A voice controlled smart home automation system using artificial intelligent and internet of things

Mohamed A. Torad[1], Belgacem Bouallegue[2], Abdelmoty M. Ahmed[2]
[1]Department of Communication and Electronics Engineering, Higher Technological Institute, El Sharkeya, Egypt
[2]Computer Engineering Department, College of Computer Science, King Khalid University, Abha, Saudi Arabia

Article Info

Article history:

Received Jun 19, 2021
Revised May 27, 2022
Accepted Jun 06, 2022

Keywords:

Arduino
AI
Chat form
IoT
NLP
Raspberry Pi

ABSTRACT

The objective of this work is to take a step further in this direction by incorporating voice control and artificial intelligence (AI) into internet of things (IoT)-based smart home systems to create more efficient automated smart home systems. Accordingly, a home automation system proposal is presented, in which the related functions can be controlled by voice commands using an android or web application via a chat form. The user issues a voice command, which is deciphered by natural language processing (NLP). To accommodate the user's request, the NLP classifies it into operation commands. Arduino and Raspberry Pi are used to translate the commands extracted from NLP into reality. Based on this, home applications can be controlled. Also, the utilities consumption could be calculated, saved, and paid on time. This is in addition to the introduction of a machine learning (ML)-based recommendation system for automated home appliance control. In this approach, the mobile or web application is considered as the central controller, deciding the appropriate actions to fulfill the user's desires. The presented work has been put into practice and tested. It proved to be applicable, as well as having the potential for making home life more comfortable, economic, and safe.

Corresponding Author:

Mohamed A. Torad
Department of Communication and Electronics Engineering, Higher Technological Institute
10th of Ramadan City, El Sharkeya, Egypt
Email: mohamed.torad@hti.edu.eg

1. INTRODUCTION

Artificial intelligence (AI) is one of the newest fields in science and engineering. The Turing test is used to determine if a system is intelligent or not. A machine passes the test if a human inquisitor can't identify if the written responses are from a machine or a person after posing a series of written questions.

Natural language processing (NLP) is an AI application that represents a computer science discipline that converts human voice into commands and is classified under human-computer interaction category. The majority of NLP algorithms are built using machine learning (ML). NLP is employed here to allow the user to interact with household appliances using a human voice based on machine learning [1]-[7].

Recently, what is known as the internet of things (IoT) has made it possible to connect machines to machines. Building new services over the network using radio-frequency identification (RFID) technology combined with sensors and/or mobile applications helps a lot in positively driving IoT applications [8]-[16]. Intelligent home systems have grown rapidly in recent years and they are being at the core of IoT, information technology, and communication technologies. In a smart home system, power and lighting consumption can be controlled and monitored remotely using web interface or mobile devices. To apply IoT, Arduino and Raspberry Pi have been used to translate the commands derived from NLP into actions.

This paper integrates IoT and voice recognition technology in addition to NLP-based AI to improve smart homes systems control. The paper is divided into five sections; section 1 introduces the study; section 2 highlights relevant works; section 3 details the proposed system and its components; section 4 displays system implementation; and section 5 concludes the study. IoT and AI are now integral components of modern systems, and doping IoT with AI has been documented in numerous studies.

Dey *et al.* [17] used only the wireless sensors to build home automation systems, using the hypertext markup language (HTML) web page as interface to correlate between Arduino boards and web page. David *et al.* [18] presented an alternative method that uses a Raspberry Pi board rather than an Arduino board. They presented Android applications that might be used to control the appliances' functions, however voice commands are not supported. Kumar and Shimi [19] used an Arduino board, a relay circuit, and a voice recognition module in addition to an adjustable bed to create a home automation system with voice commands, however the functionality was limited. Satria *et al.* [20], facilitates mobile device Bluetooth as an interface to govern the home appliances, but the main drawback in this work was the Bluetooth coverage distance limitation. Sen *et al.* [21], overcame this drawback by switching from Bluetooth to the global system for mobile (GSM) for home appliance control, resulting in a progressive increase in the total cost of the home automation system deployment. Erić *et al.* [22], demonstrated the differences between text to speech and speech to voice engines in the context of a smart home system. Baby *et al.* [23], proposed creating a smart home system using a chatbot based on natural language processing. A web application has been used to facilitate user authentication for security reasons. Quadri and Sathish [24], developed an IoT monitoring system that allows them to control the door wirelessly using a smartphone and a Raspberry Pi. Yousef and Torad [25] created a well-organized conversational AI agent that can understand and analyze human behavior based on day-to-day encounters and interactions, in addition to the State of the art of voice control for home/domotic systems mentioned in [26].

Smart home systems-related studies have been evaluated, summarised, and contrasted in Table 1 for the sake of brevity. The report also provided a comparison of the proposed smart home system to highlight its key characteristics in contrast to existing systems. The suggested system, as shown in the table, seeks to overcome the constraints of the existing system and is accompanied with a real-world implementation scenario for validation reasons. The contributions of this paper, as well as the model presented in [25] will be integrated with IoT model to manipulate a more efficient smart home system.

Table 1. Summary of smart home systems

	Focus criteria										
Smart home system	Indoor control	Security	Safety	Monitoring	Weather forecast	Wireless interface	Controller	Smartphone	Web-based	NLP	Machine learning
Dey *et al.* [17]	√	√		√		WiFi	Arduino Mega		√		
David *et al.* [18]	√		√	√		Bluetooth/WiFi	Arduino Mega	√	√		
Kumar and Shimi [19]	√						Arduino Mega	√			
Satria *et al.* [20]	√			√		Bluetooth	Arduino Mega	√			
Sen *et al.* [21]	√			√		GSM	Arduino Mega	√			
Erić *et al.* [22]	√			√		WiFi	Arduino		√	√	
Baby *et al.* [23]	√	√	√	√		WiFi	Raspberry Pi		√		√
Quadri and Sathish [24]	√	√	√	√		WiFi	Raspberry Pi	√	√		
Proposed system	√	√	√	√	√	WiFi	Arduino Mega + Raspberry Pi + node MCU	√	√	√	√

2. RESEARCH METHOD

The proposed system architecture is depicted in Figure 1. This work aims to make voice commands in human natural language a better and easier way to connect with smart home appliances or web applications. Also a recommendation system to control home appliances is provided. A back-end server is configured, especially for heavy processing operations such as NLP, recommendation system, and image processing. The server is created via an open source microframework for python (flask). The present system includes the following main modules:

a) Authentication module
b) Remote voice control module
c) Smart recommendation module
d) Weather information module

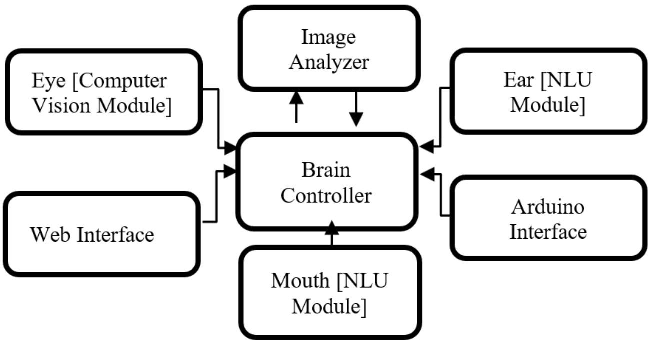


Figure 1. The proposed system architecture

2.1. Authentication module

The proposed system can perform autontication by two ways. First by captures the user's face image and compares it to the one saved on the system server, and if the two match, the user is authenticated. Secondly, a username and password can be used for authentication. It should be noted that the image identification is carried out utilizing image processing algorithms via computer vision module using OpenCV library. This process can be done through android or web applications. This authentication system is a one-time process that increases the credibility of the system.

2.2. Remote voice control module

At chat bot, the user's voice commands are converted from speech to text and then delivered to the server via a hypertext transfer protocol (HTTP) request. NLP consists of eight layers as [25]:

a) Layer 1 (expanding contractions): this layer uses the pros of the regular expressions module supplied in the python language.
b) Layer 2 (tokenization): it is a pre-processing phase achieved to split the incoming string into a list of tokens or words. Process done through the natural language toolkit (NLTK) library.
c) Layer 3 (spelling correction): where the spelling correction algorithm is applied.
d) Layer 4 (intent and tense detection): text-classifier used for the purpose of detecting user's intent. Naïve-Bayes classifier algorithm was used [17].
e) Layer 5 (part-of-speech tagging): tag each of the resulted list of tokens into its suitable part-of-speech (POS) tag i.e. proper noun/determiner/preposition. this layer process done through the NLTK library.
f) Layer 6 (information extraction layer): which facilities the pros resulted from the chunking in addition chinking methods supported by the NLTK library for the purpose of extracting the vital chunks from user's input list and aggregating them into a specific list, e.g. ['coffee machine', 'on'].
g) Layer 7 (organization): the resulted extracted information list is organized into an exact dictionary which contains an appropriate label to every entry, e.g. {'appliance': 'light', 'state': 'on'}.
h) Layer 8 (execution): it provides a controlling machine upon all the previous layers.

The NLP algorithms perform a process on the given texts extracted from the human voice to determine the user intent. Accordingly, the needed algorithms are run for realizing this intent. All the processes are completely processed and provide the appropriate response which is done by the NLP in the order of milliseconds. The system server is connected to Raspberry Pi via message queue telemetry transport (MQTT) connectivity protocol. Representational state transfer application program interface (RESTful API)'s is utilized to make it more efficient and easier to establish a connection between the server and web or mobile applications.

2.3. Smart recommendation module

Recommender systems have grown in popularity in recent years, and they are now used in a number of applications such as news, movies, books, music, search queries, research articles, social tags, and other similar applications. The NLP represents the first step in the recommendation system and is usually followed by a ML step. A recommendation system is a subclass of information filtering system that looks for ways to predict the preference or rating of the user's behaviour in relation to various home appliances. As illustrated in Figure 2, there are three possible approaches to developing a recommendation system. The approach under consideration is a "content-based filtering technique" which collects and analyzes data based on users' behaviour. This is done by using nearest neighbour algorithm, which is characterized by fast computation of nearest neighbours. This is still an active zone of research in ML recommender systems. The nearest neighbour search implementation involves the brute-force computation of distances between all pairs of points in the dataset. The recommender system considered in this paper is developed in python language using three modules, namely sci-kit learn, pandas and numpy. The three modules are referred to in python as indicated in Figure 3.

The recommender class is initially implemented with the aid of its constructor, then the selected dataset and user values are continuously passed to the selected pandas module. After that, the chosen data is fitted to the proposed algorithm, and a large number of neighbors is determined. Accordingly, the suitable behaviour is recommended by the system.

If a user requests to watch a movie on the Internet, for example, there are a lot of options, creating a problem for many Internet users. Therefore, certain delivered information must be filtered, preferred, and/or excelled. The recommender system will generate a list of recommended movies based on the user's previous choices. This is accomplished by exploring a vast database of information and delivering personalized content to users.

2.4. Back-end controller module

Back-end controller is important because it is responsible for the communication between the module implemented and all the interfaces, all interfaces discussed in details into next section. The python programming language was chosen since its libraries are open source and well-suited to this proposed work. The "flask", a micro web framework developed in python editor, was chosen. The server in question is configured using "flask", and is connected with the android and web interfaces by through a "restful application program interface" (API), which uses HTTP requests to get, put, post and delete data.

For the hardware branch, we use MQTT cloud service. MQTT is a machine-to-machine (M2M)/connectivity protocol. It was designed as a very lightweight subscribe/publish messaging transport. It is valuable for connections with remote locations where a minor code footprint is essential and/or network bandwidth is at top.

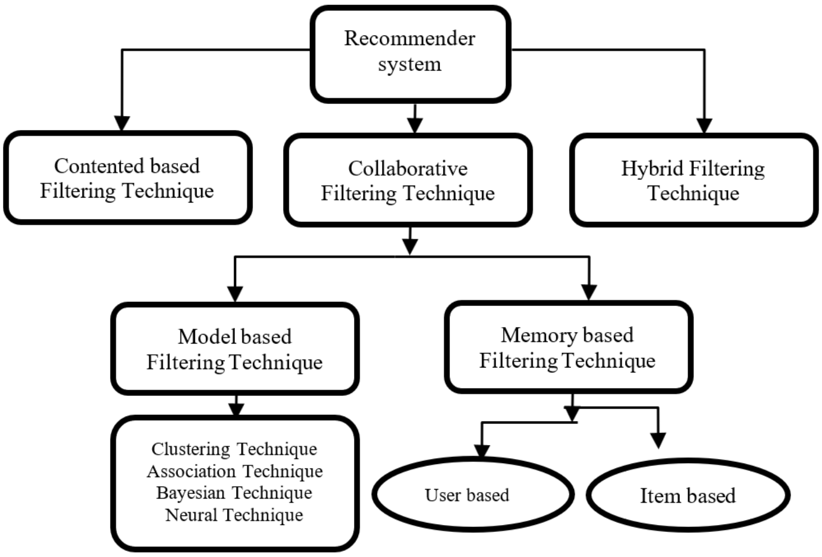
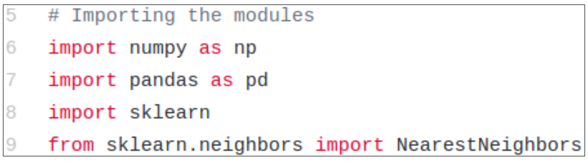

Recommender system
Contented based Filtering Technique
Collaborative Filtering Technique
Hybrid Filtering Technique
Model based Filtering Technique
Memory based Filtering Technique
Clustering Technique
Association Technique
Bayesian Technique
Neural Technique
User based
Item based

Figure 2. Classification of the recommender systems

3. RESULTS AND DISCUSSIONS

The following are presented as proofs of concept: a) movie recommender: it's used to recommend movies for user. This recommendation based on the user dataset. So, based on user behaviours recommendation generated for the user; and b) temperature degree: returns the temperature of a specific city by means of weather API. (The above two functions are shown in Figure 4). Figure 5 depicts the implementation of the main API, which uses the POST (HTTP request) method to receive data from the interface, and the NLP module to interpret the user's intents via chatbot or voice commands. If there is a request for activation/ deactivation a certain appliance, the API delivers it to the hardware, which generates a response using the mouth module and sends it back to the interface.



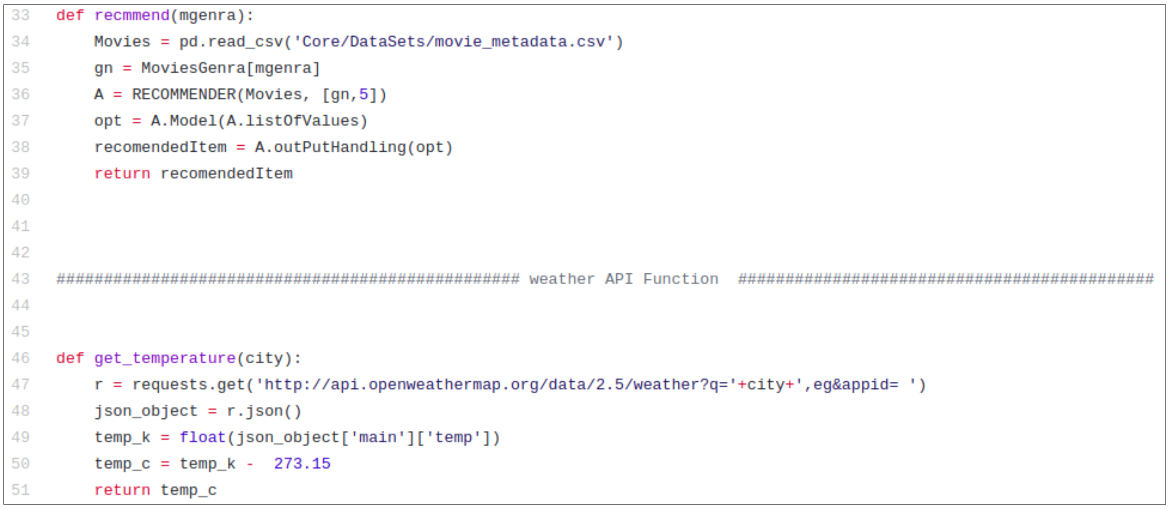
```python
# Importing the modules
import numpy as np
import pandas as pd
import sklearn
from sklearn.neighbors import NearestNeighbors
```

Figure 3. Importing three modules in python used in ML

3.1. Air-conditioning module

The air-conditioner model is implemented using numpy, pandas and scikit-learn python modules. To generate a dataset of the user’s behaviour and train on it to generate and predict results based on day and time attributes, the proposed classification system processes are performed on the server, and certain data is required from the server, which is gathered by the sensors. and make a classification based on the user’s previous decisions and user’s data to classify turning the air-conditioner on or off then send the value that has been predicted to the server. Decision tree is the classification model used here. Such data includes:

a) Date: the date on which the decision has been made.
b) Time: the time in which the decision has been made.
c) Interior value: temperature degree inside.
d) Exterior value: temperature degree outside.
e) User value: the user decision to turn the conditioner on or off.



```python
def recmmend(mgenra):
    Movies = pd.read_csv('Core/DataSets/movie_metadata.csv')
    gn = MoviesGenra[mgenra]
    A = RECOMMENDER(Movies, [gn,5])
    opt = A.Model(A.listOfValues)
    recomendedItem = A.outPutHandling(opt)
    return recomendedItem



################################################## weather API Function  ############################################


def get_temperature(city):
    r = requests.get('http://api.openweathermap.org/data/2.5/weather?q='+city+',eg&appid= ')
    json_object = r.json()
    temp_k = float(json_object['main']['temp'])
    temp_c = temp_k -  273.15
    return temp_c
```

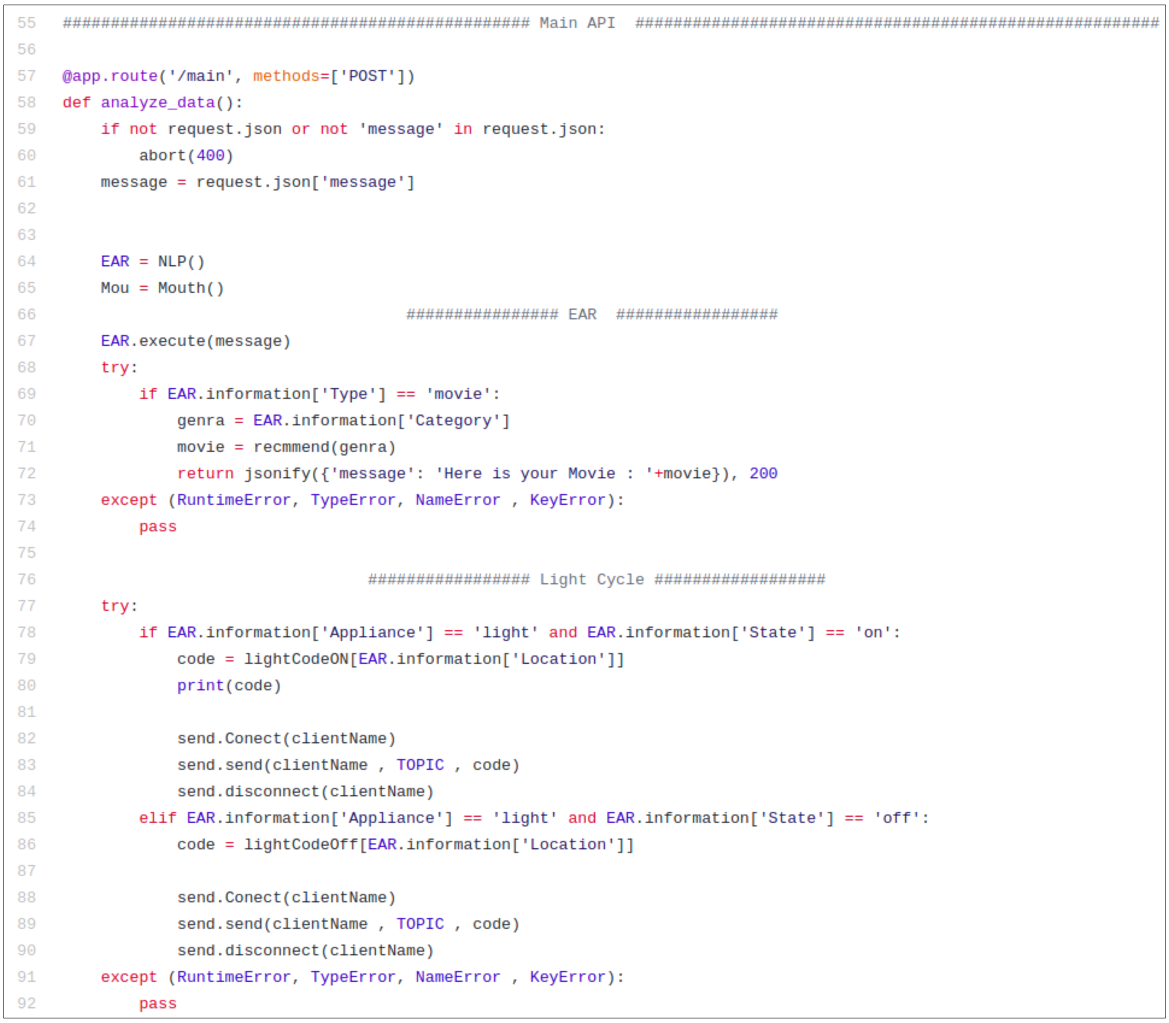
Figure 4. Movie recommender system and weather API function calls

3.2. Home appliances control

With the help of a classification model, it was possible to predict the user’s behavior in turning on/off (e.g., light) in the home automation system based on his behavior with the time of day and the room light state throughout several months of the year. Classification models provide an overcome and predictions for type of suggestion by training on several samples or data created by saving the user’s behaviour on the home lights in a dataset. This dataset will be available after certain time and could be improved with dataset rising. Naïve Bayes classifier have been employed to provide reasonable accuracy based on the Bayes rule and secondary probabilities for on and off events.

For the web interface, the front-end used to adapt data into graphical interface for user to interact with and view data via digital interaction using cascading style sheets (CSS), HTML and Java script for the front and back end. Numerous keywords were used such as “web of things”, “internet”, “mobile computing”, “web” and concatenation of them. Current surveys on the web of things (WoT)/IoT and mobile sensing were also studied for related efforts.

On the hardware side, Arduino and Raspberry Pi were used. On the Arduino side, we can utilize Arduino to control nearly any home appliance and make things simple to use by leveraging NLP to give voice commands to the mobile device or web interface, as previously described. When you’re not at home, you can use the internet to control your house’s applications. You may use Arduino to control light, fans, air conditioning, doors, and curtains, as well as calculate water flow, electricity usage, and control television by setting favorite channels and turning it on when you need to use any functions, or hardware such as sensors, motors, and wires. The firmware used was Arduino integrated development environment (IDE), and an Arduino uno kit with 32 kB of memory and 16 MHz clocks was utilized as a control signal to switch functions on/off through a set of relays. We connect Arduino to the internet using the ethernet shield, which is backwards compatible with earlier versions but requires the newer W5500 chip, as well as the ethernet and SD libraries (to read and write SD card).



```
################################################# Main API  #######################################################

@app.route('/main', methods=['POST'])
def analyze_data():
    if not request.json or not 'message' in request.json:
        abort(400)
    message = request.json['message']


    EAR = NLP()
    Mou = Mouth()
                                    ################ EAR  #################
    EAR.execute(message)
    try:
        if EAR.information['Type'] == 'movie':
            genra = EAR.information['Category']
            movie = recmmend(genra)
            return jsonify({'message': 'Here is your Movie : '+movie}), 200
    except (RuntimeError, TypeError, NameError , KeyError):
        pass

                                ################# Light Cycle ##################
    try:
        if EAR.information['Appliance'] == 'light' and EAR.information['State'] == 'on':
            code = lightCodeON[EAR.information['Location']]
            print(code)

            send.Conect(clientName)
            send.send(clientName , TOPIC , code)
            send.disconnect(clientName)
        elif EAR.information['Appliance'] == 'light' and EAR.information['State'] == 'off':
            code = lightCodeOff[EAR.information['Location']]

            send.Conect(clientName)
            send.send(clientName , TOPIC , code)
            send.disconnect(clientName)
    except (RuntimeError, TypeError, NameError , KeyError):
        pass
```

Figure. 5. Main API function calls

In this paper, MQTT is used to connect Raspberry Pi IoT devices and control them from anywhere in the world. MQTT is a lightweight messaging protocol that is ideal for communication between connected devices in the internet of things. MQTT is made up of three components: a broker, a publisher, and a subscriber. A broker is a type of intermediary that facilitates communication between devices. A publisher is a device that sends messages. A subscriber listens to the messages sent by the publisher. There is one more important component in MQTT known as topic. A topic is necessary for communication between different devices. For example, “device a” wants to send a message to “device b”. To do so, there must be a point of convergence between the two, which is the topic. Cloud MQTT is a broker service that provides free MQTT communication for a certain number of devices. Here, the system is linked to a smart mirror control that displays the time, date, weather outdoors, and impending calendar appointments. The hardware requirements integrated with the Raspberry Pi 3 are liquid crystal display (LCD) display, wooden frame, high definition multimedia interface (HDMI) and Power supply programmed with python integrated development and learning environment (IDLE) 2. Many libraries were used to construct the smart mirror program:

a) Tkinter library: to create graphical user interface (GUI) programs.
b) Locale library: open access to the portable operating system interface (POSIX) (to access to locale database).
c) Threading library: accelerate the program by doing many parallel operations instead of doing operations sequentially.
d) Time library: to connect the time to the date of the personal computer (PC).
e) Requests library: to open and retrieve links.
f) Java script object notation (JSON) library: to send links.
g) Traceback library: (most recent call last) print exception tracebacks inside your programs.
h) Feedparser library: to analyze the abstracts (e.g. rich site summary (RSS)).
i) Python imaging library (PIL) library: to handle images in terms of image processing in an easy way.
j) Context manager: to run two processes with one another, if in a set of instructions among them Google News, free geoip were used in smart mirror for getting news to show (my internet protocol (IP), latitude, longitude, country code, country name) followed by dark sky to take the weather (create account on it then taking the link which on it my weather).

Smart home applications can be accessed by mobile application or web application, with user authentication performed by collecting the user's face image and comparing it to the server database. After that, the server corresponds the result of image processing algorithms via computer vision module using OpenCV library to identify the user (or user can authorize and access by entering his own username and password). The authentication system is a one-time process that increases the credibility of the system. The user voice commands are converted from speech to text and then sent via an HTTP request to our server.

At the user login case, if user already had an account. In the event that the user needs to log in, the system generates a login form and asks the user to enter his face photo, after which the system validates the authorization. If the user is authorized, the system will open and the user will be successfully logged in; if the user is not authorized, an error message will be displayed and the login will be ended at the not recognized face. For clarification login flowchart is diplayed in Figure 6.

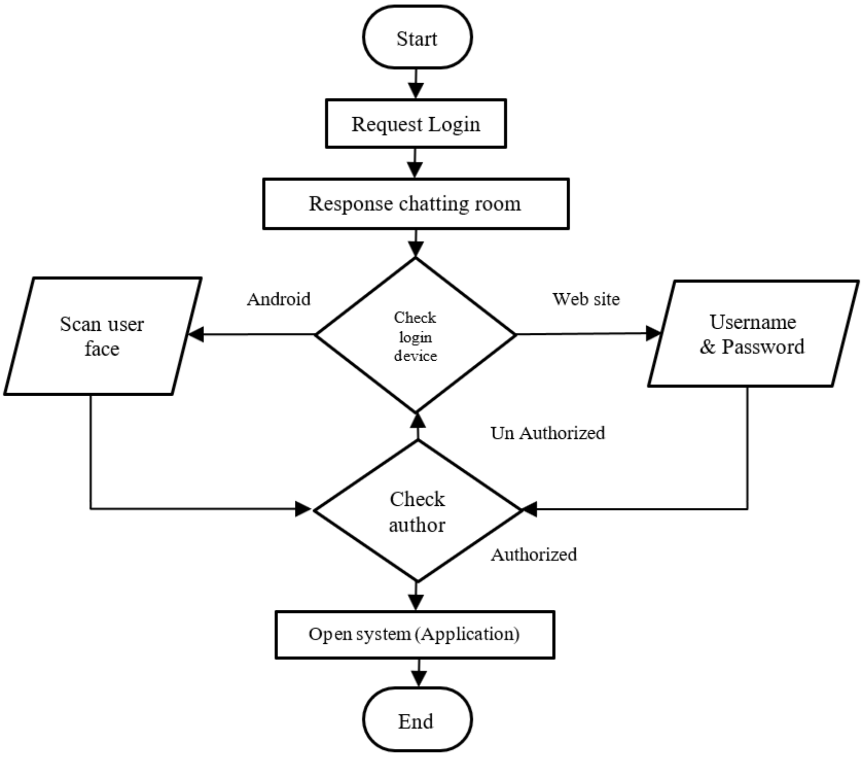


Figure 6. The login flowchart

In the recommender use case, when the user wants to ask for item recommendation, the user first makes a service request. The system then provides a chat form in which the user requests a recommendation. After that, the system analyzes and understands the item and recommends that it be displayed if it is available on the system; if it is not available on the system, an error message is displayed.

In the get weather use case, when the user wants to ask for the weather, the system then performs analysis and understanding of the user's request before checking the weather condition. Finally, the weather is displayed if the weather condition is available. If not available, a system error message return. The appliance control use case is when the user wants to ask for application control (e.g., light, fans, air-condition, doors, curtains, water flow calculate, calculate electricity consumption, and control TV). First, the system generates the chat form, user asks for the application control. After that, system performs application control and switch application on/off. If its available, system displays the item. If not, a system error message return.

4. CONCLUSION

To operate home appliances, the majority of present home automation systems follow a series of procedures or commands. To overcome the constraints of previous automation systems (such as the necessity to connect the system to the internet and construct an AI recommendation system), web and android applications based on voice commands have been utilized to make controlling household appliances more flexible by interpreting these commands into reality using Arduino and Raspberry Pi boards. In addition, a recommendation system has been developed to predict user's behavior based on the data saved in the dataset. Future works include addiding more home appliances with with approparite ML and AI algorithms.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Research Group Project.

REFERENCES

[1] S. Kumar, B. Kumar, K. Sharma, R. Raj, and S. Kumar, "IoT Based Secured Home Automation System Using NLP," *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, 2021, pp. 1-5, doi: 10.1109/ICAECA52838.2021.9675570.

[2] S. Kumar, S. Benedict, and S. Ajith, "Application of Natural Language Processing and IoT Cloud in Smart Homes," *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, 2019, pp. 20-25, doi: 10.1109/ICCT46177.2019.8969066.

[3] R. Boorugu and G. Ramesh, "A Survey on NLP based Text Summarization for Summarizing Product Reviews," *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020, pp. 352-356, doi: 10.1109/ICIRCA48905.2020.9183355.

[4] N. Chumuang *et al*., "Development a home electrical equipment control device via voice commands for elderly assistance," *2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 2020, pp. 1-7, doi: 10.1109/iSAI-NLP51646.2020.9376818.

[5] M. K. Singh, R. D, D. Pandian, and A. Sadana, "Surface Automation - Interacting with Applications using Black Box Approach," *2021 6th International Conference for Convergence in Technology (I2CT)*, 2021, pp. 1-6, doi: 10.1109/I2CT51068.2021.9417891.

[6] M. Marakana, A. Parmar, and P. Shah, "A Smart Android Application with Machine Learning Extension to Operate Computer and IoT Devices," *2021 2nd International Conference for Emerging Technology (INCET)*, 2021, pp. 1-5, doi: 10.1109/INCET51464.2021.9456382.

[7] U. Paudel, A. Dolan, S. Majumdar, and I. Ray, "Context-Aware IoT Device Functionality Extraction from Specifications for Ensuring Consumer Security," *2021 IEEE Conference on Communications and Network Security (CNS)*, 2021, pp. 155-163, doi: 10.1109/CNS53000.2021.9705050.

[8] P. Cichosz, *Data Mining Algorithms: Explained Using R*, First Edition. NY, USA: John Wiley and Sons, 2015.

[9] M. Shahroz, M. F. Mushtaq, M. Ahmad, S. Ullah, A. Mehmood, and G. S. Choi "IoT-Based Smart Shopping Cart Using Radio Frequency Identification", *IEEE Access*, vol. 8, pp. 68426-68438, 2020, doi: 10.1109/ACCESS.2020.2986681.

[10] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015, doi: 10.1109/COMST.2015.2444095.

[11] A. Kamilaris and A. Pitsillides, "Mobile Phone Computing and the Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 885-898, 2016, doi: 10.1109/JIOT.2016.2600569.

[12] M. A. Razzaque, M. M. -Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A Survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70-95, 2016, doi: 10.1109/JIOT.2015.2498900.

[13] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182-8201, 2019, doi: 10.1109/JIOT.2019.2935189.

[14] L. Chettri and Bera "A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16-32, 2020, doi: 10.1109/JIOT.2019.2948888.

[15] M. A. Torad and Y. H. Hossamel-din, "Smart ambulance using IoT for blood transfer facilities," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no.1, pp. 97-103, 2021, doi: 10.11591/ijeecs.v22.i1.pp97-103.

[16] M. A. Torad and M. A. Salam, "Smart helmet using internet of things," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 10, no. 2, pp. 90-98, 2021, doi: 10.11591/IJRES.V10.I2.PP90-98.

[17] S. Dey, T. Kundu, S. Mukherjee, and M. Sarkar, "Web based real-time home automation and security system," *International Journal of Electrical and Electronic Engineering and Telecommunications*, vol. 4, no. 3, pp. 1-8, Jul. 2015. [Online]. Available: http://www.ijeetc.com/uploadfile/2017/0731/20170731015905615.pdf

[18] N. David, A. Chima, and A. Ugochukwu, "Design of a Home Automation System Using Arduino," *International Journal of Scientific and Engineering Research*, vol. 6, no. 6, pp. 795-803, 2015. [Online]. Available: https://www.ijser.org/researchpaper/Design-of-a-Home-Automation-System-Using-Arduino.pdf

[19] M. Kumar, Shimi S. L., "Voice Recognition Based Home Automation System for Paralyzed People – A Review," *International Journal of Engineering Research and General Science*, vol 3, no. 5, pp. 671-673, 2015. [Online]. Available: http://pnrsolution.org/Datacenter/Vol3/Issue5/83.pdf

[20] A. Satria, M. L. Priadi, L. A. Wulandhari, and W. Budiharto, "The Framework of Home Remote Automation System Based on Smartphone," *International Journal of Smart Home*, vol. 9, no.1, pp. 53-60, 2015, doi: 10.14257/ijsh.2015.9.1.06.

[21] S. Sen, S. Chakrabarty, R. Toshniwal, and A. Bhaumik, "Design of an Intelligent Voice Controlled Home Automation System," *International Journal of Computer Applications*, vol. 121, no. 15, pp. 39-42, 2015, doi: 10.5120/21619-4904.

[22] T. Erić, S. Ivanović, S. Milivojša, M. Matić, and N. Smiljković, "Voice control for smart home automation: Evaluation of approaches and possible architectures," *2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 2017, pp. 140-142, doi: 10.1109/ICCE-Berlin.2017.8210613.

[23] C. J. Baby, F. A. Khan, and J. N. Swathi, "Home automation using IoT and a chatbot using natural language processing," *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2017, pp. 1-6, doi: 10.1109/IPACT.2017.8245185.

[24] S. A. I. Quadri and P. Sathish, "IoT based home automation and surveillance system," *International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2017, pp. 861- 866, doi: 10.1109/ICCONS.2017.8250586.

[25] M. Yousef and M. A. Torad, "A Treatise on Conversational AI Agents: Learning from humans' behaviour as a design outlook," *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 2019, pp. 1-4, doi: 10.1109/ICECTA48151.2019.8959585.

[26] G. Biagetti, P. Crippa, L. Fallaschetti, S. Orcioni, and C. Turchetti, "Distributed speech and speaker identification system for personalized domotic control," *Mobile Networks for Biometric Data Analysis, Lecture Notes in Electrical Engineering*, 2016, pp. 159-170, doi: 10.1007/978-3-319-39700-9_13.

BIOGRAPHIES OF AUTHORS

Mohamed A. Torad received the B.Sc. degree in electrical engineering from Higher Technological Institute (HTI), in 2007. From 2008, he has been a research assistant at communication and electronics department, Higher Technological Institute, received his M.D. from Ain Shams University (ASU) in 2013. and received his PhD degree from Ain Shams University (ASU) of in 2016. He still working at communication and electronics department from 2007 till now, working at Future University at Egypt (FUE) and supervises number of graduation projects at Culture and Science City. Work as reviewer at many conferences (e.g., International Conference on Microelectronics (ICM), IEEE International Multi-Conference on Systems, Signals & Devices). He can be contacted at email: mohamed.torad@hti.edu.eg.

Belgacem Bouallegue received his B.Sc. and M.Sc. degrees from University of Monastir, Tunisia, and Ph.D. from Graduate School of Engineering Science and Technology, University of Southern Brittany in in Lorient, France with the cooperation of University of Monastir, Tunisia. He is currently an Assistant Professor in Department of Computer Engineering at College of Computer Science, King Khalid University, Saudi Arabia. His research interests include Integrated System Design, Fault Tolerance, HW/SW Co-design, Parallel Computers, Embedded Systems and IoT, Network on Chip NoC, AI, IPs and MPSoCs, Machine Learning, Deep Learning, Wireless Sensor Networks Security, and Cryptography. He is working in collaboration with Lab-STICC Laboratory, Lorient, France and LIP6, Computer Science Research Laboratory, PARIS Cedex 05, France. He can be contacted at email: belgacem.bouallegue2015@gmail.com.

Abdelmoty M. Ahmed received his B.Sc., M.Sc.and PhD degrees from Systems and Computers Engineering , Faculty of Engineering, Al- Azhar University in Cairo, Egypt, He is currently His research interests include Digital image processing, Artificial intelligent, pattern recognition, Human Computer Interaction, Computer Graphics, machine learning ,Deep Learning , E-Learning, Intelligence Systems Engineering , Computer Vision and IOT systems., he is senior lecturer in computer engineering department at College of Computer Science, King Khalid University, Abha, Saudi Arabia, He is also interested in researching the technical fields that serve deaf and dumb and also works in the automatic translation of the Arabic Sign Language. he is having 20 years of teaching and research experience at various reputed Universities of Egypt and Saudi Arabia. His Ph.D thesis focused on the automatic translation of the Arabic Sign Language. He has published more research articles in reputed SCI and scopus indexed journals and conferences. He can be contacted at email: abd2005moty@yahoo.com and amoate@kku.edu.sa.