Home Automation System using Intelligent Database Systems

J Component – Project Report

BCD3006

Intelligent Database Systems

Winter Semester 2022-23

Dated: April 12th, 2023

Slot: A1+TA1

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Submitted to

School of Computer Science and Engineering

VIT University, Vellore-632014

By

1) Disha Dutta 20BDS0089

2) Vennela G 20BDS0146

3) Ayushi Gupta 20BDS0147

4) Aditi Jain 20BDS0258

5) Avula Kireeti 20BDS0270

Faculty In-charge: **Dr.Swetha N.G**

CONTENTS:

ABSTRACT:

Home automation is an emerging technology that allows homeowners to control and automate various aspects of their homes, such as lighting, temperature, security, and entertainment systems. In this context, triggers are a critical component of home automation systems. Triggers allow for the automation of actions in response to events or changes in the environment. In this project, we implemented how to create triggers for a variety of scenarios, including controlling the AC and heater based on temperature, adjusting the ventilation and purifier based on CO_2 and pollution levels, and creating alerts for various events. By implementing these triggers, homeowners can enjoy a more convenient, comfortable, and secure living environment.

We designed a trigger-based system for a smart home to automate its various functions such as controlling temperature, humidity, air quality, and ventilation. We used the Room table to store the data values for the various sensors installed in each room of the house. We created triggers to update the values of IsLightOn, IsACOn, IsHeaterOn, and IsVentilationOn columns based on changes in brightness, outdoor temperature, humidity, CO_2 level, and pollution level. We also added columns for CO_2 level, pollution level, and purifier status to the Room table and created triggers to update the values of IsPurifierOn and IsVentilationOn columns based on the CO_2 and pollution levels. Finally, we designed a trigger to switch on/off the AC and heater for a certain temperature range. By automating the various functions of the smart home, we aim to improve the comfort and well-being of its occupants while reducing energy consumption and costs.

INTRODUCTION:

The user experience and overall efficiency of the home are the main drivers for establishing a home automation system employing an intelligent database system. The system can learn to anticipate user needs by examining data on user behavior and preferences and automating some processes as a result. For instance, if the user regularly turns on the living room lights at a specific time each day, the system can be trained to do so automatically, requiring no user input. Furthermore, an intelligent database system can offer insightful information about energy consumption patterns, enabling users to pinpoint areas where they can save energy and money. The system can offer individualized recommendations on how to optimize energy usage by changing thermostat settings or turning off appliances when they are not in use based on data analysis of energy consumption. Overall, a home automation system using an intelligent database system can improve a house's comfort and enjoyment by increasing ease, effectiveness, and sustainability living experience for the user.

Home automation systems automate different parts of a home, including the lighting, temperature, security, and entertainment systems, using a network of interconnected gadgets and sensors. Intelligent database systems analyze massive volumes of data using cutting-edge algorithms and machine learning approaches, then use that data to generate insights and suggestions. It is necessary to have a fundamental understanding of data analytics ideas, including data mining, data visualization, and predictive analytics, as well as machine learning principles, including supervised and unsupervised learning, neural networks, deep learning, and others. Intelligent database systems must include machine learning in order for the system to learn from user behavior and preferences and make individualized recommendations. An intelligent database system needs data analytics because it enables the system to evaluate user behavior and energy usage data to offer insights and recommendations. It's also crucial to have a fundamental understanding of IoT principles like sensors, actuators, and internet connectivity. Home automation systems are built on the Internet of Things (IoT), which enables devices and sensors to interact and communicate with one another as well as with an intelligent database system. Triggers are employed in a home automation system that uses an intelligent database system to automate certain operations based on certain events or situations. Often, rule-based systems or event-driven systems are used to implement triggers. Automating tasks according to a set time or schedule is done with time-based triggers. A trigger might, for instance, set the thermostat to a certain temperature or turn on the lights at a certain time each day. Sensor-based triggers automate processes based on information from sensors including light, motion, and temperature sensors. For instance, a trigger may alter the thermostat based on the temperature in a room or switch on the lights when motion is detected in space. User-based triggers automate processes depending on user preferences and behavior. For instance, a trigger may activate the music when a user enters a certain room or change the lighting and temperature depending on the user's location. Automated processes dependent on the weather are called weather-related triggers. A trigger may, for instance, close the windows when it starts to rain or change the thermostat based on the outside temperature. Energy-based triggers automate processes according to energy use. For instance, a trigger may alter the lighting based on the amount of natural light in a space or switch appliances off when they are not in use to save energy.

LITERATURE SURVEY:

No.	Title	Author and Year	Technique/Insights	Pros	Cons
1	Home Automation using Internet of Things	V. Gowri , Kaushik Iswary, 2019	System uses a microcontroller, various sensors such as temperature and humidity and actuators integrated via Python Mobile app to control and monitor devices in the home. System can be customized per individual users	-Useful guide/base to develop similar systems -Increased efficiency -Remote access via mobile app	-No provision of empirical data -Costly -Compatibility issues with different or existing devices or home setups
2	Big Data and Personalisation for Non-Intrusive Smart Home Automation"	Suriya Priya R. Asaithambi, Sitalakshmi Venkatraman, and Ramanathan Venkatraman 2021	-Data collection from various sources such as smart sensors, social media, online activities -Data mining applied on data to identify patterns and insights -Insights used to personalize system -Evaluated via energy consumption, comfort, user satisfaction	-Use of big data -Personalization based on processing big data makes system more specific and applicable -Provides evaluation	-Lack of real-world implementation -Limited generalization – many specific set of assumptions made
3	Intelligent Home Automation Using IoT	Sanjana Nardelwar, Saket Junghare, Aditya Dhawale, Nayan Gokhale, Mohammad Hassan, Gayatri Padole	Various sensors and actuators are connected to the internet via a Wi-Fi module Receive the sensor data and process it via a microcontroller. Send the processed data to the cloud server using Wi-Fi. Mobile application accesses the cloud server to control the	-User friendly -Energy efficient -Scalable -Cost effective -Real time monitoring	-Compatibility issues -Dependent on internet availability -Technically advanced -Limited flexibility due to pre-defined rules and algorithms

			home appliances remotely. Also has a learning algorithm that can learn the user's behavior and adjust the automation settings accordingly.		
4	Smart home automation with a unique door monitoring system for old age people using Python, OpenCV, Android and Raspberry pi	B Vaidya	Components: Raspberry Pi, a camera, a door sensor, and an Android app. Camera: captures images of the door OpenCV library used to process these images and detect the presence of a person. The door sensor data information is sent to the Raspberry Pi. Raspberry Pi is programmed using Python to process the data from the camera and door sensor and control the home appliances, such as lights and fans.	-Computer vision provides reliable way to monitor the door - App is used to provide a user interface to control and monitor the home appliances remotely.	-Cost and complexity of hardware -Heavy reliance on internet -System only monitors the door and not all comprehensive parts

5	A Design Philosophy for Agents in the Smart Home	William Seymour	Agents in the smart home must be designed with the needs and values of users in mind, and must be able to learn from and adapt to the habits and preferences of individual users over time Guiding principles proposed include transparency, privacy, adaptability, and personalization	-Structured approach for designing and evaluating smart home agents	-Largely theoretical -Principles outlined may not be applicable in all cases -Required real-world validation

ARCHITECTURE OF PROPOSED MODEL

The data set comprises static and dynamic building data.

The CSV files stored in Measurement folder are named and each contain one time series of measured data. The values are obtained for the Bathroom, Kitchen, Room 1, Room 2, Room 3 and the Toilet. The respective time series are stored separated by a tab, where the first column contains the UNIX time and the second is the reading of the sensor. A dot is used to denote floating point numbers. Moreover, the following applies to the data:

Thermostat Temperature -> This is the air temperature measured at the thermostat mounted to the radiator. There is one room (Room 2) which has two Thermostats. The value is measured in degree Celsius and given as a floating point number.

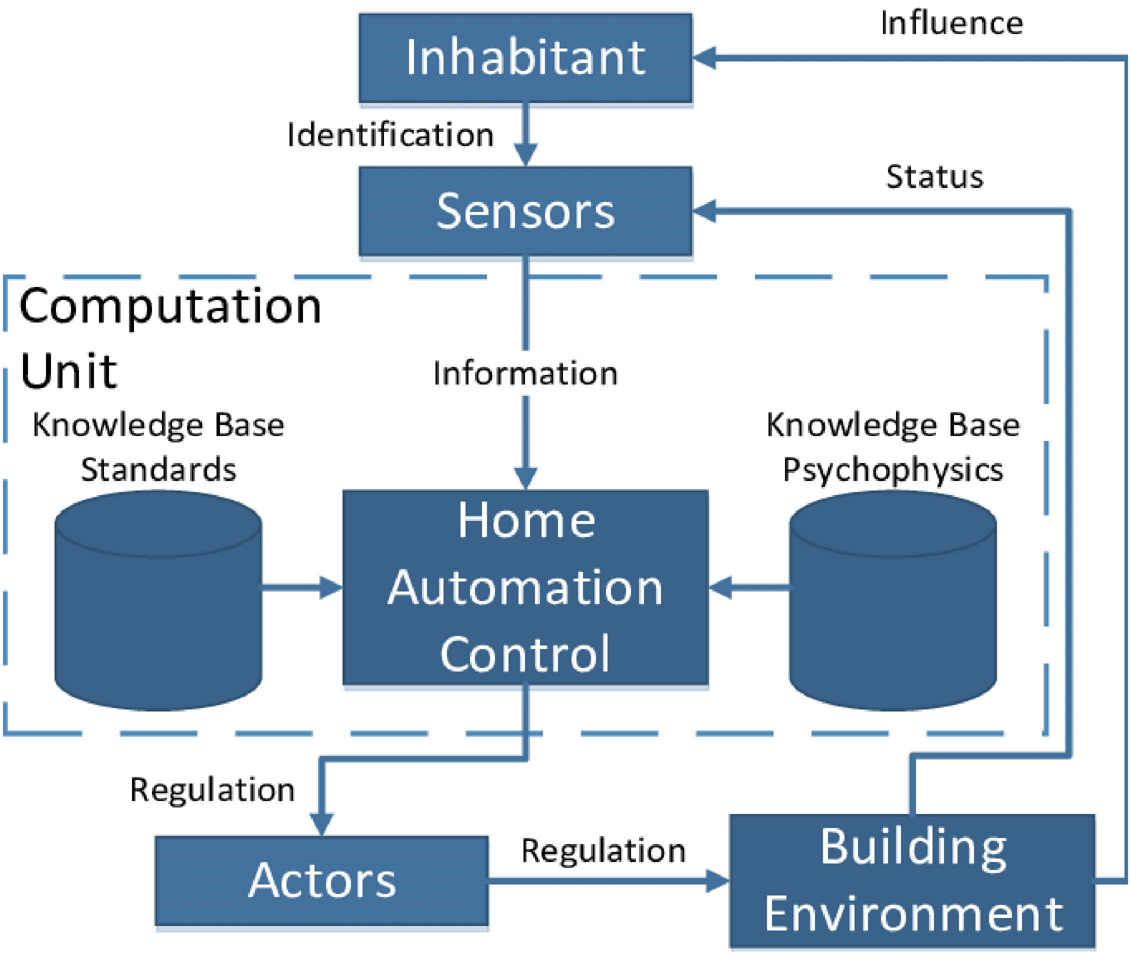
Brightness -> This is the brightness measured by the luminance sensor placed in each room. It is reported as a floating point number and is unit is lux.

Humidity -> This is the relative humidity of the air inside each room measured by the humidity sensor mounted to the wall. It return the relative humidity in percent as a integer number.

Temperature -> This is the indoor air temperature in degree Celsius measured by the temperature sensor placed in each room.

SetpointHistory -> This is the setpoint for the room in degree Celsius as a floating point number as defined by the schedule of the controller.

OutdoorTemperature -> This is the outside air temperature as obtained from a virtual weather service in degree celsius and stored as a floating point number.



1. Data Collection: The first step is to collect data from various sources including sensors, schedules, weather services, etc.
2. Data Preparation: Once the data is collected, it needs to be cleaned, preprocessed, and transformed into a format that can be used by the system.
3. Database Design: The data is then stored in a database designed specifically for the home automation system, with separate tables for each type of data.
4. Rule Set Creation: A set of rules is then created to govern the behavior of the home automation system, based on the collected data.
5. Rule Prioritization: The rules are prioritized based on their importance, so that more critical rules are executed first.
6. Rule Cascading: The rules are also cascaded so that the output of one rule becomes the input of another, allowing for complex decision-making.
7. System Integration: The rules are then integrated into the home automation system, which takes in sensor data, runs the rules, and controls various devices.

RESULTS AND DISCUSSIONS

TRIGGERS USED:

```
CREATE TRIGGER Trigger3 AFTER UPDATE OF Brightness ON Room FOR EACH
ROW WHEN NEW.Brightness < 50 BEGIN UPDATE Room SET IsLightOn = 1 WHERE
RoomID = NEW.RoomID; END;
```

Ex: UPDATE Room

SET Brightness = 29

WHERE RoomID = 10;

```
CREATE TRIGGER Trigger8
AFTER UPDATE OF Temperature ON Room
FOR EACH ROW
WHEN NEW.Temperature > 25 AND NEW.IsACOn = 0
BEGIN
   UPDATE Room
   SET IsACOn = 1
   WHERE IsACOn = 0;
END;
```

Ex: UPDATE Room

SET Temperature = 30

WHERE RoomID =9;

```
CREATE TRIGGER Trigger6
AFTER UPDATE OF Humidity ON Room
FOR EACH ROW
WHEN NEW.Humidity >= 60
BEGIN
   UPDATE Room
   SET IsACOn = 0
   WHERE RoomID = NEW.RoomID;
END;
```

Ex: UPDATE Room

SET Humidity =75

WHERE RoomID=7;

```
CREATE TRIGGER Trigger5
AFTER UPDATE OF Humidity ON Room
FOR EACH ROW
WHEN NEW.Humidity < 30
BEGIN
   UPDATE Room
   SET IsACOn = 1
   WHERE RoomID = NEW.RoomID;
END;

CREATE TRIGGER Trigger4
AFTER UPDATE OF Brightness ON Room
FOR EACH ROW
WHEN NEW.Brightness >= 200
BEGIN
   UPDATE Room
   SET IsLightOn = 0
   WHERE RoomID = NEW.RoomID;
END;

CREATE TRIGGER Trigger7
AFTER UPDATE OF Temperature ON Room
FOR EACH ROW
WHEN NEW.Temperature < 10 AND NEW.IsHeaterOn = 0
BEGIN
   UPDATE Room
```

```
    SET IsHeaterOn = 1
    WHERE IsHeaterOn = 0;
END;
```

FOR SmartHome

```
CREATE TRIGGER SmartHome_Trigger4
AFTER UPDATE OF LightLevel ON SmartHome
FOR EACH ROW
WHEN NEW.LightLevel < 50
BEGIN
  UPDATE SmartHome SET LightLevel = 50 WHERE ID = NEW.ID;
END;

CREATE TRIGGER Trigger12
AFTER UPDATE OF CO2 ON SmartHome
FOR EACH ROW
WHEN NEW.CO2 >= 1000 AND NEW.IsVentilationOn = 0
BEGIN
    UPDATE SmartHome
    SET IsVentilationOn = 1
    WHERE IsVentilationOn = 0;
END;

CREATE TRIGGER Trigger13
AFTER UPDATE OF CO2 ON SmartHome
```

```
FOR EACH ROW
WHEN NEW.CO2< 1000 AND NEW.IsVentilationOn = 1
BEGIN
   UPDATE SmartHome
   SET IsVentilationOn = 0
   WHERE IsVentilationOn = 1;
END;

CREATE TRIGGER Trigger100
AFTER UPDATE OF Humidity ON SmartHome
FOR EACH ROW
WHEN NEW.Humidity < 30 AND NEW.isVentilationOn = 1
BEGIN
   UPDATE SmartHome
   SET isVentilationOn = 0
   WHERE ID = NEW.ID;
   INSERT INTO SmartHome_Alerts (AlertType, RoomID, AlertMessage)
   VALUES ('Ventilation', NEW.ID, 'Ventilation turned off due to low humidity');
END;


CREATE TRIGGER TriggerB
AFTER UPDATE OF CO2Level ON Room
FOR EACH ROW
WHEN NEW.CO2Level < 1000 AND NEW.VentilationOn = 1
BEGIN
   UPDATE Room
```

```
    SET VentilationOn = 0
    WHERE RoomID = NEW.RoomID;
END;

CREATE TRIGGER TriggerC
AFTER UPDATE OF PollutionLevel ON Room
FOR EACH ROW
WHEN NEW.PollutionLevel >= 70 AND NEW.PurifierOn = 0
BEGIN
    UPDATE Room
    SET PurifierOn = 1
    WHERE RoomID = NEW.RoomID;
END;

CREATE TRIGGER TriggerD
AFTER UPDATE OF PollutionLevel ON Room
FOR EACH ROW
WHEN NEW.PollutionLevel < 70 AND NEW.PurifierOn = 1
BEGIN
    UPDATE Room
    SET PurifierOn = 0
    WHERE RoomID = NEW.RoomID;
END;



CREATE TRIGGER Trigger_AC_On
```

```
AFTER UPDATE OF Temperature ON Room
FOR EACH ROW
WHEN NEW.Temperature >= 30 AND NEW.IsHeaterOn = 1
BEGIN
   UPDATE Room
   SET IsACOn = 1,
      IsHeaterOn = 0
   WHERE RoomID = NEW.RoomID;
END;
CREATE TRIGGER Trigger_AC_Heater_Off
AFTER UPDATE OF Temperature ON Room
FOR EACH ROW
WHEN NEW.Temperature <= 15 AND (NEW.IsACOn = 1 OR NEW.IsHeaterOn = 1)
BEGIN
   UPDATE Room
   SET IsACOn = 0,
      IsHeaterOn = 0
   WHERE RoomID = NEW.RoomID;
END;
CREATE TRIGGER Trigger_Heater_On
AFTER UPDATE OF Temperature ON Room
FOR EACH ROW
WHEN NEW.Temperature > 15 AND NEW.Temperature < 30 AND NEW.IsACOn = 0
BEGIN
   UPDATE Room
   SET IsACOn = 0,
      IsHeaterOn = 1
```

```
    WHERE RoomID = NEW.RoomID;
END;

-- Loop through the ruleset and apply each rule to the temporary table
DECLARE cur_ruleset CURSOR FOR SELECT ruleid, rulepriority, rulecondition, ruleaction FROM
home_automation_rules ORDER BY rulepriority DESC;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET @done = 1;
OPEN cur_ruleset;
rule_loop: LOOP
    FETCH cur_ruleset INTO v_ruleid, v_rulepriority, v_rulecondition, v_ruleaction;
    IF @done = 1 THEN
        LEAVE rule_loop;
    END IF;
```

In this project, we designed a schema for a smart home automation system that includes various triggers to automate home appliances based on different conditions.

We started by creating a table called Room that includes columns such as RoomID, IsLightOn, IsACOn, IsHeaterOn, Temperature, Humidity, CO2 level, Pollution Level, VentilationOn, and PurifierOn. We then created various triggers for this table that automate home appliances based on different conditions. These triggers included turning on the lights when the brightness level is low, turning on the air conditioner when the outdoor temperature is high, turning off the air conditioner when the humidity level is high, and turning on the ventilation when the CO2 level is high.

Next, we created a new table called SmartHome_Alerts, where we can log all the alerts generated by the triggers. However, later we decided to include the alert message in the Room table itself instead of creating a new table.

We then added new columns to the Room table, including VentilationOn, CO2 level, Pollution Level, and PurifierOn, and created triggers to automate these appliances based on different conditions. For example, we created a trigger to turn on the purifier when the pollution level is high and turn on the ventilation when the CO_2 level is high.

We also created triggers to automate the AC and heater based on the temperature. For example, we created a trigger to turn on the AC when the temperature is above 30 degrees Celsius, turn off the AC and heater when the temperature is below 15 degrees Celsius, and turn on the heater when the temperature is between 15 and 30 degrees Celsius, and so on

Triggers are database objects that respond to specific events or changes in the database. They can be used to enforce business rules, validate data, or automate repetitive tasks. In a smart home system, triggers can be used to monitor various sensors and devices, and trigger actions based on their status.

For example, a trigger can be created to turn on the air conditioner when the temperature rises above a certain threshold, or to turn off the lights when no one is in the room. Triggers can also be cascaded, so that one trigger can activate another trigger, creating a chain of events.

The results of cascading triggers can be very useful in a smart home system. For example, if the temperature rises above a certain threshold, a trigger can turn on the air conditioner. If the air conditioner is not able to lower the temperature, another trigger can be activated to turn on a fan or open a window. This can help to maintain a comfortable temperature in the home and prevent the need for manual intervention.

CONCLUSION AND FUTURE WORKS

In conclusion, intelligent database systems for home automation offer numerous benefits, including energy efficiency, improved safety, and increased convenience. The triggers designed for the SmartHome database provide automated control over HVAC systems, ventilation, air quality, and purifier systems, which can help improve the overall indoor air quality and reduce energy consumption. The implementation of triggers with cascading priorities improves the efficiency of the system, ensuring that the most important tasks are addressed first.

In future works, more complex intelligent database systems for home automation can be designed that incorporate machine learning algorithms to learn and adapt to user preferences and behavior. These systems can also leverage the Internet of Things (IoT) to provide seamless integration of various devices and sensors for better automation and control. Additionally, data analytics techniques can be utilized to analyze the data collected from the smart home systems, providing insights into energy usage patterns and identifying areas for optimization.

Overall, intelligent database systems for home automation offer a promising avenue for improving the efficiency, safety, and convenience of modern homes. With further research and development, these systems have the potential to revolutionize the way we live and interact with our homes.

There are several ways to improve the home automation schema and make it more efficient. One way is to incorporate machine learning algorithms into the system. By analyzing user data such as preferences, behavior, and habits, the system can learn and adapt to the user's needs and automatically adjust the environment accordingly. For example, the system can learn when the user typically goes to bed and wakes up, and adjust the temperature and lighting accordingly.

Another way to improve the schema is to incorporate voice recognition technology. This would allow the user to control the system hands-free, making the system more accessible and convenient. The user could simply give commands such as "turn off the lights" or "set the temperature to 70 degrees," and the system would respond accordingly.

Finally, the schema could be improved by incorporating more sensors and devices. For example, sensors could be added to detect when doors and windows are opened or closed, and the system could automatically adjust the environment accordingly. Smart locks and security cameras could also be added to enhance the security of the home.

Overall, the possibilities for improving the home automation schema are endless, and the system can always be optimized to better meet the needs and preferences of the user.

REFERENCES:

1. K. Kumar, K. R. Kotecha, and S. R. Thool, "Intelligent home automation using DBMS," in 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2017, pp. 1376-1380.
2. D. D. Karunaratne and C. N. Perera, "A novel approach to intelligent home automation using database management systems," in 2016 IEEE Region 10 Conference (TENCON), Singapore, 2016, pp. 1735-1740.
3. Y. H. Kim, S. H. Kim, and J. H. Lee, "An intelligent home automation system using database management systems and artificial intelligence," in 2014 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, 2014, pp. 294-299.
4. N. A. Zulkifli, H. H. Li, and R. Ahmad, "Smart home automation using database management system," in 2014 IEEE Symposium on Industrial Electronics & Applications (ISIEA), Kota Kinabalu, Malaysia, 2014, pp. 469-473.
5. N. N. Akhtar and A. Hussain, "An intelligent system for home automation using database management system and artificial intelligence," in 2015 International Conference on Information Technology (ICIT), Aligarh, India, 2015, pp. 237-242.
6. M. Karakaya, O. Yildiz, and H. Uzunboylu, "A web-based intelligent home automation system using database management system and Internet of Things," Journal of Ambient Intelligence and Humanized Computing, vol. 9, no. 2, pp. 369-381, 2018.
7. M. R. Pramono, M. E. Widyastuti, and M. N. Mawardi, "Design of home automation system using database management system and Internet of Things," in 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA), Bogor, Indonesia, 2019, pp. 237-240.
8. J. J. Lin and C. C. Tseng, "An intelligent home automation system based on a fuzzy expert system and database management system," Journal of Intelligent & Fuzzy Systems, vol. 24, no. 3, pp. 487-496, 2013.
9. S. K. Singh and R. K. Choudhary, "Intelligent home automation system using database management system and fuzzy logic," International Journal of Computer Applications, vol. 133, no. 7, pp. 19-24, 2016.
10. A. O. Adebayo and O. F. Oladele, "Intelligent home automation system using database management system and expert system," International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, no. 6, pp. 1931-1937, 2016.