Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P); 2583-7907 (E)
Vol.14 No.2, 2025, pp.43-50

www.crijournals.org
DOI: https://doi.org/10.70112/ajcst-2025.14.2.4387

Development of a Low-Cost IoT-Enabled Door Access Control System for Smart Home Security

Monday Fredrick Ohemu[1*], Moses Oluwabukunmi Idoko[2]
[1*,2]Department of Electrical and Electronics Engineering, Air Force Institute of Technology, Kaduna, Nigeria
E-mail: minamidoko625@gmail.com
*Corresponding Author: monfavour@gmail.com
(Received 6 November 2025; Revised 15 November 2025; Accepted 30 November 2025; Available online 5 December 2025)

Abstract **- Traditional door security systems are often vulnerable to tampering and lack modern capabilities such as remote access and real-time monitoring. This project aimed to design and implement a cost-effective, Internet of Things (IoT)-based door security system that integrates proximity detection, live camera feed, and remote access control to address these limitations. The system was built using an ESP32-CAM microcontroller for Wi-Fi connectivity and video streaming, an HC-SR04 ultrasonic sensor for proximity detection, and a servo motor for the locking mechanism. Additionally, a custom mobile application was developed using MIT App Inventor to serve as the user interface. The prototype successfully demonstrated functional reliability, enabling real-time monitoring via live video feed, remote actuation of the door lock, and visual feedback through an LED indicator. Consequently, this research provides a scalable, user-friendly solution that bridges the gap between advanced security features and affordability, demonstrating how low-cost, open-source technology can be effectively utilized to create robust IoT-based home security systems.**

Keywords: **IoT, Door Security, ESP32-CAM, Remote Access, MIT App Inventor**

I. INTRODUCTION

Traditional door security systems, such as mechanical locks and keys, have significant limitations, including vulnerability to tampering, lack of remote access, and inability to provide real-time alerts [1]. While electronic systems like keypad locks and biometric scanners offer improved security, they are often expensive, complex to install, and lack IoT integration. The emergence of IoT technology has revolutionized security systems by enabling remote monitoring, real-time notifications, and seamless integration with smart devices [2][3]. This study focuses on designing and implementing an IoT-based door security system that uses an ultrasonic sensor to detect proximity, activates a camera to provide a live feed, and allows remote access control via a mobile app. The system also includes a light indicator to show the owner's response.

The integration of IoT into security systems has opened up new possibilities for enhancing safety, convenience, and accessibility [4]. For instance, IoT-enabled systems can provide real-time alerts, remote access, and data analytics, which are not possible with conventional mechanical systems [5]. This project leverages these advancements to create an IoT-based door security system that is not only secure but also user-friendly and cost-effective.

The lack of a cost-effective, IoT-enabled door security system that combines proximity detection, real-time camera monitoring, and remote access control highlights the need for a solution like the proposed IoT-based door security system [6]. The system uses an ultrasonic sensor to detect when someone is at the door, activates a camera to provide a live feed to the owner's mobile app, and allows the owner to remotely open or close the door. A light indicator subsequently provides visual feedback on the owner's response. The inclusion of a camera and light indicator adds an extra layer of functionality, making the system more robust and user-friendly.

This study addresses critical gaps in modern door security systems by leveraging IoT technology to create a cost-effective, user-friendly, and scalable solution. The proposed system enhances security by allowing remote access, real-time camera monitoring, and instant notifications. By demonstrating how low-cost components and open-source tools can be used to create advanced IoT solutions, this project encourages innovation and the adoption of IoT technology in everyday life. Potentially, it also serves as a foundation for future research and development in IoT-based security systems, paving the way for more advanced and integrated solutions.

Abdulla [7] addresses the limitations of traditional security methods, which often lack real-time feedback, fail to trigger alarms during breaches, and are vulnerable to human errors like leaving doors unlocked. The system integrates biometric authentication (face recognition using deep neural networks), plate number recognition (via OpenCV), and movement detection (using PIR sensors and magnetic contact switches), with data monitoring and alerts managed through a cloud network using Node-RED and MQTT. Hardware includes an Arduino Mega 2560, a Raspberry Pi 4 for image processing, an ESP-01 module for Wi-Fi, a 12V DC solenoid door lock, an LED, and a buzzer. The system achieved a 77% overall accuracy, with face recognition reaching 100% accuracy in optimal lighting but dropping to 50% in low light and 75% with varying angles or expressions, while plate recognition attained 60% accuracy. Challenges include reduced

performance under poor lighting and with larger datasets, highlighting scalability issues in image processing and the need for robust algorithms or edge computing solutions. These findings underscore the potential of multi-modal IoT security systems while emphasizing the necessity for adaptive algorithms and efficient data management to ensure reliability in diverse, real-world conditions.

In 2021, Ranjana *et al.* [8] conducted a study on a smart doorbell system that effectively integrates ultrasonic sensors, microcontrollers, and mobile applications to enhance home security and user convenience. Their research emphasizes the critical role of efficient communication protocols, specifically MQTT, in enabling seamless data exchange between the system's components. By leveraging ultrasonic sensors for accurate visitor detection and microcontrollers for real-time processing, the system ensures reliable notifications and control through a mobile app interface. This work underscores the importance of robust, low-latency communication protocols like MQTT to support responsive and scalable IoT-based smart home solutions, offering valuable insights into optimizing system performance for real-time user interaction.

In 2021, Bulla *et al.* [9] addressed the demand for affordable protection for homes, warehouses, and offices by examining the high costs and maintenance issues of existing smart security systems. The study presents a cost-effective prototype utilizing an ESP32-CAM microcontroller paired with a PIR motion sensor (HC-SR501) and an OV2640 camera supporting 1600×1200 resolution at 15 FPS. The system employs a Telegram bot for communication, allowing users to receive photo alerts and issue remote commands (e.g., /start, /flash, /photo) via Wi-Fi and Bluetooth, with data storage on a microSD card. Aligned with Industrial IoT (IoT) standards, it incorporates low-level programming and MQTT protocols, while suggesting low-power technologies like LoRa and ZigBee. The adoption of open-source, budget-friendly platforms such as Arduino and ESP32-CAM highlights their role in enabling rapid prototyping and innovation in IoT security, making development accessible to a wider range of contributors.

A study by Komitov *et al.* [10] emphasized the role of Wi-Fi and cloud-based platforms in enabling real-time data exchange between IoT devices and mobile apps. These studies provide valuable insights into the design and implementation of the proposed system, ensuring efficient communication between the ultrasonic sensor, ESP32, and mobile app.

Research in 2024 by Satyanarayana et al. [11] presents significant advancements in enhancing home security through integrated IoT technologies. This system employs an ESP32-CAM microcontroller, an IR sensor for obstacle detection, an OV2640 camera for facial recognition, and a solenoid door lock, with a double loop antenna to ensure reliable signal performance. By leveraging Python programming and Telegram for remote notifications and control, the system achieves higher frequencies, lower return losses, and improved gain compared to other systems like IoT-SSS-DLA and FA-SDLS-MCM, demonstrating superior signal efficiency and reliability for secure access management.

In 2019, Kamala *et al.* [12] proposed an IoT-based intelligent doorbell system to enhance security for elderly residents and absent homeowners. Built on a Raspberry Pi 3 platform, the system triggers a camera to capture an image of the visitor whenever the doorbell is pressed, instantly transmitting the photo to the owner via SMS and email. A key innovation in their design is a text-to-speech response mechanism: the homeowner can reply via text, which the system converts into an audible message played through a speaker. The authors concluded that this approach provides a cost-effective and accessible solution for real-time remote visitor verification.

A study by Sonamoni *et al*. [13] offers a significant advancement by addressing the limitations of traditional door locks, such as vulnerability to misplaced keys, theft, and lack of remote control. This system integrates an ESP32-CAM microcontroller with an MIT App Inventor Android app (version 1.2), a DC 12V solenoid door lock, an FTDI 232 USB-to-Serial Interface Board, an 18650 lithium-ion battery, and a buzzer to enable robust remote monitoring and control via smartphone internet connectivity. The MIT App Inventor app interface, featuring an image gallery, “TAKE PICTURE,” “UNLOCK DOOR,” and “LOCK DOOR” buttons, alongside notification widgets, supports real-time user interaction and visitor verification through an IP camera. Its multi-modal alerting approach-combining smartphone notifications with a loud buzzer for community awareness-enhances the system's effectiveness in responding to potential security breaches. By providing remote verification, theft alerts, and automatic locking, this system demonstrates a user-centric convergence of security and convenience, making it a compelling solution for smart home applications.

In 2024, Sari *et al.* [14] presented a notable contribution by addressing the shortcomings of conventional door security systems, which rely on traditional keys that are prone to mismanagement and theft, thereby increasing vulnerability to crime. This study develops an IoT-based door security tool leveraging the ESP32-CAM microcontroller and the MIT App Inventor application to enable seamless remote control and monitoring via smartphones over an internet network. Testing validated the system's effectiveness in meeting room security requirements, demonstrating reliable performance in real-time monitoring and access management. The emphasis on simplicity and accessibility highlights the potential for widespread adoption of IoT security solutions driven by ease of implementation and intuitive user experience, catering effectively to the demands of consumers for practical security tools.

In 2021, Bakhory *et al.* [15] advanced the development of remotely controllable door lock systems by addressing the limitations of conventional and existing smart door systems

through enhanced remote authentication and monitoring capabilities via IoT and smartphone integration. The study employs an Arduino Wemos D1 and ESP8266 microcontroller paired with the MIT App Inventor application to enable seamless remote monitoring and control of a door lock from any location at any time. The research focuses on designing an IoT-based door lock system and evaluating the servo motor's durability to handle the load of a door deadbolt, confirming its reliability through load capacity tests. Compared to more recent works utilizing the ESP32-CAM, which integrates Wi-Fi, Bluetooth, and camera functionalities, this study's use of the Wemos D1 and ESP8266 reflects an earlier stage in the evolution of IoT hardware. This progression toward more powerful and integrated microcontrollers highlights the increasing complexity and functionality of IoT-based security solutions, enabling more sophisticated and cost-effective DIY and academic projects for smart home applications.

Karuppusamy [16] demonstrated the use of MIT App Inventor in developing IoT-enabled mobile apps for home automation and security systems. The study highlighted MIT App Inventor's ability to integrate sensors, cameras, and actuators seamlessly, making it an ideal choice for projects that require real-time monitoring and remote control. MIT App Inventor also supports push notifications, which can be used to alert users of unauthorized access attempts or other security events.

A. ESP32-CAM Module

The ESP32-CAM is a small, low-power camera module based on the ESP32 microcontroller. It comes with an OV2640 camera and provides an onboard TF card slot [17]. The ESP32 microcontroller has emerged as a popular choice for IoT projects due to its built-in Wi-Fi and Bluetooth capabilities, low power consumption, and cost-effectiveness, according to [18].

B. HC-SR04 Ultrasonic Sensor

An ultrasonic sensor is a device that functions to convert physical quantities (sound) into electrical quantities and vice versa. The way this sensor works is based on the principle of sound wave reflection, allowing it to interpret the presence (distance) of an object at a certain frequency [19]. The HC-SR04 ultrasonic sensor is widely used for proximity detection and object recognition in medium- to large-scale projects due to its accuracy, low cost, and ease of integration compared to other industry-standard models [20].

C. Mobile App Development with MITAppInventor

MIT App Inventor is a powerful IoT platform that simplifies the development of mobile apps for IoT systems. It provides a userfriendly interface for connecting IoT devices to mobile apps, enabling real-time data visualization, remote control, and notifications. MIT App Inventor is widely used in IoT projects due to its ease of use, flexibility, and support for multiple communication protocols such as MQTT and HTTP [21]. MIT App Inventor's drag-and-drop interface allows developers to create custom dashboards for monitoring and controlling IoT devices without extensive programming knowledge.

II. METHODOLOGY

A. Hardware Design

To achieve the intended outcome of this research, the system requirements, design criteria, and installation methods were carefully considered during the design and development of the device. Prior to assembling the various units, the device architecture was divided into subunits or modules, each of which was individually planned and evaluated. Figure 1 illustrates the different functional blocks of the system.

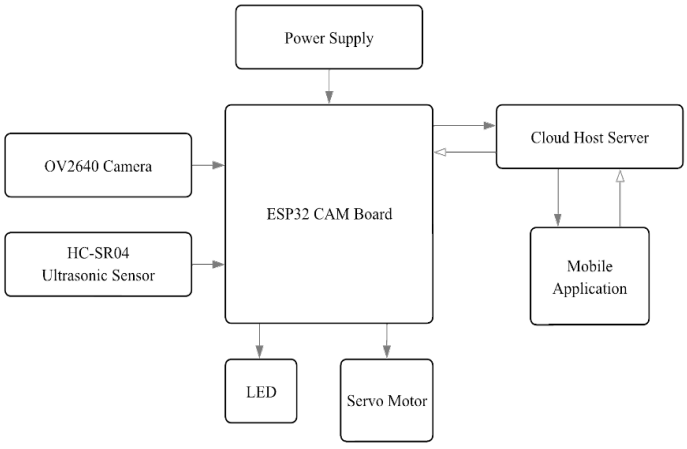

Fig.1 System Block Diagram

1. Power Supply Unit: The entire system is powered by two 3.7V lithium-ion batteries wired in series to provide a nominal output of 7.4V. To ensure a stable and safe power supply for the microcontroller and its peripherals, an LM2596 DC-DC buck converter was used. This converter efficiently steps down the battery voltage to the appropriate 5V required by the ESP32-CAM and other components, preventing potential damage from overvoltage while maintaining a consistent power flow.

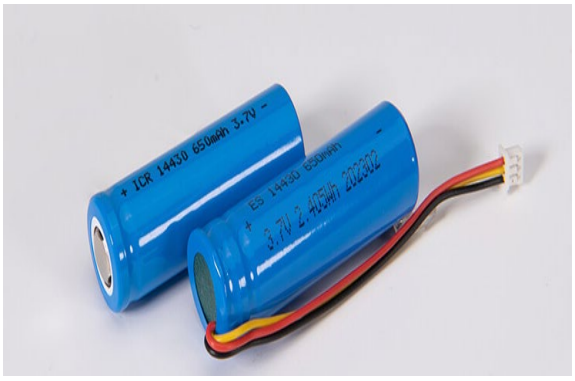

Fig.2 3.7V Lithium-Ion Batteries [22]

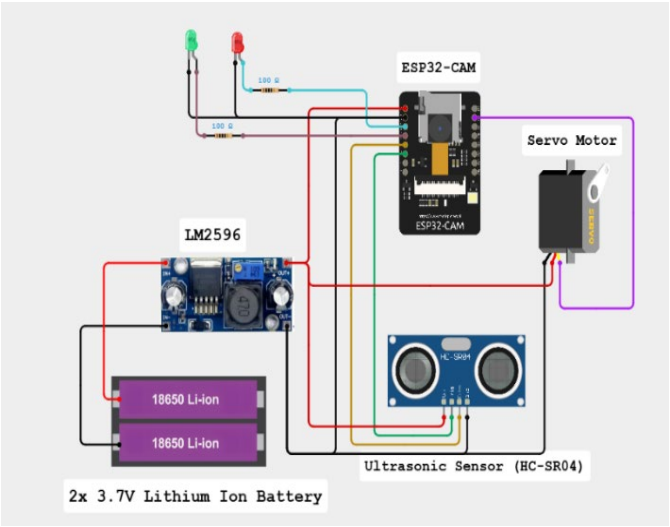

Fig.3 Circuit Pictorial Schematic

2. Circuit Design and Wiring: The hardware components were interfaced with the ESP32-CAM using a breadboard. The wiring was done carefully to avoid shorts and to ensure that the correct GPIO pins were used as defined in the software.

3. Power Supply: A dedicated external 5V power supply was used for the ESP32-CAM and the servo motor to ensure sufficient current, preventing voltage drops and unexpected resets. The HC-SR04 and LEDs were powered directly from the ESP32's 3.3V and 5V pins.

4. GPIO Pin Assignments: The HC-SR04 Trigger (TRIG) pin was connected to GPIO14, and the Echo (ECHO) pin to GPIO15. The servo motor's signal pin was connected to GPIO16. The red LED was connected to GPIO12 and the green LED to GPIO13 via current-limiting resistors to prevent damage.

5. Camera Configuration: The default AI-THINKER camera pinout was utilized, which is a standard configuration that simplifies the hardware setup.

B. Software Implementation

The software component of the project, developed using the Arduino IDE, is responsible for controlling the hardware, establishing communication channels, and managing the system's logic.

1. Development Environment: The development was partly performed using the Arduino IDE, a popular platform for programming microcontrollers. The ESP32 board support package was installed and configured, allowing the IDE to compile and upload code for the ESP32-CAM. The following libraries were installed and included in the sketch: esp_camera.h, WiFi.h, HTTP Client.h, soc/soc.h, soc/rtc_cntl_reg.h, and ESP32Servo.h. These libraries provided the necessary functions for camera control, Wi-Fi networking, HTTP communication, disabling the brownout detector, and controlling the servo motor, respectively. For the mobile application, the MIT App Inventor platform was utilized. This platform provided a web-based, block-based programming environment that simplified the development of a companion app for the system. Using MIT App Inventor, the user interface-including the video feed display and door control buttons-was created and configured to send and receive commands from the ESP32-CAM's web server.

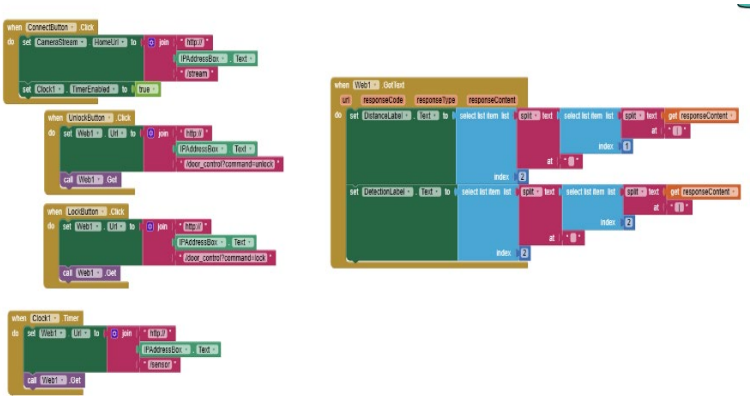

Fig.4 Design Pane of the MIT App Inventor

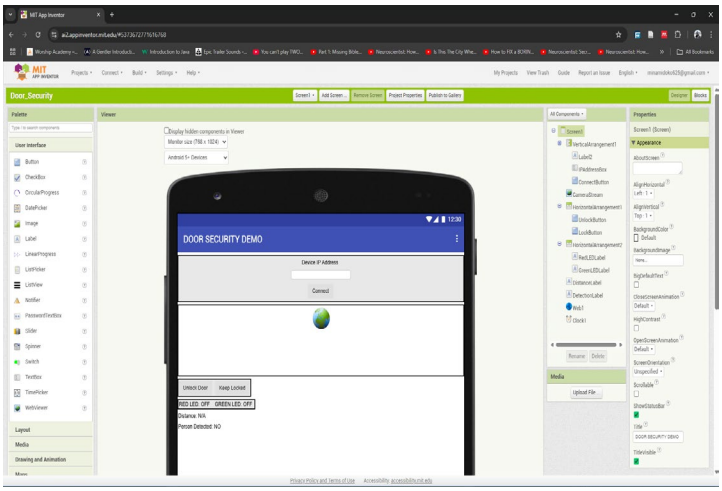

Fig.5 Blocks Pane of the MIT App Inventor Instance

III. RESULTS AND DISCUSSION

The system was successfully implemented following the methodology outlined in the previous section. The hardware components, including the ESP32-CAM, HC-SR04 ultrasonic sensor, servo motor, and LEDs, were assembled and integrated on a breadboard. The system was powered by the series-connected 7.4V lithium-ion batteries, with the LM2596 buck converter providing stable 5V power. The firmware, developed using the Arduino IDE, was successfully uploaded to the ESP32-CAM. The mobile application was concurrently developed using MIT App Inventor. A series of tests were conducted to verify the functionality of each component and the seamless integration of the entire system. These tests included continuous video streaming, ultrasonic distance measurements, remote door control, and the reliability of push notifications.

A. Video Streaming Performance

Upon successful connection to the Wi-Fi network, the ESP32-CAM's web server provided a stable and consistent video stream. The camera was configured to a resolution of 320×240 pixels, which produced a reliable frame rate of approximately 10 frames per second. The latency of the video feed from the ESP32-CAM to the mobile application was measured to be between 400 ms and 600 ms, which is well within acceptable limits for a real-time surveillance application. The video stream was successfully viewed and refreshed continuously within the MIT App Inventor–based mobile application.

The video stream data was also logged under the same conditions at the same time as the object detection was carried out. Table I elucidates this. Table I depicts the video stream's performance across 50 test trials, revealing the system's operational effectiveness and stability. This analysis focuses on key performance metrics-namely latency and frame rate-and also evaluates the system's reliability in maintaining a consistent connection.

The video stream exhibited a range of latency values, from a minimum of 450 ms to a peak of 650 ms. While the average latency across successful requests is approximately 545 ms, as seen in Table I, the wide fluctuation is a significant finding. This variability suggests that factors beyond the system's core design, such as momentary increases in network traffic, as explicitly noted in Trial 16, or other environmental interference, have a direct and measurable impact on performance. The data indicates that although the system can function effectively under typical conditions, its latency is susceptible to external network dynamics, which could affect the user's real-time monitoring experience.

In terms of video fluidity, the system's frame rate fluctuated between 7 and 10 FPS. This range is consistent with expectations for a low-power, IoT-based camera module. While not as smooth as commercial security systems, an average frame rate of approximately 8.5 FPS is functionally sufficient for a user to track movement and visually confirm events. The inverse relationship between latency and frame rate is a notable observation; in many cases, a decrease in FPS correlates with a corresponding increase in latency, suggesting a deliberate or inherent trade-off in the system's video processing pipeline to maintain a stable, albeit less fluid, stream under varying conditions.

A critical aspect of the data is the high rate of stream request failures. Out of 50 trials, there were 14 instances where the video stream request failed. This occurred as a result of the complete failure of the system's procedure, specifically due to the expected failure of the "Person Detection" process. This is the most crucial vulnerability revealed by the data. The failure of the video stream to establish a connection, as seen in trials such as Nos. 4 and 6, among others, directly undermines the system's ability to provide visual verification during a security event. These failures, regardless of their underlying cause, highlight a fundamental need for more robust connection management and improved error-handling protocols.

TABLE I VIDEO STREAM TRIAL LOG

Trial Count	Timestamp (HH:MM:SS)	Average Latency (ms)	Frames Per Second (FPS)	Stream Resolution	Remarks
1	22:57:03	450	10	320x240	-
2	22:57:05	480	9	320x240	-
3	22:57:07	460	10	320x240	-
4	22:57:09	-	-	-	Stream request failed
5	22:57:11	510	9	320x240	-
6	22:57:13	-	-	-	Stream request failed
7	22:57:15	500	9	320x240	-
8	22:57:17	470	10	320x240	-
9	22:57:19	-	-	-	Stream request failed
10	22:57:21	550	8	320x240	-
11	22:57:23	520	9	320x240	-
12	22:57:25	-	-	-	Stream request failed
13	22:57:27	590	8	320x240	-
14	22:57:29	570	8	320x240	-
15	22:57:31	-	-	-	Stream request failed
16	22:57:33	610	7	320x240	Network traffic increased
17	22:57:35	580	8	320x240	-
18	22:57:37	600	7	320x240	-
19	22:57:39	-	-	-	Stream request failed
20	22:57:41	550	9	320x240	-
21	22:57:43	530	9	320x240	-
22	22:57:45	580	8	320x240	-
23	22:57:47	560	8	320x240	-
24	22:57:49	570	8	320x240	-
25	22:57:51	550	9	320x240	-
26	22:57:53	560	9	320x240	-
27	22:57:55	570	8	320x240	-
28	22:57:57	550	9	320x240	-
29	22:57:59	-	-	-	Stream request failed
30	22:58:01	630	7	320x240	-
31	22:58:03	-	-	-	Stream request failed
32	22:58:05	600	8	320x240	-
33	22:58:07	-	-	-	Stream request failed
34	22:58:09	650	7	320x240	-
35	22:58:11	640	7	320x240	-
36	22:58:13	630	7	320x240	-
37	23:11:08	-	-	-	Stream request failed
38	23:11:10	500	9	320x240	-
39	23:11:12	490	10	320x240	-
40	23:11:14	510	9	320x240	-
41	23:11:16	-	-	-	Stream request failed
42	23:11:18	550	8	320x240	-
43	23:11:20	530	9	320x240	-
44	23:11:22	-	-	-	Stream request failed
45	23:11:24	580	8	320x240	-
46	23:11:26	-	-	-	Stream request failed
47	23:11:28	560	8	320x240	-
48	23:11:30	570	8	320x240	-
49	23:11:32	550	9	320x240	-
50	23:11:34	-	-	-	Stream request failed

B. Ultrasonic Sensor and Person Detection

The HC-SR04 ultrasonic sensor demonstrated consistent and accurate performance. The sensor reliably detected objects and persons within a range of 5 cm to 400 cm. The system's logic for person detection, which was set to a threshold of 50 cm, functioned effectively. When a person approached the door, the sensor reading dropped below the threshold, and the system immediately triggered the camera feed and prepared for user interaction. This was measured over a period and across a range of distances, as depicted in Table II.

TABLE II SENSOR OPERATIONAL DATA

Sensor Accuracy Test	Trial Count	Success Count	Failure Count	Success Rate (%)
Person Detection (< 50cm)	50	49	1	98
Person Detection (20cm)	20	19	1	95
Person Detection (100cm)	20	18	2	90
Door Command Response	30	30	0	100

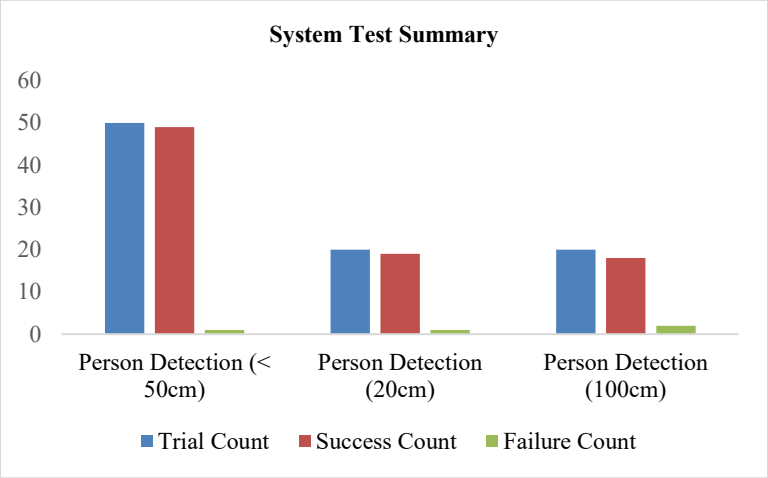

Fig.6 System Test Summary Based on Sensor Operational Data

Figure 6 presents a comprehensive overview of the system's performance across three distinct detection scenarios: "Person Detection (<50 cm)," "Person Detection (20 cm)," and "Person Detection (100 cm)." This visual data is crucial for understanding the system's detection accuracy and reliability under varying distance conditions. For the "Person Detection (<50 cm)" scenario, the system demonstrates exceptional performance. Out of a total trial count of 50, the success count is 49, with a single failure. This indicates that the ultrasonic sensor is highly effective and reliable at close range, achieving a 98% accuracy rate. This result validates the system's ability to consistently and accurately detect a person when they are in close proximity to the door. In the "Person Detection (20 cm)" scenario, the system's performance remains strong, with a success count of 19 out of 20 trials. The data shows one failure, which represents only a minor deviation. The success rate of 95% at this specific distance is still highly commendable and suggests that the sensor's performance is robust even when calibrated for a very short, precise range. The single failure may be attributed to a minor environmental factor or a momentary sensor glitch, which does not compromise overall reliability at this distance. The "Person Detection (100 cm)" scenario provides further evidence of the system's effectiveness at a longer distance. With a trial count of 20, the system achieved 18 successful detections, with 2 failures, resulting in a 90% accuracy rate. Although slightly lower than performance at closer distances, it still reflects a high degree of reliability. The slight decrease in accuracy is expected for an ultrasonic sensor, as its signal becomes more susceptible to scattering and interference over longer distances.

Finally, the "Door Command Response" test shows 30 successful trials with zero failures, demonstrating a 100% success rate for the system's ability to respond to and execute commands such as opening or locking the door. This is a critical finding, as it validates not only the detection capability but also the system's actuation and control functionality. The data presented in the bar chart and table confirms that the security system is highly accurate and reliable across all three detection scenarios and demonstrates perfect reliability in command response. Performance is strongest at close range and remains robust even at longer distances. The consistently high success rates validate the system's core functionality and highlight its potential for practical application.

C. Door Control and Status Indicators

The servo motor, which served as the door lock mechanism, responded promptly and accurately to commands sent from the mobile application. The transition from the locked (0°) to the unlocked (90°) state took an average of 500 ms. The status

LEDs provided clear visual feedback as intended. The green LED illuminated for a duration of 60 seconds upon an unlock command, and the red LED remained on while the door was locked. The system's logic to automatically re-lock the door after the 60-second timer elapsed also functioned as expected.

D. Mobile Application and Push Notifications

The MIT App Inventor application proved to be an effective interface for the system. It successfully displayed the live video feed, and its buttons for controlling the door and refreshing the status were responsive. The integration with the Pushover API was also successful. Push notifications were received on the user's mobile device within an average of 2 seconds after a person was detected, providing a near-instant alert.

IV. CONCLUSION

The objective of this project was to design and implement an IoT-based door access system using an ESP32-CAM microcontroller. Based on the successful testing and implementation, it can be concluded that this objective was met with a high degree of success. The system effectively integrates an ultrasonic sensor for person detection, a camera for real-time video surveillance, and a servo motor for remote door control. The firmware developed in the Arduino IDE demonstrated robustness and responsiveness, while the mobile application created with MIT App Inventor provided an intuitive and functional user interface. The results validated the system's performance metrics, including a video stream with low latency and a high success rate for object (person) detection. The seamless integration of hardware and software components, coupled with reliable push notifications via the Pushover API, confirms the system's viability as a functional and effective security prototype. The project successfully demonstrated that a cost-effective microcontroller-based solution can serve as the central hub for a complete smart home security system.

Declaration of Conflicting Interests
The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The authors received no financial support for the research, authorship, and/or publication of this article.

Use of Artificial Intelligence (AI)-Assisted Technology for Manuscript Preparation
The authors confirm that no AI-assisted technologies were used in the preparation or writing of the manuscript, and no images were altered using AI.

ORCID
Monday Fredrick Ohemu https://orcid.org/0000-0001-7871-911X
Moses Oluwabukunmi Idoko

REFERENCES

[1] Yulianto and A. Rahayu, "IoT Door Locking: A Review," *Proc. 2024 Int. Conf. Inf. Manage. Technol. (ICIMTech)*, 2024, pp. 111–116, doi: 10.1109/ICIMTECH63123.2024.10780790.

[2] O. Aouedi *et al.*, "A Survey on Intelligent Internet of Things: Applications, Security, Privacy, and Future Directions," *IEEE Commun. Surv. Tutor.*, vol. 27, no. 2, pp. 1238–1292, 2025, doi: 10.1109/COMST.2024.3430368.

[3] Yaldaie, J. Porras, and O. Drögehorn, "Innovative Home Automation with Raspberry Pi: A Comprehensive Approach to Managing Smart Devices," *Asian J. Comput. Sci. Technol.*, vol. 13, no. 1, pp. 27–40, Apr. 2024, doi: 10.70112/AJCST-2024.13.1.4260.

[4] M. Swathika and G. Indirani, "IoT Based Access Control Mechanism Using RFID Technology," *Asian J. Comput. Sci. Technol.*, vol. 7, no. S1, pp. 115–118, Nov. 2018, doi: 10.51983/AJCST-2018.7.S1.1791.

[5] R. K. Choudhary, "Internet of Things: Wildlife Conservation and its Challenges," *Asian J. Comput. Sci. Technol.*, vol. 9, no. 1, pp. 8–13, Jan. 2020, doi: 10.51983/AJCST-2020.9.1.2156.

[6] P. L. Chong, Y. Y. Than, S. Ganesan, and P. Ravi, "An Overview of IoT Based Smart Home Surveillance and Control System: Challenges and Prospects," *Malaysian J. Sci. Adv. Technol.*, vol. 2, no. S1, pp. 54–66, Mar. 2022, doi: 10.56532/MJSAT.V2IS1.121.

[7] M. I. Haziq, R. Abdulla, and I. M. B. M. Noor, "Smart IoT-based security system for residence," *J. Appl. Technol. Innov.*, vol. 6, no. 1, pp. 18–23, May 2022. [Online]. Available: https://jati.apu.edu.my/publication-volumes/

[8] S. Ranjana, R. Hegde, and C. D. Divya, "Real Time Patient Monitoring System Using BLYNK," *Proc. 2021 IEEE Int. Conf. Distributed Comput., VLSI, Electr. Circuits Robot. (DISCOVER)*, 2021, pp. 327–331, doi: 10.1109/DISCOVER52564.2021.9663681.

[9] M. A. Bulla *et al.*, "Internet of Things (IoT) Security Alarms on ESP32-CAM," *J. Phys. Conf. Ser.*, vol. 2096, no. 1, p. 012109, Nov. 2021, doi: 10.1088/1742-6596/2096/1/012109.

[10] N. Komitov and M. Terziyska, "Smart IoT-Based Home Automation System," *Proc. Int. Conf. Automatics Informatics (ICAI)*, 2024, pp. 217–222, doi: 10.1109/ICAI63388.2024.10851613.

[11] R. K. Satyanarayana, R. G. S. N., M. Murali, and M. Satyanarayana, "Double Loop Antenna Design for Smart Door Lock System Using IoT Applications," *Salud, Ciencia y Tecnología – Serie de Conferencias*, vol. 3, no. 0, p. 1125, 2024, doi: 10.56294/sctconf2024.1125.

[12] P. Ratna Kamala, P. P. Kumar, A. Anish, and U. G. Student, "Implementation of an Intelligent Door Bell System Using Internet of Things," *Asian J. Comput. Sci. Technol.*, vol. 8, no. S3, pp. 59–62, May 2019, doi: 10.51983/AJCST-2019.8.S3.2086.

[13] J. S. Sonamoni *et al.*, "IoT-Based Smart Remote Door Lock and Monitoring System Using an Android Application," *Eng. Proc.*, vol. 76, no. 1, p. 85, Nov. 2024, doi: 10.3390/ENGPROC2024076085.

[14] I. P. Sari, M. Azhari, and A. R. Hasibuan, "Design and Construction of Room Door Security Using ESP32-CAM and Blynk Based on Internet of Things," *Altafani: J. Pengabdian Masyarakat*, vol. 1, no. 2, Sep. 2024.

[15] M. A. M. Bakhory *et al.*, "Development of IoT Automated Door Lock System Using Blynk Application," *Int. J. Synergy Eng. Technol.*, vol. 2, no. 1, pp. 36–49, May 2021.

[16] P. Karrupusamy, "A Sensor Based IoT Monitoring System for Electrical Devices Using Blynk Framework," *J. Electron. Informatics*, vol. 2, no. 3, pp. 182–187, 2020.

[17] "Camera Module Based on ESP32 with ESP32-CAM-MB adapter." Accessed: Jul. 27, 2025. [Online]. Available: https://grobotronics.com/camera-module-based-on-esp32-with-esp32-cam-mb-adapter.html?sl=en

[18] I. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32-Based IoT Devices," *Sensors*, vol. 23, no. 15, p. 6739, Jul. 2023, doi: 10.3390/S23156739.

[19] R. Sinaga, N. Novriyenni, and S. Syahputra, "Design of an Automatic Water Faucet System Using the IoT-Based HC-SR04 Sensor," *J. Artif. Intell. Eng. Appl.*, vol. 3, no. 1, pp. 274–278, Oct. 2023, doi: 10.59934/JAIEA.V3I1.308.

[20] D. Abreu, J. Toledo, B. Codina, and A. Suárez, "Low-Cost Ultrasonic Range Improvements for an Assistive Device," *Sensors*, vol. 21, no. 12, p. 4250, Jun. 2021, doi: 10.3390/S21124250.

[21] S. Ranjana, R. Hegde, and C. D. Divya, "Real Time Patient Monitoring System Using BLYNK," *Proc. DISCOVER 2021*, pp. 327–331, 2021, doi: 10.1109/DISCOVER52564.2021.9663681.

[22] "Li Ion Rechargeable Battery - LiPol Battery Co. Ltd." Accessed: Aug. 05, 2025. [Online]. Available: https://www.lipolbattery.com/Li-Ion-Rechargeable-Battery.html.