed in [4]. The system employs ESP8266 for interaction with sensors and actuators.

An IoT based Hybrid Energy System using ESP 8266 has been designed in [5]. This system switches between solar and wind energy through a website using ESP8266 WiFi module.

In [6] an air conditioning control system using REST protocol based on nodeMCU ESP8266 has been proposed. In this system, the NodeMCU is connected to an IR LED to send the IR command and LDR to know the AC power condition status.

A reliable, flexible, secure and economical sensor network for smart homes has been presented in [7]. In this system, the heterogeneous sensor and actuator nodes based on

ESP8266 devices are deployed in the home environment. These nodes communicate to a middleware which runs home automation server. This server also operates as broker to facilitate the MQTT connectivity protocol. The middleware facilitates controlling of wireless nodes over local as well as remote network.

In [8], a smart home automation using IoT based sensing and monitoring platform has been designed. In this system, EmonCMS platform has been used for collecting and visualizing monitored data and remote controlling of home appliances and devices. The sensing of different variables inside the house is accomplished using the NodeMCU-ESP8266 microcontroller board.

A lightweight and universal intelligent service platform that provides efficient operations and various value-added services for indoor IoT systems has been proposed in [9]. The proposed platform has a flexible framework which consists at most four layers: terminal, gateway, local server and cloud server layers. In this platform, ESP8266 is used for the design of lightweight terminals.

In [10], an offline domotic system using voice commands has been presented. The system comprises of an user interface and a communication platform for a modular home automation system. A speech recognition module is used to develop the user interface. This interface facilitates the identification of commands from different users. Communication among the modules is established via ESP8266.

An Energy Preserving System for Smart Rooms has been presented in [11]. This system uses ESP8266 to provide lighting controls.

A data center temperature monitoring with ESP8266 based wireless sensor network and cloud based dashboard with real time alert system has been proposed in [12]. In this system, ESP8266 connects to cloud through API for posting temperature data to the cloud dashboard on real time and the cloud event management system generates alerts whenever the high temperature alert event is fired.

3 System Model

The proposed system model is depicted in Figure 1. It consists of a NodeMCU based Wireless Transmitter, NodeMCU based Wireless Display and an Android mobile. The android mobile phone also serves as Wireless Access Point.

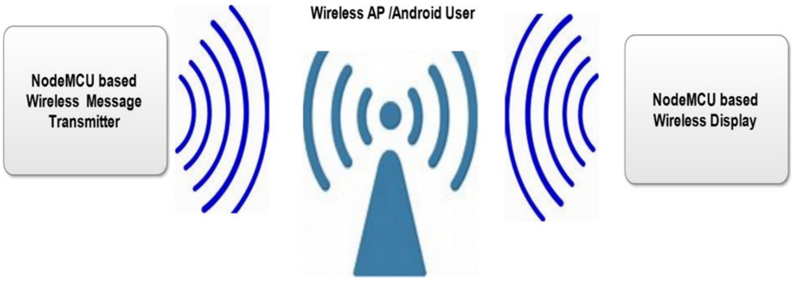

Wireless AP /Android User
NodeMCU based Wireless Message Transmitter
NodeMCU based Wireless Display

Fig. 1. System Model

3.1 NodeMCU based Wireless Transmitter

Figure 2 depicts NodeMCU based wireless transmitter. It consists of a 4x3 keypad and an I2C LCD interfaced to NodeMCU development board. As the number of GPIO pins are limited, the LCD is connected to the board using an I2C module (Figure 3). Whenever the transmitter is turned ON, it starts scanning for wireless access point using SSID and password as configured in the Android Mobile. The obtained IP address from the Access Point is displayed on the LCD display, if the Access Point is available. Otherwise, “Not connected” message is displayed. This IP address is used by the android application to communicate with the transmitter.

After displaying the IP address initially, the transmitter waits for the configuration commands from the android application. The configuration commands specify the number of digits in the message and the maximum value for each digit. For example, let us assume that the length of the message is 4 and the maximum values for these digits are 1, 2, 3, 2 respectively. In this case, the first digit can take two values (0 or 1), second digit can take 3 values (0 or 1 or 2), third digit can take 4 values (0 or 1 or 2 or 3) and the fourth digit can take three values (0 or 1 or 2). The Table 1 shows a sample message format for a school application. In this application, the android user is the head of the department and the wireless transmitter is kept outside his room, on the receptionist’s table. The visitor or the receptionist on the behalf of the user enters the 4-digit message and presses the # button to send the message to the android user.

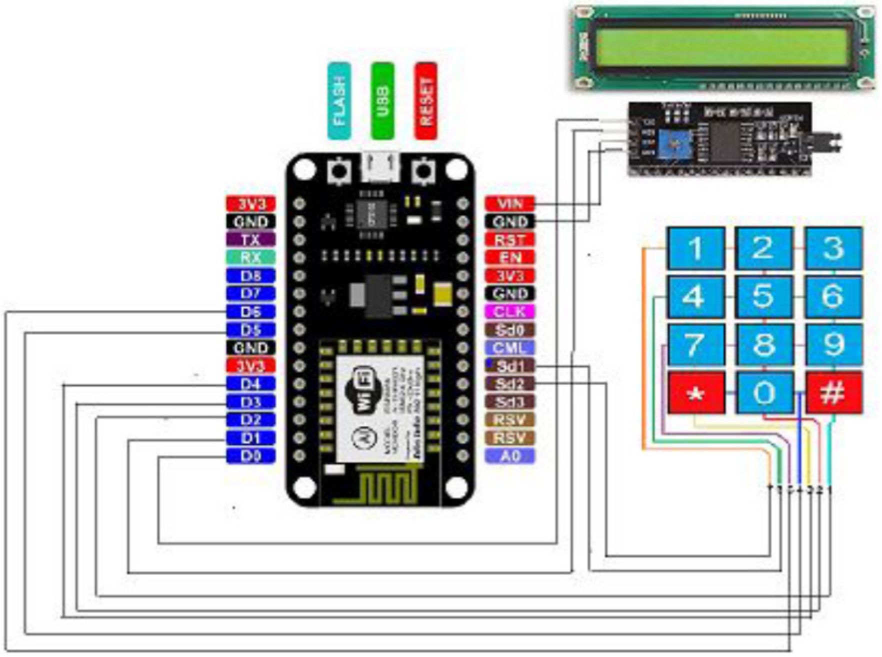


Fig. 2. NodeMCU based Wireless Transmitter

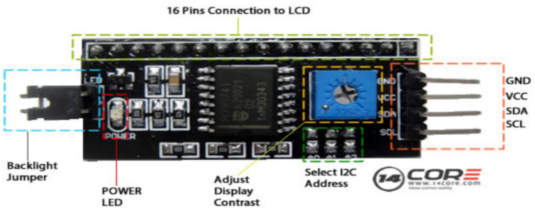


Fig. 3. I2C Module

Table 1. Various fields in the sample message format

First digit	**Second digit**	**Third digit**	**Fourth digit**
0-Male	0-Faculty	0- 5 minutes	0-Low priority
1-Female	1-Student	1-10 minutes	1-Medium priority
	2-Parent	2-15 minutes	2-Higher priority
		3-20 minutes	

3.2 Wireless Display Unit

The Figure 4 depicts the NodeMCU based Wireless Display unit. This unit receives the UDP packets comprising of token numbers from the android user upon selecting a message. The token number is displayed on the LCD display unit.

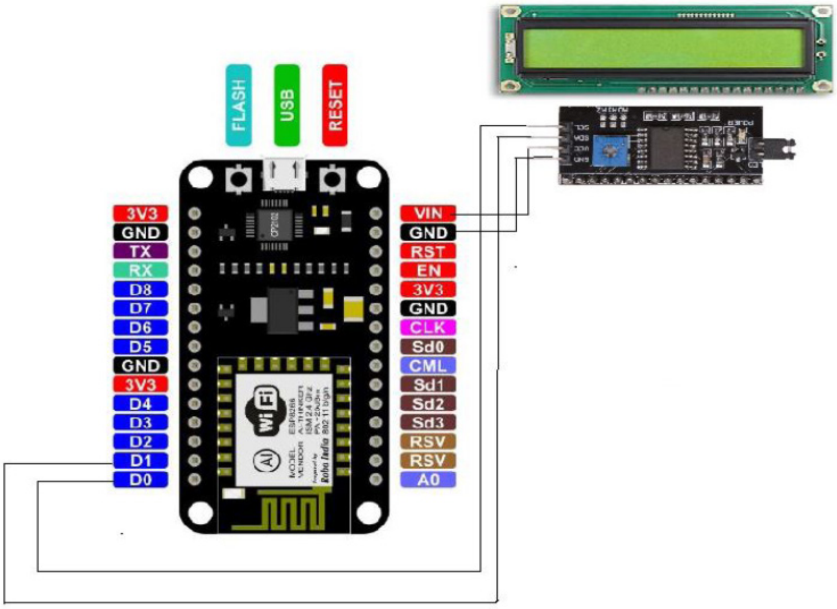


Fig. 4. NodeMCU based Wireless Display Unit

3.3 Android User

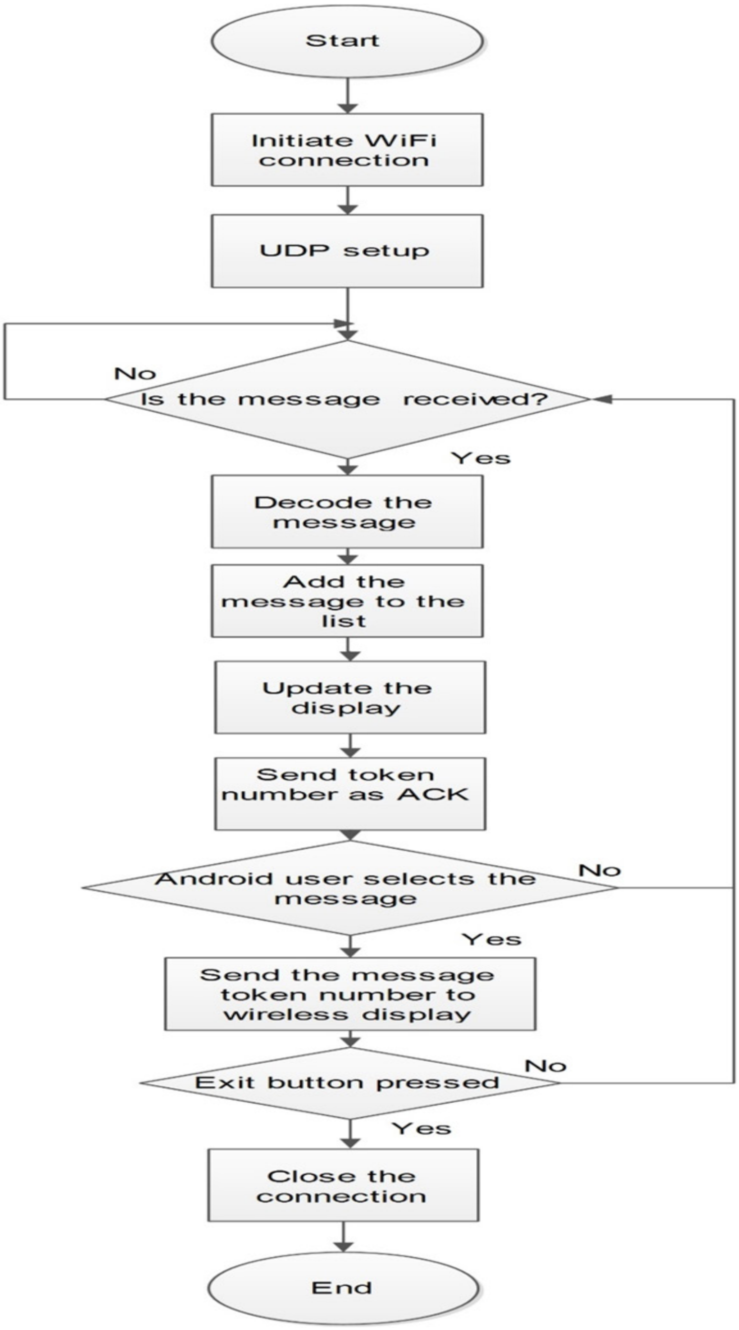


Fig. 5. Android User

Various operations performed by the android user are depicted in the flowchart of Figure 5. The android user configures the android mobile as the access point by enabling the mobile hotspot option. The access point assigns the IP addresses to the Wireless Transmitter and Wireless Display. The assigned IP addresses are displayed on the LCD displays attached to Wireless Transmitter and Wireless Display units. These addresses are noted down for further configuration of the android application to facilitate the network communication between the android user and Wireless Transmitter and Wireless Display units. The android user manually enters these IP addresses in the text boxes and clicks the command button in the user interface to establish connection with these devices. Once the connection is established, the android application starts listening to the messages from Wireless Transmitter using UDP protocol. After receiving the message, the application appends the decoded message to the list and displays it on the mobile screen. The token number is sent back to the Wireless Transmitter as an acknowledgement. The Wireless Transmitter displays the token number to the visitor. The android user selects the message using checkbox displayed against it. He can select the message based on first come first serve or priority basis. Once the message is selected by the android user, the corresponding token number is sent to the Wireless Display Unit using the UDP protocol.

4 Implementation

At the Wireless Transmitter and Display unit, ESP8266WiFi.h, WifiUDP.h and LiquidCrystal_I2C.h libraries are included. The ESP8266WiFi.h library is used to connect to a network with a specific SSID and a password. The library also offers functions to create a network/hotspot through the ESP8266. The WifiUdp.h library is a standalone UDP library used to create datagram packets and exchange them through the Inet Address, Port numbers etc. The send() and receive() methods provided by the library are used in exchanging the packets. The LiquidCrystal_I2C.h library is useful for all the LCD processes such as init(), display(),clear() etc. It allows to send commands to the I2C module through D1 and D2 pins of NodeMCU.

At the android user side, the application is mainly divided into 4 activities, namely Home Activity, Connect Activity, Set Message Activity and View Devices Activity.

4.1 Connect Activity

This activity accepts the IP address and the port number of the microcontroller to be connected in two EditTexts and stores them in a constant array. On Button click, the application is ready to receive UDP packets/messages. Figure 6 depicts the GUI for connect activity.

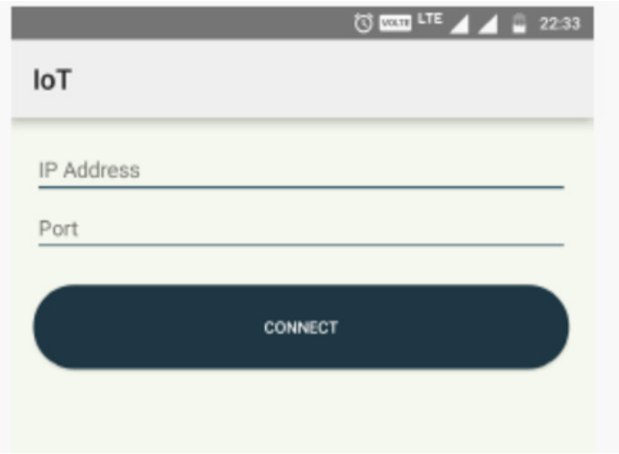


Fig. 6. Connect Activity

4.2 Set Message Activity

This activity uses the LayoutInflater function of android development. The number of digits in the message are asked from the user and accordingly a list is inflated containing the number of buttons specified as depicted in Figure 7. Each button allows the user to enter a maximum of 10 messages and upon clicking the save button, the values are stored in constant array lists. A Spinner with Inflated Buttons is present in this activity. For each positional digit value, the message to be displayed upon receiving the data by the android user is set in this activity as depicted in Figure 8.

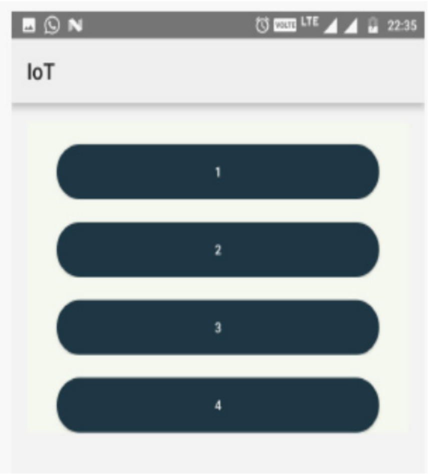
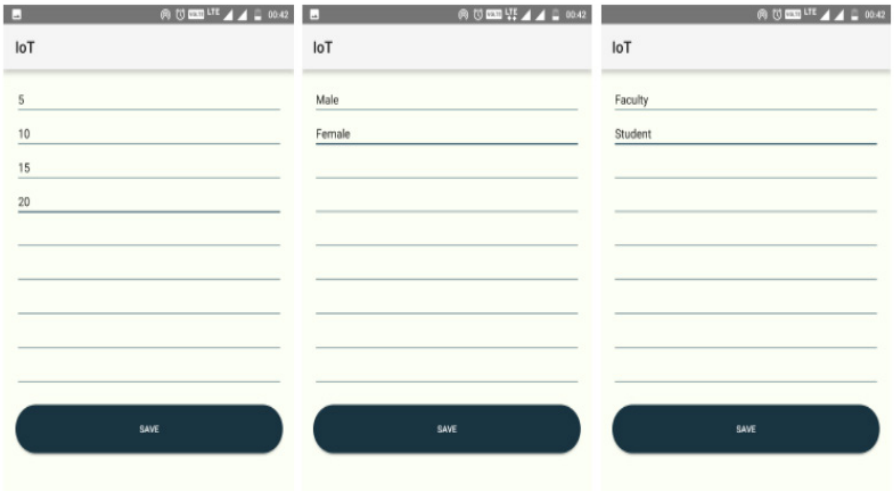


Fig. 7. Set Message Activity



IoT

5

10

15

20

SAVE

IoT

Male

Female

SAVE

IoT

Faculty

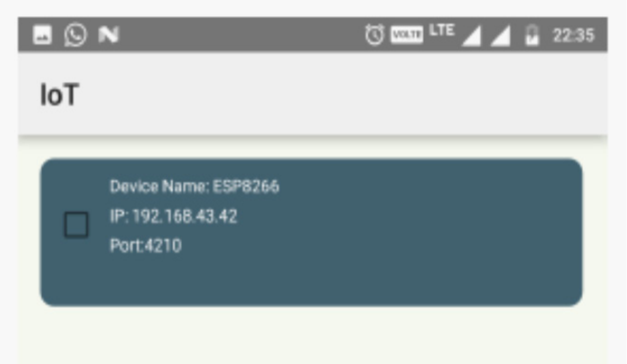
Student

SAVE

Fig. 8. Message to be displayed upon receiving the data

4.3 View Devices Activity

In this activity, the connected devices are displayed along with their IP addresses and port numbers as shown in Figure 9. Upon clicking the checkbox a list pops up showing all the decoded messages sent from the Wireless Transmitter.



IoT

□ Device Name: ESP8266
IP: 192.168.43.42
Port:4210

Fig. 9. View Devices Activity

4.4 Home Activity

Figure 10 depicts the home activity GUI. This activity is a center point to all the activities. All the links to other activities are present in this activity. This activity also runs a continuous thread which updates the messages received into an array list. This is done using a handler class. A handler allows one to send and process messages and runnable objects associated with the message queue.

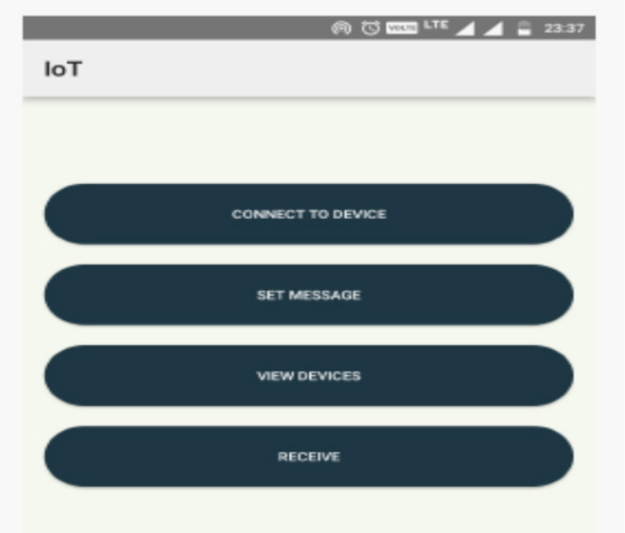
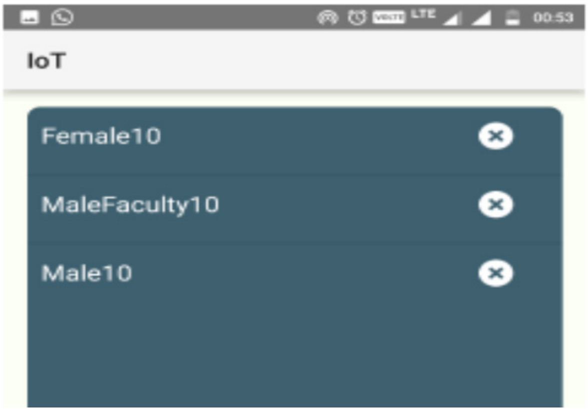

IoT

Fig. 10. Home Activity

5 Results

The received messages are shown in Figure 11. There are three visitors in the queue, namely a female student, a male faculty and a male student each requiring up to 10 minutes of discussion time. Since the priority of visit is low, it is not explicitly displayed along with the messages. The android user selects the message by clicking the checkbox against it. This event sends the token number corresponding to the message to the Wireless Display Unit. The buzzer connected to the display draws the attention of the visitors waiting outside. At this juncture, the visitor whose token number matches with the number displayed on the display enters the room for discussion.


IoT

Female10

MaleFaculty10

Male10

Fig. 11. Received messages at the Android User

6 Conclusion

The proposed visitor management system provides a cost effective technological solution to the smart offices by exploiting the capabilities of the android smart phones along with the low cost NodeMCU ESP8266 WiFi modules available in the market. The system helps in enhancing the productivity of the receptionists by reducing the burden on them. It can even be deployed in the absence of receptionists also. In any case, it is the good idea to display the operational instructions so that the system can be used without any additional assistance.

7 References

[1] NodeMCU website: http://nodemcu.com/

[2] ESPressif Website. http://www.espressif.com/

[3] G. Marques and R. Pitarma, Monitoring Health Factors in Indoor Living Environments Using Internet of Things, Advances in Intelligent Systems and Computing Recent Advances in Information Systems and Technologies, pp. 785–794, 2017.

[4] T.A. Abdulrahman, O.H. Isiwekpeni, N.T. Surajudeen-Bakinde, A.O. Otuoze, Design, Specification and Implementation of a Distributed Home Automation System, Procedia Computer Science, Volume 94, pp. 473-478, 2016.

[5] Prakhar Srivastava, Mohit Bajaj, Ankur Singh Rana, IOT based controlling of hybrid energy system using ESP8266, 2018 IEEMA Engineer Infinite Conference (eTechNxT), New Delhi, March 2018, https://doi.org/10.1109/ETECHNXT.2018.8385294

[6] Laurentius Kuncoro Probo Saputra. Yuan Lukito, Implementation of Air Conditioning Control System Using REST Protocol Based on NodeMCU ESP8266, 2017 International Conference on Smart Cities, Automation & Intelligent Computing Systems, Yogyakarta, Indonesia, November 2017, https://doi.org/10.1109/ICON-SONICS.2017.8267834

[7] Ashutosh Bhatt, Jignesh Patolia, Cost effective digitization of Home Appliances for Home Automation with low-power WiFi devices, International Conference on Advances in Electrical,Electronics,Information,Communication and Bio-Informatics(AEEICB16), Chennai, India, Feb 2016, https://doi.org/10.1109/AEEICB.2016.7538368

[8] Majid Al-Kuwari, Abdulrhman Ramadan, Yousef Ismael, Laith Al-Sughair, Adel Gastli, Senior Member, Mohieddine Benammar, Smart-home automation using IoT-based sensing and monitoring platform, 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018), Doha, Qatar, April 2018, https://doi.org/10.1109/CPE.2018.8372548

[9] Zheng Wan, Jun Li, Wenqiang Duan, Yuechi Chen, Naixue Xiong, Lightweight and Universal Intelligent Service Platform in Indoor Environment, Cluster Computing., Springer US, Feb 2018, https://doi.org/10.1007/s10586-018-2099-x

[10] Javier Errobidart, Alejandro Jose´ Uriz, Esteban Gonza´lez, Iva´n Exequiel Gelosi, Juan Alberto Etcheverry, Offline Domotic System using voice commands, 2017 Eight Argentine Symposium and Conference on Embedded Systems (CASE), Buenos Aires, Argentina, Aug 2017, https://doi.org/10.23919/SASE-CASE.2017.8115370

[11] Khalid M.O Nahar, Ra'ed M Al- Khatib, EPSSR: Energy preserving system for smart rooms, 2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS), Amman-Jordan, Dec 2017, https://doi.org/10.1109/IT-DREPS.2017.8277800

[12] Saraswati Saha, Anupam Majumdar, Data Centre Temperature Monitoring with ESP8266 Based Wireless Sensor Network and Cloud Based Dashboard with Real Time Alert System, 2017 Devices for Integrated Circuit (DevIC), Kalyani, India, March 2017, https://doi.org/10.1109/DEVIC.2017.8073958

8 Authors

Santhosha Rao received B.E. degree in Electronics and Communication Engineering from Mangalore University and M.Tech degree in Digital Electronics and Advanced Communication from Manipal University, Manipal, India. He obtained his Ph.D degree from Manipal University, Manipal in the area of Cross Layer Design in Energy constrained Wireless Networks. Since 1998 he has been with Manipal Institute of Technology, MAHE, Manipal, India, where he is currently Senior Associate Professor in the Department of Information and Communication Technology. He has published several research papers in National and International conferences and journals.

Smitha A is currently working as an Assistant Professor in the Department of Information and Communication Technology in Manipal Institute of Technology, MAHE, Manipal, India. During her post-graduate course, she served as an Intern in ESIGELEC, France. She has 5 years of teaching experience and has research interests in the field of Internet of Things, Image Processing and Pattern Recognition. She also has research experience in the field of secure data transfer in Vehicular Adhoc Networks and wireless communication.

Kunal Kulkarni received his B.Tech in Computer and Communication Engineering from Manipal University, Manipal. He is a proficient Java Programmer and AWS certified solution Architect. Currently, he is working on projects pertaining to modern cryptographic algorithms and different web services.

Article submitted 03 September 2018. Resubmitted 16 October 2018. Final acceptance 20 October 2018. Final version published as submitted by the authors.