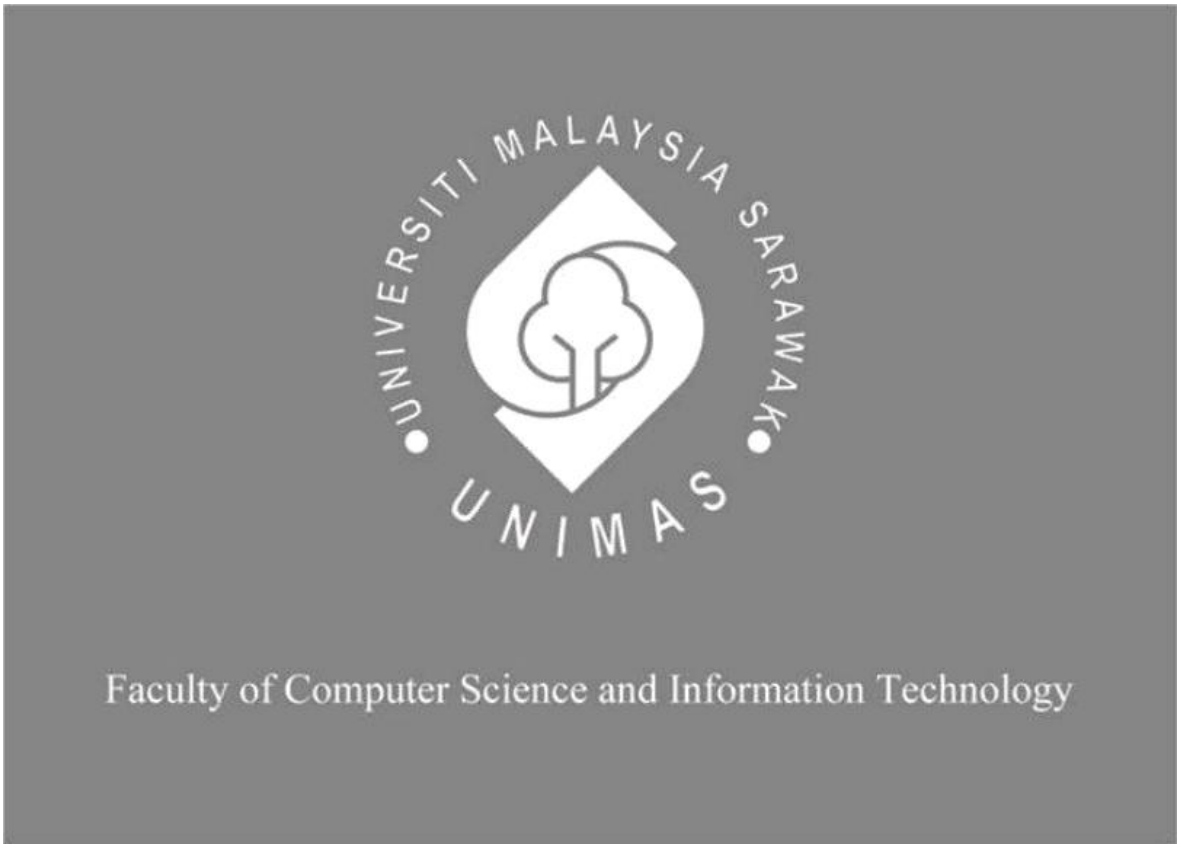

I-SENSOR ALERT: INTERNET OF THINGS-BASED INTRUSION PREVENTION ALERT

Fatin Nazifa Binti Shahrin

Bachelor of Computer Science with Honours

(Network Computing)

2024

SENSOR ALERT: INTERNET OF THINGS-BASED INTRUSION PREVENTION ALERT

FATIN NAZIFA BINTI SHAHRIN

This project is submitted in partial fulfillment of the
requirements for the degree of
Bachelor of Computer Science with Honours
(Network Computing)

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK
2024

AMARAN I-SENSOR: AMARAN PENCEGAHAN PENCEROBOHAN BERASASKAN IOT

FATIN NAZIFA BINTI SHAHRIN

Projek ini merupakan salah satu keperluan untuk

Ijazah Sarjana Muda Sains Komputer Dan

Teknologi Maklumat

(Pengkomputeran Rangkaian)

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2024

DECLARATION OF ORIGINALITY

I hereby declare that this research together with all of its content is none other than of my own work, with consideration of the exception of research based information and relative materials that were adapted and extracted from other resources, which have evidently been quoted or stated respectively.

Signed,

…… 28 July 2025

FATIN NAZIFA BINTI SHAHRIN
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak

ACKNOWLEGDEMENT

I would like to express my sincere gratitude to all those who have supported and contributed to the completion of this final year project.

First and foremost, I would like to extend my deepest appreciation to the Faculty of Computer Science and Information Technology for providing me with the resources, knowledge, and platform to pursue this project. Their continuous support throughout my academic journey has been invaluable.

I am profoundly grateful to my supervisor, Dr. Nazri Bin Khairuddin Yap, for his guidance, encouragement, and constant support throughout this project. His expertise and insights have been instrumental in shaping this work and have greatly contributed to my learning experience.

I would also like to thank the examiners for their time, effort, and constructive feedback, which have helped improve the quality of this project.

My heartfelt thanks go to my parents for their unwavering love, encouragement, and belief in me. Their support has been a constant source of strength, and I am forever grateful for their sacrifices and dedication.

Finally, I would like to express my appreciation to my friends for their moral support, motivation, and help during this project. Their companionship made this journey more enjoyable and memorable.

Thank you all for your invaluable contributions and encouragement.

Table of Contents

List of Figures

List of Tables

ABSTRACT

This project focuses on the development of an Internet of Things (IoT)-based intrusion prevention system that utilizes door sensors for enhanced security. In today's world, conventional security systems often suffer from slow response times, which can result in significant damage or loss before any action is taken. The aim of this project is to create a real-time alert system that enables immediate detection of intruders, offering a quick response to unauthorized access. The system is designed to be user-friendly, allowing for easy monitoring and control through mobile devices, and can be easily integrated into existing security setups. Using a combination of Arduino, sensors, and security cameras, the project offers a modular and cost-effective solution for monitoring sensitive areas in offices or businesses.

The system consists of door sensors that detect movement and send notifications to the user's smartphone, allowing for instant awareness and response. The project follows an Agile methodology to ensure adaptability and continuous improvement based on user feedback. This approach helps address issues such as human error and slow reaction times that are common in traditional security systems. Ultimately, this project aims to provide a reliable and efficient security solution for businesses, ensuring the safety of sensitive documents and valuable assets.

Abstrak

Projek ini memberi tumpuan kepada pembangunan sistem pencegahan pencerobohan berasaskan Internet of Things (IoT) yang menggunakan penderia pintu untuk keselamatan yang dipertingkatkan. Dalam dunia hari ini, sistem keselamatan konvensional sering mengalami masa tindak balas yang perlahan, yang boleh mengakibatkan kerosakan atau kerugian yang ketara sebelum sebarang tindakan diambil. Matlamat projek ini adalah untuk mencipta sistem amaran masa nyata yang membolehkan pengesanan segera penceroboh, menawarkan respons pantas kepada akses tanpa kebenaran. Sistem ini direka bentuk untuk mesra pengguna, membolehkan pemantauan dan kawalan mudah melalui peranti mudah alih, dan boleh disepadukan dengan mudah ke dalam persediaan keselamatan sedia ada. Menggunakan gabungan Arduino, penderia dan kamera keselamatan, projek ini menawarkan penyelesaian modular dan kos efektif untuk memantau kawasan sensitif di pejabat atau perniagaan.

Sistem ini terdiri daripada penderia pintu yang mengesan pergerakan dan menghantar pemberitahuan kepada telefon pintar pengguna, membolehkan kesedaran dan tindak balas segera. Projek ini mengikut metodologi Agile untuk memastikan kebolehsuaian dan penambahbaikan berterusan berdasarkan maklum balas pengguna. Pendekatan ini membantu menangani isu seperti ralat manusia dan masa tindak balas yang perlahan yang biasa dalam sistem keselamatan tradisional. Akhirnya, projek ini bertujuan untuk menyediakan penyelesaian keselamatan yang boleh dipercayai dan cekap untuk perniagaan, memastikan keselamatan dokumen sensitif dan aset berharga

CHAPTER 1: INTRODUCTION

1.1 Introduction

In today's era of globalization, technology has significantly influenced all areas of communication. Nowadays, technology is a familiar concept, serving as a means or container that provides essential support for human survival and comfort in the modern age. One example is the laptop, which has become increasingly commonplace compared to 20 years ago.

The integration of door sensors into smart systems has revolutionized the way we interact with living spaces (Simatupang & Tambunan, 2022). The client can now receive real-time notifications on their smartphones, ensuring they are always aware of who enters or exits their spaces. Additionally, these sensors can work in tandem with other smart devices, such as security cameras or alarms, to create a comprehensive security network (Anitha, 2017).

As technology continues to advance, door sensors are becoming more sophisticated, incorporating features such as remote access, integration with artificial intelligence, and enhanced connectivity options. This evolution not only enhances user experience but also opens new possibilities for automation and smart living.

In summary, door sensors are vital to modern security and automation systems. Their ability to provide real-time feedback and integrate seamlessly with other technologies makes them indispensable for anyone looking to enhance their home or business's safety and efficiency.

1.2 Problem Statements

Conventional security systems usually react slowly, which can cause serious harm or loss before any remedial measures are taken. This inherent response time lag weakens security systems' effectiveness, especially in urgent situations where prompt action is required. As a result, the safety of people on the property may be in danger, and valuable goods may be endangered. These systems' incapacity to react quickly highlights The need for more sophisticated and adaptable security solutions that can eliminate any threats instantly. Human

mistake is still a major organizational security threat, especially when workers are rushing and forget to lock their office doors. These mistakes can present serious security threats by letting unauthorized people enter locations that are restricted. The integrity of the workplace may be jeopardized by theft, data breaches, and other security problems brought on by this oversight.

The presence of unauthorized individuals within secure locations poses a serious threat to organizational security, as these individuals can easily access restricted areas and potentially cause the loss of valuable items. Effective security for alerting intruders is necessary to prevent such incidents, ensuring that only authorized personnel can enter sensitive areas.

Budget constraints within government offices often limit the ability to implement comprehensive and modern security systems, leaving these facilities vulnerable to various threats. Financial limitations can hinder the adoption of advanced technologies and practices that are essential for effective security management. As a result, outdated security measures may remain in place, increasing the likelihood of security breaches and compromising the safety of personnel and assets.

1.3 Objectives

i) To develop a user-friendly interface that allows easy monitoring and control of the security system through mobile devices.

ii) To design a modular IoT security system that can be easily installed, enabling straightforward deployment without extensive modifications.

iii) To develop a real-time alert system for immediate and alert intruder detection to authorized personnel when an intruder's entry occurs.

1.4 Methodologies

According to Fireteanu (2020), agile methodology is an iterative and incremental approach to project management that prioritizes adaptability, collaboration, and customer input. This

methodology is ideal for the development of the Intruders Alerting System since it allows for continual modification and adaptation to user needs throughout the project's lifecycle.

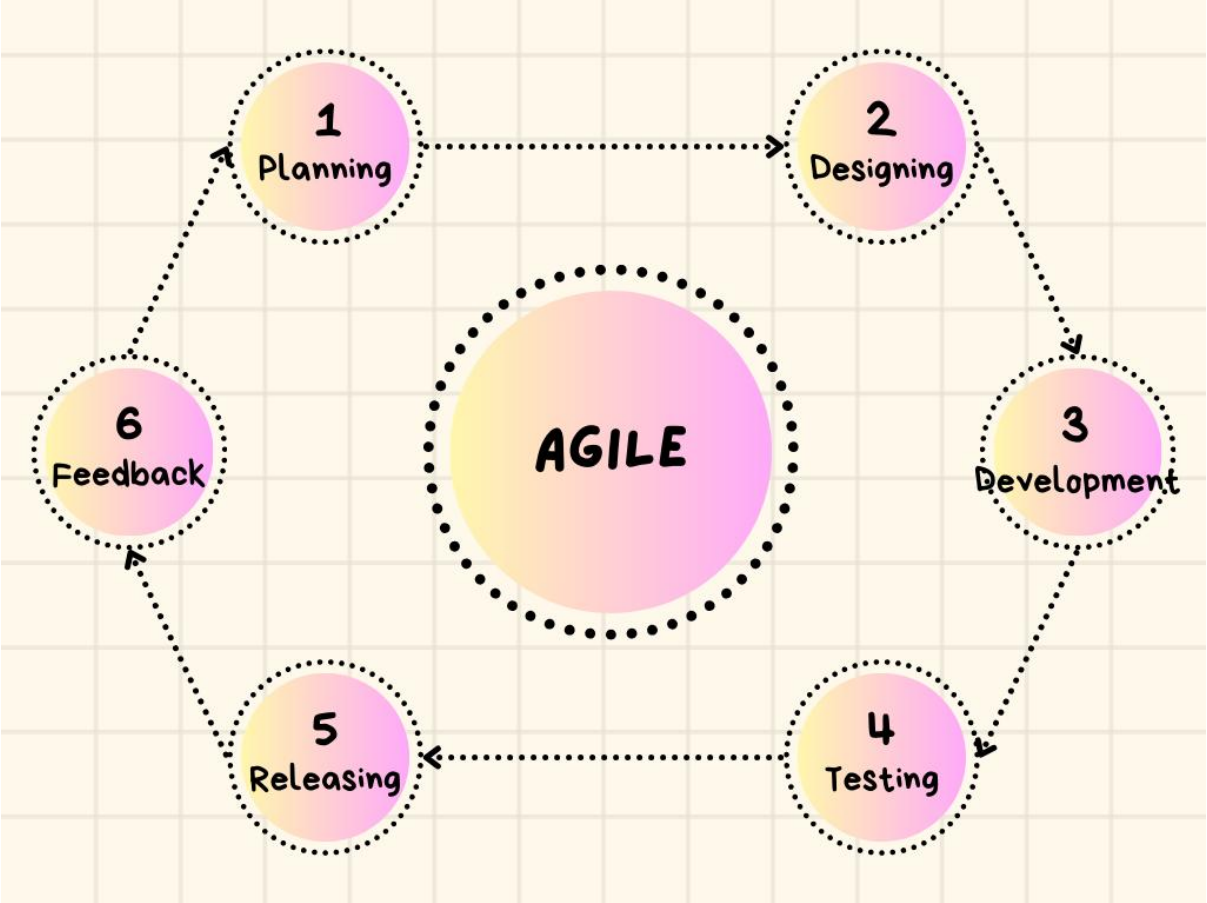

Figure 1.1: Agile Method

Step 1: Planning. In this initial phase, engaging in a comprehensive conversation with internship supervisor establishes a clear understanding of project goals. By researching existing security solutions and identifying gaps, such as the slow response times of conventional systems then it set a strong foundation for the project. This planning phase is crucial for defining the project scope, including the integration of door sensors and real-time notifications, which are central to the objectives.

Step 2: The design phase, where need to create initial sketches and then develop 3D models, is vital for visualizing the system's architecture and user interface. This aligns with the objective

to develop a user-friendly interface for mobile monitoring. By iterating through sketches and utilizing design software, ensure that the final design is intuitive and meets user expectations. This step also allows to consider how the door sensors will integrate with other smart devices, enhancing the overall security network.

Step 3: During the development phase, bring the designs to life by assembling hardware and writing code. This is where the integration of door sensors, mobile notifications, and other smart devices occurs. By focusing on creating a cost-effective and energy-efficient solution, it addresses one of the key objectives. This phase also allows for iterative coding and testing, which is a core principle of Agile methodology, enabling this project to adjust based on real-time feedback.

Step 4: The testing phase is critical for ensuring that the system functions as intended. By verifying the system's capabilities under various conditions, such as different types of intrusions it can identify any weaknesses or areas for improvement. This step directly addresses project's problem statement regarding the slow response time of conventional systems. Ensuring that notifications and alarms work effectively will enhance the system's reliability and user trust.

Step 5: Once testing is complete and the system is validated, releasing it for public use, specifically for the District Office and Finance Unit staff, marks a significant milestone. This step not only provides users with enhanced security but also establishes a feedback loop where users can report their experiences and suggest improvements. This aligns with objective of creating a detection solution capable of alerting intruders effectively.

Step 6: Gathering user feedback post-release is essential for continuous improvement. By actively seeking input from users, this project can refine the system to better meet their needs and address any shortcomings. This iterative feedback process is fundamental to Agile methodology, ensuring that the system evolves alongside technological advancements and user expectations. It also helps in addressing the problem of human error, as users can suggest features that might mitigate such risks.

1.5 Scope

User scope: The primary users of this system are District Officers and Finance Unit staff, whose offices contain sensitive documents and assets that require enhanced security

System scope: This project combines Arduino, sensors, and a security camera to construct an alert door system. The Arduino code communicates with a magnetic sensor, which detects movement at the entrance and sends notifications accordingly. The door sensors consist of a reed switch and a magnet that work together to create a closed circuit. The system provides real-time monitoring and an immediate response to unwanted access attempts.

1.6 Significant of Project

i) Security for intruder alerts is a crucial element in security systems, providing notifications whenever someone enters a monitored room, thereby enhancing overall security.

ii) The system alleviates concerns about the loss of important documents or valuables, as it minimizes the risk of theft, and develops a user-friendly interface that allows easy monitoring and control of the security system through mobile devices.

iii) Upon detecting that a door has been opened, the system promptly sends a notification to the user's device, allowing for immediate awareness and response.

1.7 Project Schedule

Figure 2 shows the FYP project schedule in the Gantt chart. The estimated duration of the project is 295 days, which starts from 06 October 2024 until 28 July 2025.

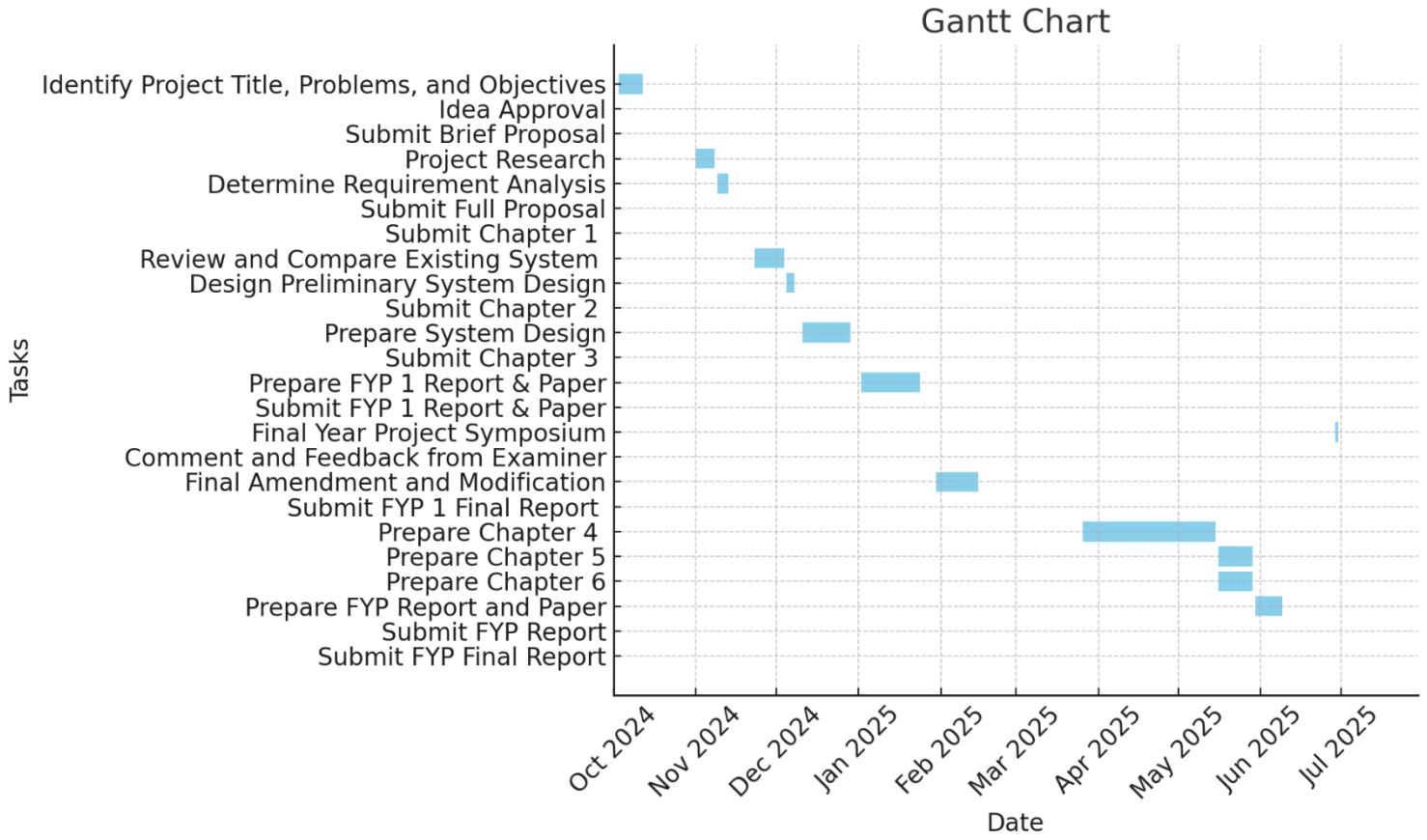

Figure 1.2: Gantt Chart

1.8 Expected Outcome

The IoT-Based Security for Intruders Alert is designed to detect unauthorized access, primarily through its door sensors, a key component of the system's security framework. The system employs magnetic door sensors that detect a door's state, whether it’s opened or closed. When the door sensor detects unauthorized access, the system immediately sends a notification to the user's device. This real-time alert allows users to be instantly aware of potential security threats and respond immediately.

1.9 Conclusion

This chapter outlined the development of the I-Sensor Alert: IoT-Based Security for Intruders Alert system, focusing on its objectives, methodologies, and expected outcomes. The proposed system leverages magnetic door sensors, motion detection, and IoT technologies to provide real-time notifications for enhanced security. Using Agile methodology, the project ensures continuous improvement, from planning and design to development and testing phases. The system aims to address limitations in conventional security systems by providing immediate alerts, reducing human error, and ensuring ease of deployment. Ultimately, the project is expected to deliver a reliable, user-friendly solution that meets the security needs of District Officers and Finance Unit staff, safeguarding sensitive assets and documents.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In this chapter, I will conduct a literature review to explore the existing technologies and hardware related to the proposed system. The goal is to gather relevant information and analyze how these components have been used in previous studies or projects. The review will be divided into two main sections: hardware and existing technology.

The hardware review will focus on the different types of hardware that have been used in similar systems. I will look into their features, capabilities, and limitations, and consider how these might be relevant to the system I am proposing. This section aims to help me understand the strengths and weaknesses of various hardware options, and whether they can be applied or improved upon in my own project.

The second part of the review will look into the existing technologies. This will involve examining the software, tools, and methodologies that are currently being used in the field. By reviewing these, I will be able to identify what has already been done and see if there are any gaps or areas where my proposed system can offer improvements. This section will help me understand how these technologies have been implemented and what lessons can be learned from their success or failure.

Together, these sections will provide a solid foundation for the development of the proposed system, helping to shape its design and identify potential challenges.

2.2 Overview of Project

The idea of the project is to improve security measures in corporate settings. The initiative intends to tackle the major issues that traditional security systems encounter, like poor reaction times, human error, and financial limitations, which frequently expose facilities to security breaches. This system offers a complete and affordable solution for monitoring and preventing

unwanted entry by combining cutting-edge features like magnetic door sensors, real-time notifications, and an intuitive mobile interface.

Through the application of Internet of Things (IoT) technology, the innovative I-Sensor Alert: Internet of Things-Based Intrusion Prevention Alert enhances security protocols in business environments. The project aims to address the main problems that traditional security systems face, such as slow response times, human error, and budgetary constraints, which often leave facilities vulnerable to security breaches. By integrating state-of-the-art technologies like magnetic door sensors, real-time notifications, and an easy-to-use smartphone interface, this system provides a comprehensive and reasonably priced solution for monitoring and avoiding unwanted entrance.

2.3 Review on Hardware used

2.3.1 Review on ESP32-CAM

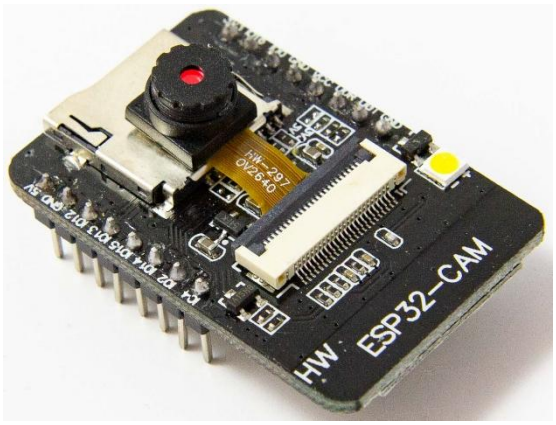

Figure 2.1: ESP-CAM

The ESP32-CAM is a compact, cost-effective microcontroller with an integrated camera module, making it ideal for IoT-based applications. According to Andreas et al. (2019), it combines the processing power of the ESP32 microcontroller with the ability to capture images and videos, allowing it to serve as both a controller and a visual monitoring device. The ESP32-CAM supports Wi-Fi and Bluetooth communication, enabling it to send real-time data to mobile devices. Additionally, it features a microSD card slot for storing captured images or videos locally. In your project, the ESP32-CAM plays a central role by capturing images or streaming live video when triggered by other sensors, such as the magnetic door sensor or motion sensor. Its ability to provide real-time monitoring enhances the security system, offering immediate insights into potential intrusions. The ESP32-CAM's compact design, low cost, and advanced features make it an efficient solution for building a reliable and modern IoT-based security system.

2.3.2 Review on Magnetic Door Sensor

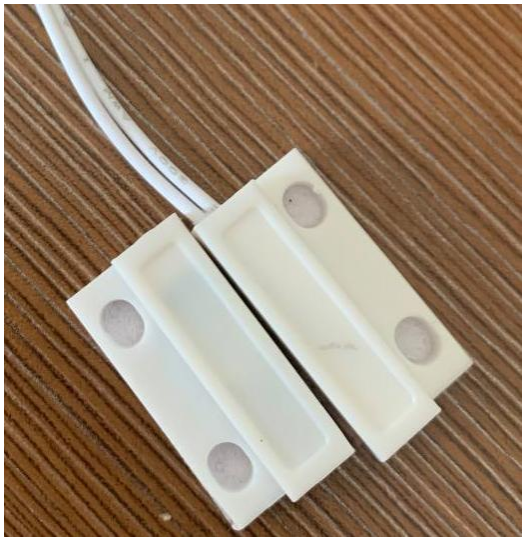

Figure 2.2: Magnetic Door Sensor

The magnetic door sensor is a straightforward, yet highly effective security component used to detect the opening or closing of doors. It consists of two main parts: a reed switch, which is an electrical switch that remains closed when in proximity to a magnet, and a magnet attached to the door. When the door is closed, the reed switch and magnet are aligned, maintaining a closed circuit. However, opening the door separates the magnet from the reed switch, breaking the circuit and triggering an alert. In your project, the magnetic door sensor acts as the primary trigger mechanism. Upon detecting unauthorized door openings, it activates the ESP32-CAM to capture images or record video and initiates a real-time alert to the user's mobile device. Its simplicity, reliability, and low power consumption make it a crucial component in the security system, ensuring accurate and immediate detection of intrusions. This sensor provides a cost-effective solution for securing doors while seamlessly integrating with other components of the system.

2.3.3 Review on Servo Motor

Figure 2.3: Servo Motor

A Micro Servo Motor is a small motor that can rotate to specific angles, usually between 0° and 180°. It is controlled by a signal from a microcontroller, like an Arduino. This type of motor is commonly used in projects where precise movements are needed, such as controlling a door lock or adjusting the position of a camera.

In this project, the micro servo motor plays an important role in automatically opening the door when it's triggered by the magnetic sensor. When the magnetic sensor detects any movement at the door, it sends a signal to the system. If the person approaching is authorized and acknowledges the system, the servo motor will receive a signal to rotate and unlock the door, allowing the person to enter.

However, if an intruder tries to access the door, the servo motor will not unlock the door. Instead, the system will trigger a buzzer, which sounds an alarm to alert others that an unauthorized person is attempting to enter. The buzzer will keep sounding until the intruder leaves or is stopped, helping to ensure that only authorized individuals can gain access to the area.

Overall, the micro servo motor is a key part of this security system, helping automate door access while ensuring that only the right people can open the door. It works with the magnetic

sensor to detect authorized users and protect against intruders by activating the buzzer when necessary.

2.4 Review on Similar Existing Systems

2.4.1 A Low Cost On-device Intruder Detection System for Smart Home Environment (Maiti et al., 2024)

According to Lovecek et al. (2013), the increasing adoption of smart home technologies has raised the demand for advanced security systems capable of protecting homes from unauthorized entry. Traditional intruder detection systems, particularly those based on cameras and sensors, have become widely available, but their high cost remains a major barrier to widespread use. As a result, recent advancements have focused on developing low-cost, on-device intruder detection systems, which have become an active area of research.

One notable contribution is the study by Maiti et al. (2024), who designed a low-cost intruder detection system for smart homes using affordable hardware like the Raspberry Pi 4 paired with a camera module. This system aims to address the issue of high-cost security solutions by leveraging efficient on-device processing, which also ensures the privacy of sensitive video data. To achieve real-time processing, the system utilizes motion detection through the frame subtraction technique and rapid face detection using Haar cascades and Kalman filters for improved accuracy and speed. As a result, the system achieves an average frame rate of 22 FPS, making it suitable for real-time intruder detection in resource-constrained environments such as smart homes.

Face detection is a critical component of many IoT-based intruder detection systems, particularly for identifying intruders. In this context, the study by Maiti et al. (2024) further improves the face identification process using template matching techniques, such as Zero-Mean Normalized Cross-Correlation (ZNCC), to speed up the process. This research demonstrates how cost-effective and efficient vision-based systems can be implemented for

intruder detection in smart home environments, where privacy and real-time performance are key factors.

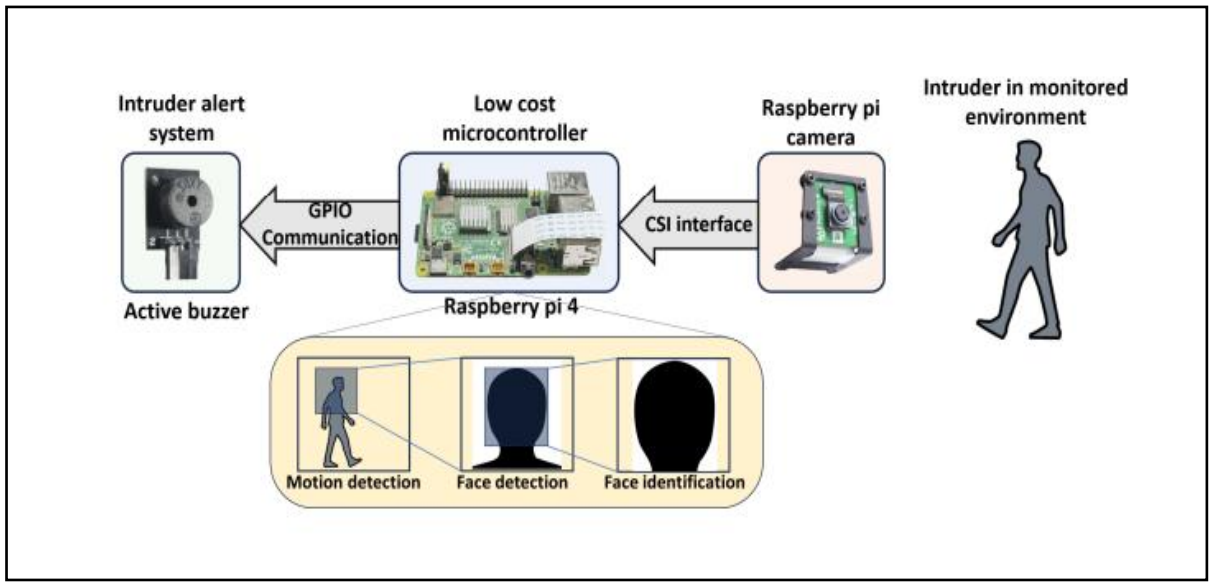

Figure 2.4: Architecture Diagram (Maiti et al., 2024)

2.4.2 Pyroelectric infrared sensor for intruder detection (Moghavvemi & Seng, 2004)

The use of Passive Infrared (PIR) sensors in intruder detection systems is widely recognized due to their efficiency and affordability. PIR sensors detect infrared radiation emitted by humans, which is then converted into an electric charge. This technology is commonly used in security systems for motion detection in homes and offices (Moghavvemi & Seng, 2004).

PIR sensors are highly sensitive to human body radiation, especially in the 8-14 μm wavelength range. When a person moves within the sensor's coverage area, the change in infrared radiation is detected, triggering an alert (Moghavvemi & Seng, 2004). These sensors are often combined with Fresnel lenses to enhance the detection range, allowing for broader coverage (Moghavvemi & Seng, 2004).

In modern IoT-based systems, PIR sensors are integrated with wireless communication networks using techniques like Frequency Shift Keying (FSK) to transmit data to a central computer (Moghavvemi & Seng, 2004). This enables remote monitoring and real-time alerts

for intrusions. The system's graphical user interface (GUI) displays an occupancy map of protected areas, making it easier to track security status.

Despite their effectiveness, PIR sensors have limitations, such as difficulty detecting stationary intruders (Moghavvemi & Seng, 2004). Combining PIR sensors with other technologies like ultrasonic or microwave sensors can improve detection in such cases. Additionally, the system's reliance on wireless communication introduces potential security concerns, which can be mitigated with secure transmission protocols.

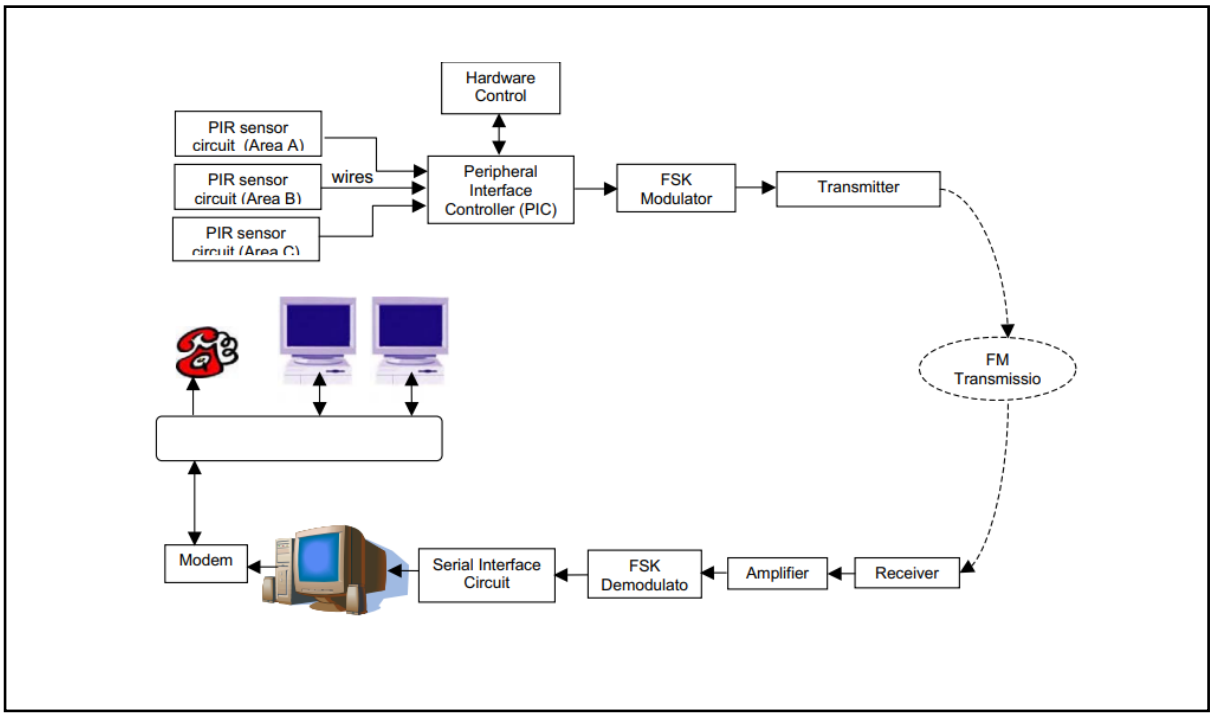

Figure 2.5: Block Diagram (Moghavvemi & Seng, 2004).

2.4.3 Prototype Smart Door Lock By Using Wireless Network Based on Arduino Uno (Masykuroh et al., 2021)

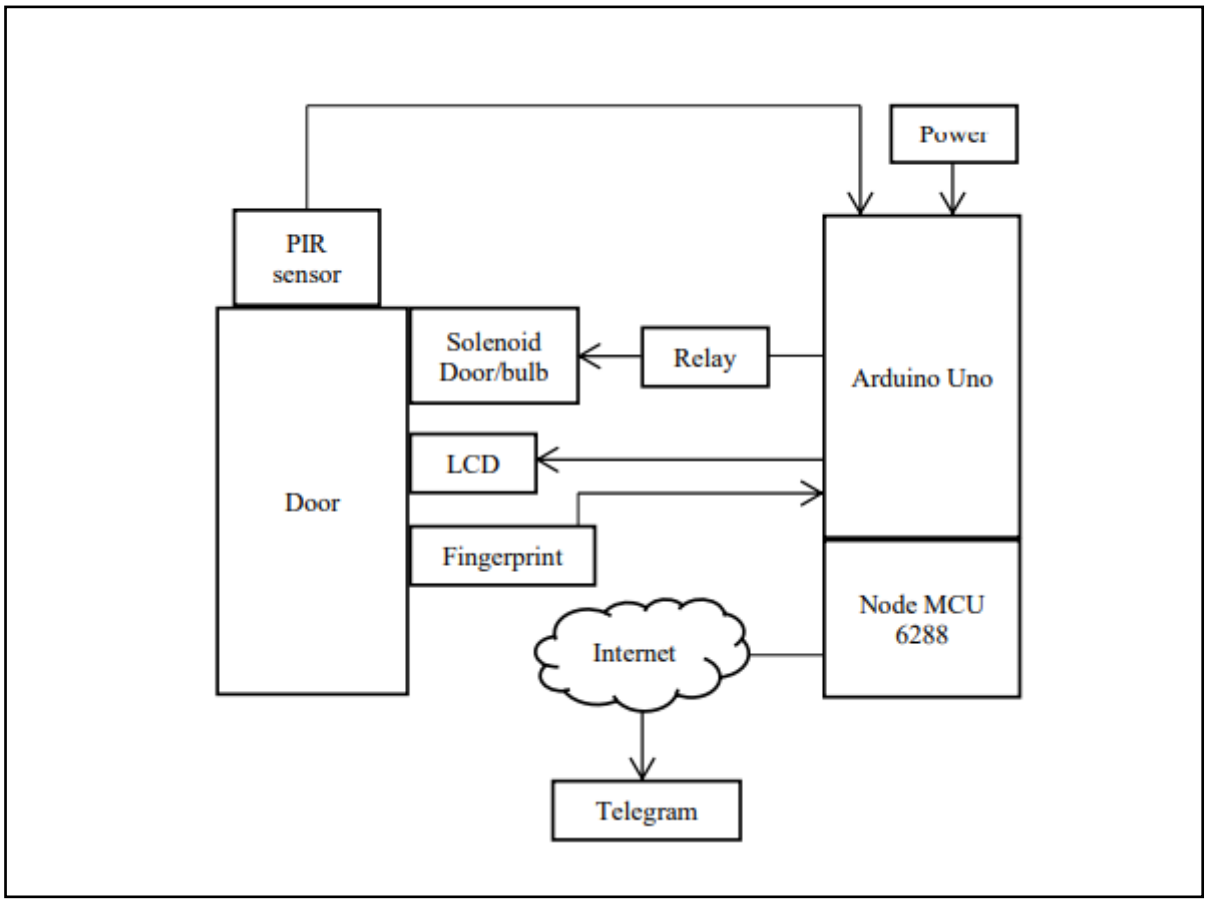

Figure 2.6: System Medelling (Masykuroh et al., 2021)

The Internet of Things (IoT) has significantly impacted home security systems, offering smarter and more efficient ways to protect properties. One key innovation is the smart door lock, which integrates various sensors and microcontrollers to enhance security. IoT-enabled systems use technologies such as fingerprint sensors, motion detectors, and remote monitoring applications to provide comprehensive protection.

Fingerprint sensors, like the FPM10A, are widely used for biometric authentication in security systems due to their high accuracy and reliability. These sensors capture unique fingerprint patterns for secure access, preventing unauthorized entry (Masykuroh et al., 2021). Alongside

fingerprint sensors, Passive Infrared (PIR) sensors, such as the HC-SR501, are employed for motion detection. PIR sensors detect heat emitted by human bodies, triggering alarms or notifications if unauthorized movement is detected (Kurnia et al., 2016).

The integration of IoT allows these systems to be controlled remotely. For example, combining Arduino Uno and the ESP8266 WiFi module enables real-time notifications via Telegram when the door is accessed. This not only enhances security but also provides users with remote monitoring capabilities, offering convenience and peace of mind (Ramadhani et al., 2021).

Solenoid door locks are another critical component in IoT-based security systems. These electronically controlled locks open when the correct fingerprint is recognized, providing a secure and automated entry system (Setiyanto et al., 2021). However, challenges such as securing the IoT network against cyberattacks and ensuring the reliability of wireless connections remain significant concerns (Shull, 2016).

In conclusion, IoT-based intruder detection systems, combining fingerprint sensors, motion detectors, and solenoid locks, offer enhanced security and convenience. However, addressing issues like network security and device compatibility is essential for widespread adoption and reliability.

2.4.4 Comparison Between the Similar Systems

This section will review the comparison of the intruder detection. Analysis of the similarities and dissimilarities will be shown through a simple table.

Table 2.1: Comparison of system

System / Features	A Low Cost On-device Intruder Detection System for Smart Home Environment	Pyroelectric infrared sensor for intruder detection	Prototype Smart Door Lock By Using Wireless Network Based on Arduino Uno	Proposed system
Wi-Fi Communication as a Standard	YES	YES	YES	YES
Camera Integration	YES	NO	NO	YES
Use motion detection	YES	YES	YES	YES
Use door state detection	NO	NO	YES	YES
Mobile App Availability	NO	NO	YES	YES

The comparison outlined in Table 2.1 clearly illustrates the key differences between the existing systems and the proposed system, focusing on their features and capabilities. One prominent similarity across all the systems, including the proposed system, is the use of Wi-Fi communication. This is crucial for enabling real-time data transmission and remote monitoring, allowing users to receive updates and control their systems remotely. Wi-Fi connectivity remains a fundamental component in modern smart home systems, ensuring constant communication and quick data transfer between devices.

When it comes to camera integration, two systems the A Low Cost On-device Intruder Detection System for Smart Home Environment and the proposed system, incorporate camera functionality, allowing for visual verification of intruders or events. This feature enhances the security system's effectiveness by providing users with a real-time visual feed. Motion detection, a core feature of most intruder detection systems, is present in all the systems compared. It plays a vital role in triggering alerts when unusual movement is detected, further boosting the system’s ability to monitor and protect the environment.

The integration of door state motion detection is another key differentiator, where only the Prototype Smart Door Lock by Using Wireless Network Based on Arduino Uno and the proposed system feature this capability. This functionality enables the system to monitor the status of the door (open or closed) and act accordingly, such as triggering an alert or activating other security measures.

Another significant difference lies in the availability of mobile applications for user interaction. The proposed system provides a mobile app for alert notifications and real-time monitoring, which meets modern user expectations for convenience and control. In contrast, the IoT Based Smart Intruder Detection System for Smart Homes does not include this mobile app integration, limiting its functionality for users who prefer mobile control. Moreover, a standout feature of the proposed system is its ability to display both data and images of intruders directly through

the Telegram application, offering an added layer of user engagement and immediacy. This integration with a popular messaging platform demonstrates a significant improvement over existing systems, enhancing user experience and convenience.

2.6 Chapter Summary

Overall, the proposed system is a more versatile, scalable, and robust solution for intruder detection and more secure. It addresses limitations present in the other systems while incorporating advanced technologies to meet modern security needs effectively. These enhancements position the proposed system as a comprehensive and forward-thinking approach to IoT-based security solutions.

CHAPTER 3: REQUIREMENT ANALYSIS AND DESIGN

3.1 Introduction

This chapter describes the methodology for implementing this project. The selected model will be further discussed in this chapter. The Agile Method would be the model used for I-Sensor Alert: Internet of Things-Based Intrusion Prevention Alert. The model applied in this project has 6 phases which are planning, design phase, development, testing, release, and feedback. The first phase would be gathering all information to plan the project. After completing all the information, design the project based on the information. During the development phase, it needs to bring the designs to life by assembling hardware and writing code. The testing phase is critical for ensuring that the system functions as intended. Once testing is complete and the system is validated, releasing it for public use, specifically for the District Office and Finance Unit staff, marks a significant milestone. Lastly, gathering user feedback post-release is essential for continuous improvement. The figure below will be a simple diagram of the model which is used in this project. This project is designed to detect unauthorized access, primarily through its door sensors, and notify the user using the application.

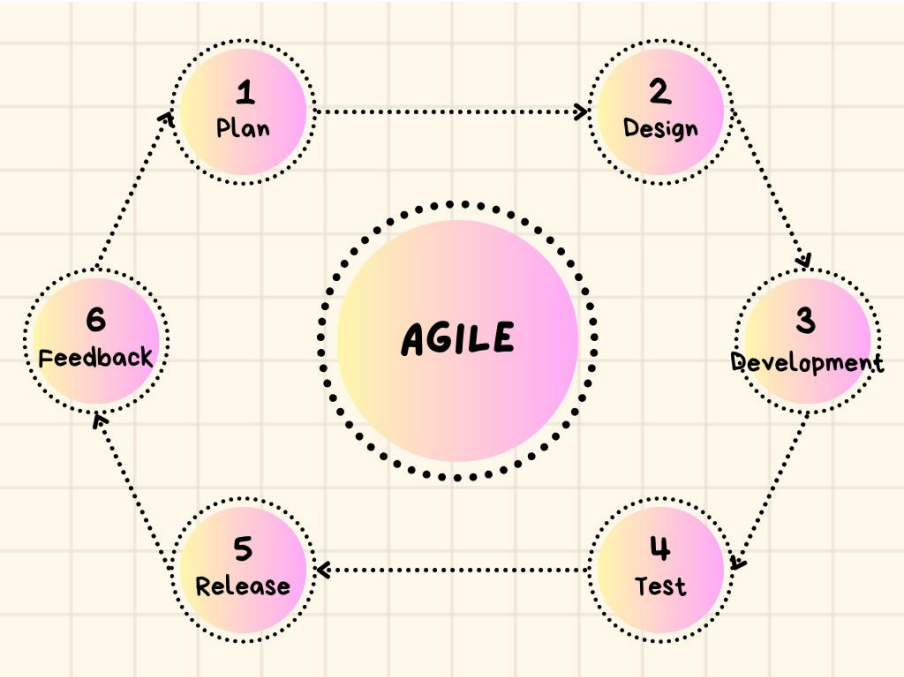

Figure 3.1: Agile Model

3.2 Plan

As mentioned above, this phase of information collection is related to the project. Then, through the information collected, problems will be identified, hence objectives will be stated according to the problems identified.

3.2.1 Data Collection

The data collection process begins with identifying the problem statement, which involves studying the background and real-world challenges that the system aims to address. This step helps in understanding the pros and cons of the current system and the limitations of existing technologies. By clearly identifying these problems, the project development becomes more focused and aligned with the actual needs. Once the problems and background are understood, the objectives and scope of the project are defined. Objectives represent the goals to be achieved, providing a clear direction for development. Additionally, the literature review examines existing systems related to the proposed project to gather relevant information and insights that will aid in the system's development.

3.2.2 Hardware Need

Based on the data collection process, the required hardware for the project is identified. The primary hardware component for this system is the ESP32-CAM microcontroller, which will be integrated with various sensors and modules. These include magnetic door sensors and micro servo motor, which play a critical role in detecting intrusions and ensuring the functionality of the system. These components were chosen based on their compatibility and suitability for addressing the identified problems and achieving the project’s objectives.

3.3 Design

3.3.1 Hardware Design

After the completion of information, this phase will progress on the design of the plan of the project. This phase aimed to acknowledge the existing system and hardware to make the best of it for solving the problem. The overall diagram of the system is shown below.

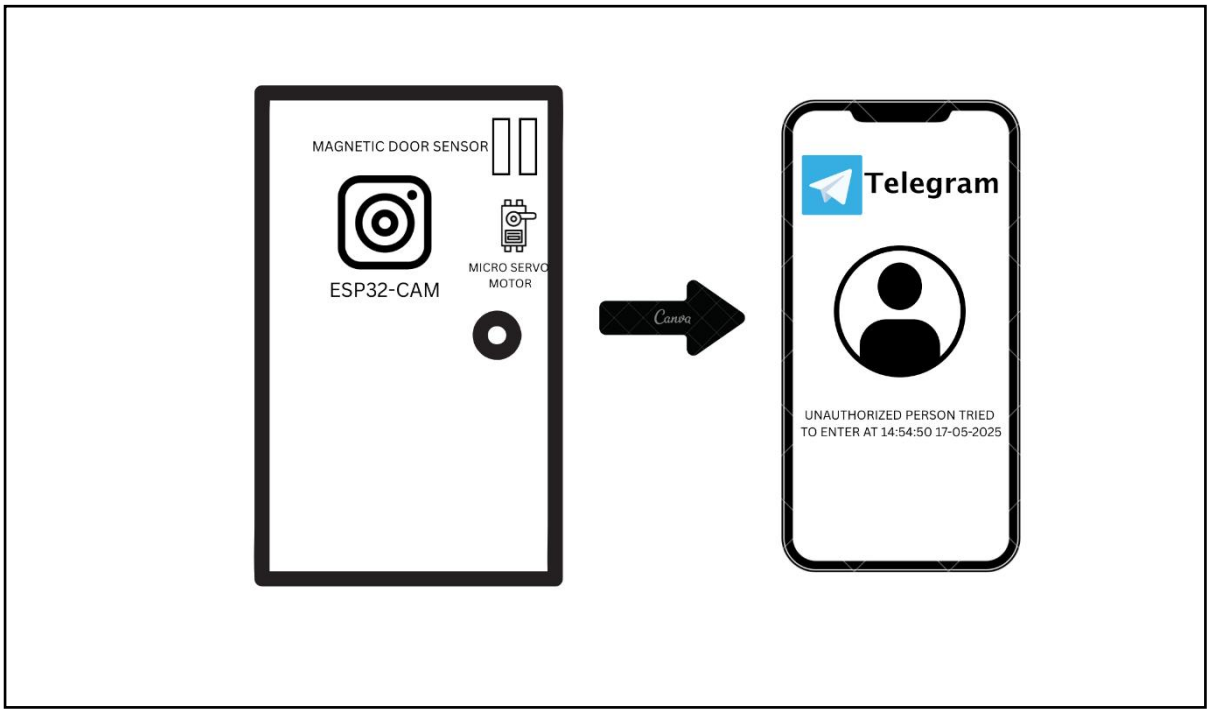

Figure 3.2: Hardware Design

Figure 3.2 shows the design of the hardware and the basic function of the system. The smart door monitoring system showcased in the image integrates several advanced hardware components to deliver enhanced security and functionality. At the core of the design is a compact camera embedded into the door frame. This camera provides live video streaming and surveillance, enabling users to monitor door activity in real-time. This feature is particularly valuable for identifying intrusions and verifying security events.

Complementing this is a magnetic door sensor located at the door's edge. This sensor detects whether the door is open or closed and acts as a primary trigger for activating the camera, capturing images, or sending notifications to the user.

The system emphasizes robust connectivity, with Wi-Fi as a standard for real-time data transmission and mobile app integration. This feature ensures scalability and makes the system suitable for deployment in remote locations.

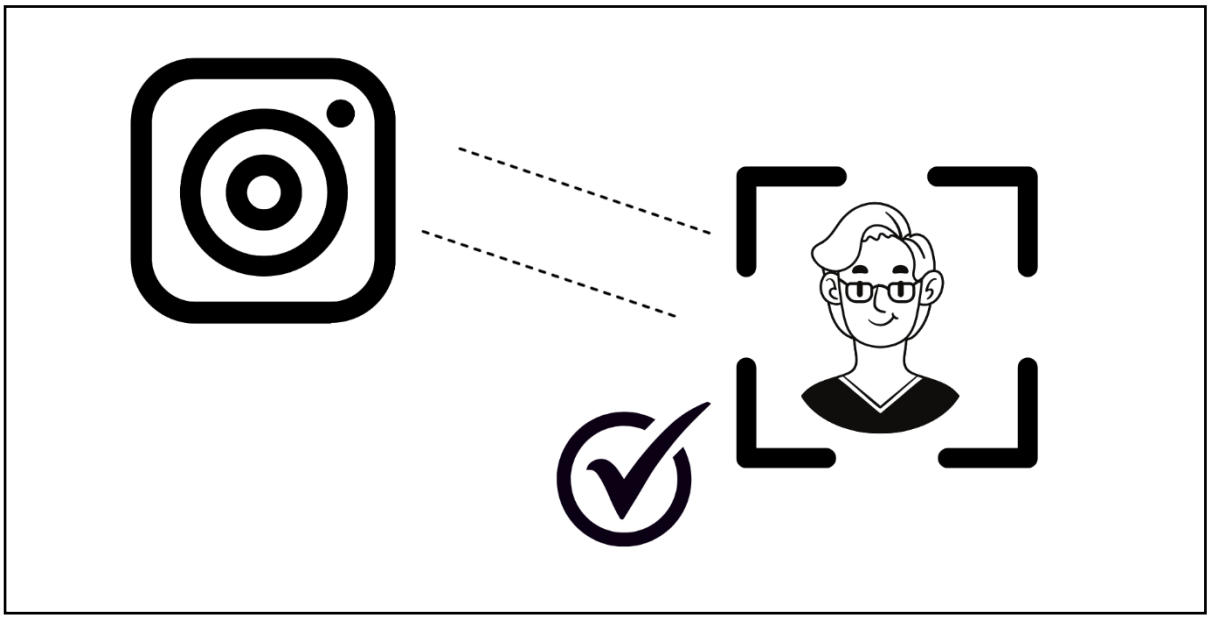

Figure 3.3: The registration part for enrol face

Figure 3.3 shows the face scanning process in the system is used during the registration phase. The ESP32-CAM module captures the facial image of the individual who wishes to be enrolled in the system. As the person stands in front of the camera, it processes their facial features, analyzing key components such as the eyes, nose, and mouth to create a unique facial profile. Once the image is captured, the system compares it to existing data to ensure it's a new face and not a duplicate.

If the facial scan is successful, meaning the system has successfully captured and analyzed the features, it indicates that the face enrollment for that user is complete. A "Hello Subject 0" message is displayed, confirming that the individual's face has been successfully registered in the system. This registration step ensures that the person's facial data is now stored and can be used for future identification whenever they approach the door. The facial recognition system

then allows for automatic authentication, enabling smooth and secure access control for the user.

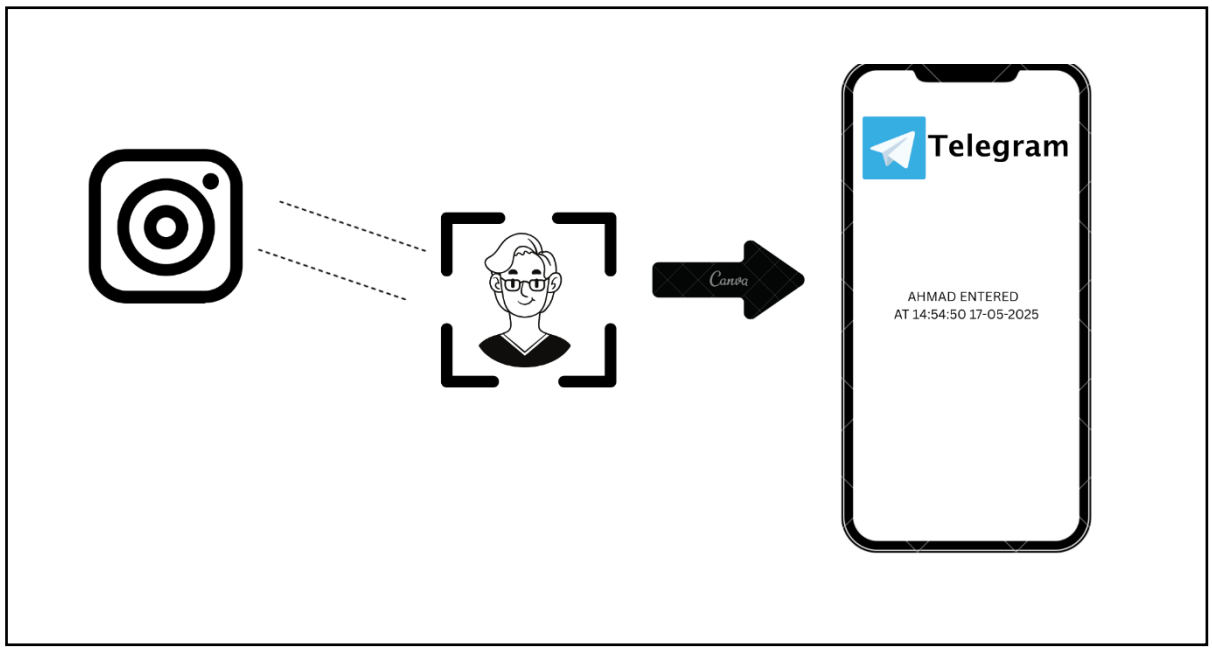

Figure 3.5: Approve access

The image 3.5 shows what happens after the user has successfully registered their face in the system. Once the person who has already enrolled approaches the door, the ESP32-CAM module scans their face again to confirm their identity. The camera captures the individual's facial features, such as their eyes, nose, and mouth, and the system compares this new scan with the facial data that was previously stored during the registration process. If the system finds a match between the scanned face and the enrolled profile, it grants the user access to the door by automatically authenticating them. At the same time, a notification alert is sent to a connected device, such as a phone or monitor, displaying a message like "AHMAD ENTERED AT 14:54:50 17-05-2025." This notification serves as a real-time confirmation that the person has been successfully identified and allowed access to the area. The notification also includes a timestamp, ensuring that the system keeps track of when the user entered. This entire process of facial recognition and automated authentication ensures both security and convenience, as

it allows the system to quickly and accurately grant access while maintaining detailed logs of all entries. This makes the access control system both efficient and reliable for everyday use.

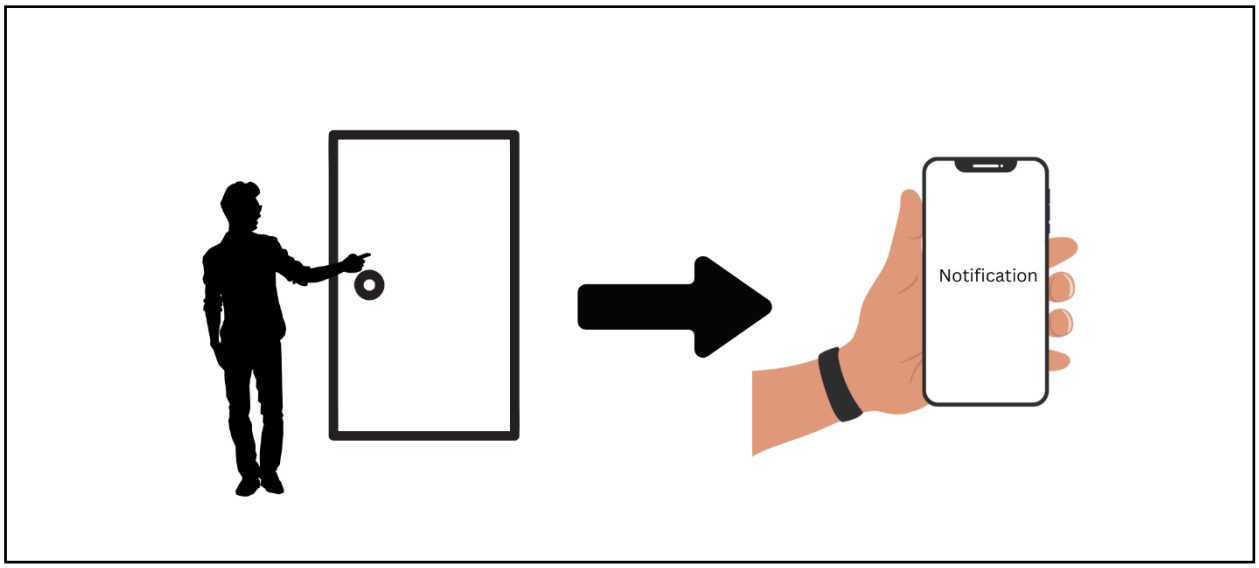

Figure 3.6: System operates to detect intrusions and notify the user

This figure 3.6 illustrates the process by which the system detects an intruder at the door and sends a notification to the user's smartphone. When someone interacts with the door, the system's sensors capture the activity and process it. The processed data is then transmitted as a real-time alert to the user's mobile device, ensuring they are promptly informed of any intrusion or event. This setup highlights the seamless integration of smart hardware with mobile technology to enhance security and user awareness.

A dedicated mobile application further enhances user experience, displaying real-time alerts, live image feeds, and data. Application integration offers secure storage and access to historical data, enabling users to review and analyze events as needed. This functionality ensures that users are always informed of security events, even when away from the premises.

3.3.2 User Interface Design

In this section, a simple design for the user interface will be displayed. The interface mainly shows the user the status of intrusion detection.

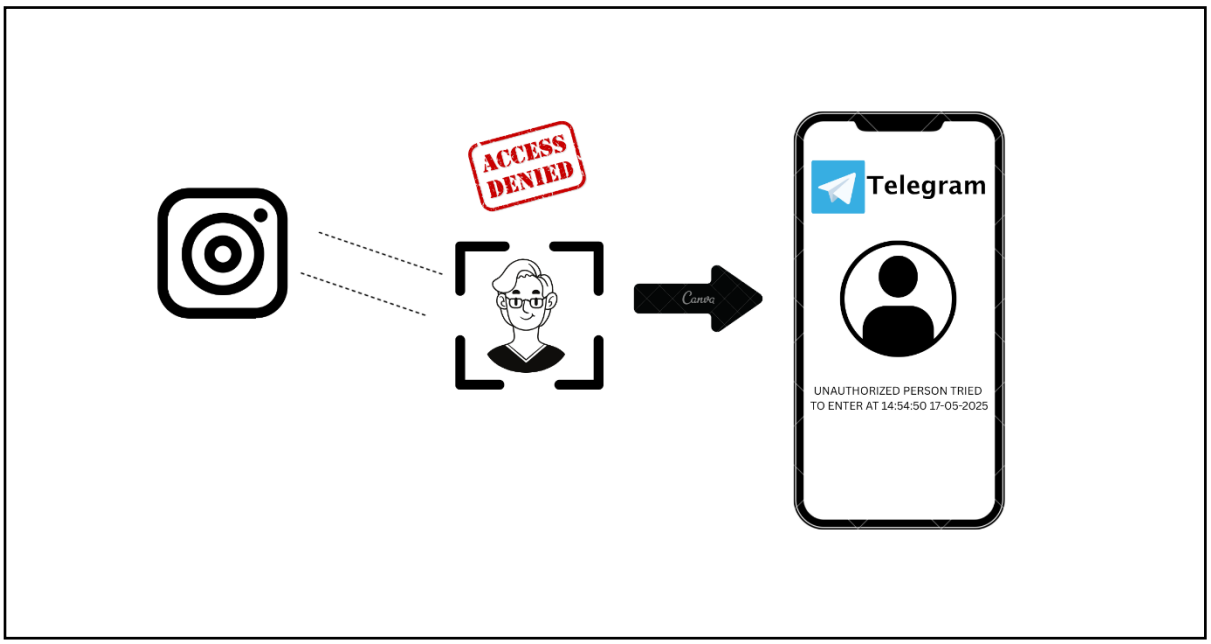

Figure 3.7: Real-time notification for unauthorized person

Figure 3.7 represents a real-time notification received on the Telegram app, which is part of the smart door monitoring system. The notification indicates that an intrusion has been detected. The app interface displays this alert prominently, with a message such as "Unauthorized Person!" alongside a timestamp to inform the user of the exact time of the event. Below the notification, the interface offers options to view image footage from the door's integrated camera or access the event history for a detailed review of past incidents. In the background, the smartphone is held by a user, and a modern office door is visible, emphasizing the real-world application of the system. Connectivity symbols, such as Wi-Fi, are also shown, highlighting the system's reliance on wireless communication for real-time updates.

The image captures the essence of how the system ensures user awareness and swift action in response to potential security breaches, making it an effective IoT-based monitoring solution. It also emphasizes user convenience with a clean, professional interface tailored for quick and intuitive interaction.

3.4 Development

In this phase, the prototype system for this project will be developed. There will be three aspects of development, which are hardware development, software development, and Arduino IDE programming. The table below outlines the approach to developing the system. It provides a concise explanation of how each component will be integrated and implemented to ensure the system operates effectively and meets its intended functionality.

Table 3.1: Development Approach for System Components

Aspect	Explanation
Hardware Development	- Integrated ESP32-CAM, servo motor, magnetic door sensors, and power supply - Include proper wiring, sensor placement, and testing.
Software Development	- Develop a user-friendly interface for real-time monitoring and notifications. - Integrate with a mobile app, Telegram.
Arduino IDE Programming	- Write and upload firmware ESP32-CAM and other components. - Enable functions like image capture, motion detection, door state detection, and real-time alerts.

The table explains the system development process, focusing on hardware assembly, software integration for user interaction, and programming to enable seamless communication and functionality. This phase will be further discussed in Chapter 4.

3.5 Test

The primary objective of the testing phase is to ensure that the Intrusion Detection and Notification System functions correctly under various scenarios. The testing process validates that the sensors, notification system, and processing logic operate seamlessly to deliver real-time responses. Additionally, it examines edge cases such as false alarms and sensor malfunctions to guarantee system reliability.

3.5.1 User Acceptance Testing (UAT)

A crucial stage in the testing process is User Acceptance Testing (UAT), which verifies that the system satisfies user needs and performs well in practical situations. UAT will be used for the project to verify that the system can identify intrusions, notify the user's mobile application, and store and retrieve event data in the application. Verifying that the system functions as planned and offers a flawless user experience is the aim.

A team of end users, including testers, will engage with the system during UAT. Users will take on real-world situations, such as multiple sensor activations, invasions, and edge cases like poor connectivity or malfunctioning sensors. Users will assess if the application storage correctly stores and retrieves event data, whether alerts are received promptly, and whether the mobile application interface is easy to use during these tests. Additionally, the testing will evaluate the system's general dependability and performance in various scenarios.

Users will be watched while they interact with the system in order to gather feedback. In addition to noting any delays in notice delivery or difficulties finishing tasks, observers will keep an eye out for behaviors like hesitation, uncertainty, or errors while interacting with the interface. A greater comprehension of how users interact with the system is ensured by this real-time observation method, which offers unfiltered feedback based on actual user behavior. To detect any problems with performance or usability, the observations and results will be examined. For instance, issues like customers experiencing confused app navigation or delayed

notifications will be fixed. This feedback will subsequently be used to improve the system's usability and performance.

The system will have undergone extensive testing to make sure it satisfies user expectations by the end of UAT. To make sure it is prepared for deployment, it will be verified that it can accurately and promptly detect intrusions, provide a user-friendly experience through its mobile app capabilities, and give accurate and timely notifications.

3.6 Release

The release phase focuses on transitioning the system from development to real-world use. It begins with deployment preparation, including finalizing software configurations, optimizing the Arduino code, and testing the mobile app. Comprehensive documentation, such as installation guides and troubleshooting instructions, is also prepared to support users and administrators. Deployment involves setting up the system in the user's environment, such as installing sensors and cameras or configuring software on user devices. Initial user training is conducted to familiarize them with the system's features, including monitoring alerts, accessing data, and managing notifications. This will be discussed in Chapter 5.

3.7 Feedback

The feedback phase is crucial for evaluating the system’s performance, usability, and effectiveness after release. User input is collected through surveys, interviews, and system performance metrics to identify areas for improvement. This phase focuses on fixing bugs, enhancing usability, and incorporating new features to ensure the system evolves to meet user needs. By valuing user feedback and acting on their suggestions, the system becomes more user-friendly and efficient. This phase highlights the importance of continuous improvement and user collaboration. This will be discussed in Chapter 5.

3.8 Overall Flow of the Project

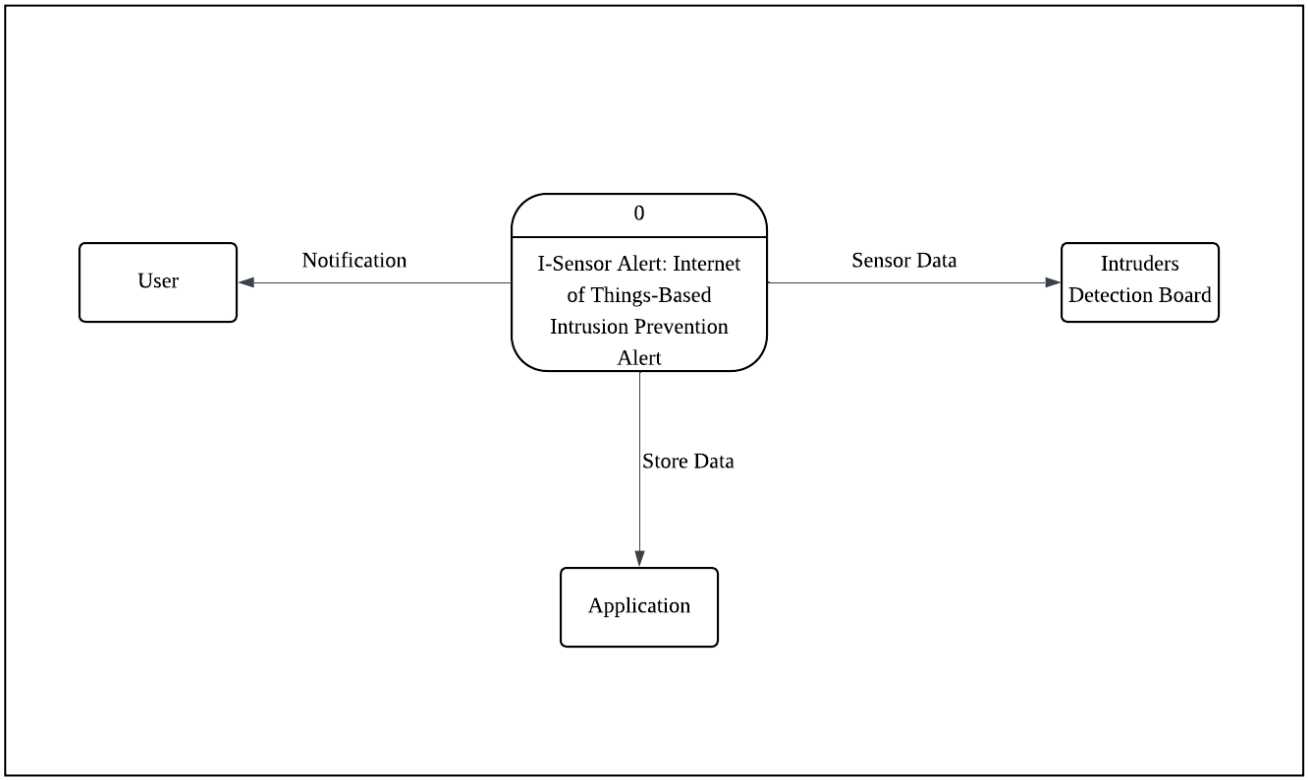

Figure 3.8: Context diagram of the system

Figure 3.8 shows the flow of the system for I-Sensor Alert: Internet of Things-Based Intrusion Prevention Alert. The provided image is a context diagram that gives an overview of the system's interactions with external entities. At the center of the diagram is the main system, labeled as "0," which represents the Intruder Detection System. This system is responsible for processing inputs from external components, managing data, and delivering outputs to other entities.

One of the key external entities in the diagram is the Intruder Detection Board, which includes hardware components such as sensors and possibly an ESP32-CAM. This board detects intrusions or events and sends the relevant data to the central system for processing. The system then analyzes this data to determine the appropriate actions, such as generating alerts or storing event logs.

The system also interacts with an application service, enabling the storage and retrieval of processed data. This connection ensures that event logs and captured information are securely saved and can be accessed later for analysis or record-keeping.

The final external entity in the diagram is the User, who interacts with the system to receive notifications and alerts. Processed information from the system is sent directly to the user, keeping them informed about detected events in real time. This ensures that the user remains aware of any intrusions or security concerns.

The arrows in the diagram indicate the flow of data between the system and these external entities. Data flows from the detection board to the system, from the system to the application for storage, and from the system to the user for notifications. This context diagram provides a high-level view of how the system operates and integrates with its key components, highlighting its primary functions and interactions.

The following diagram will show the system's processes.

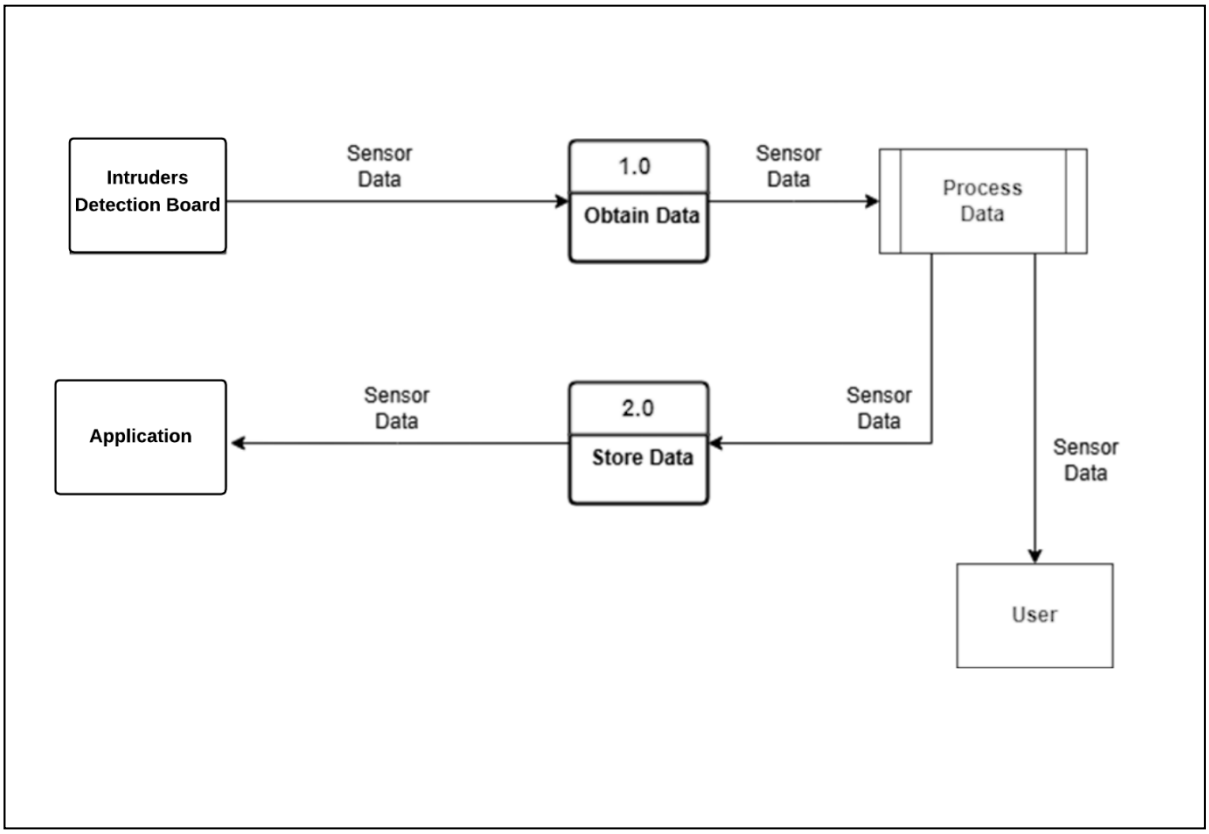

Figure 3.9: DFD Level-0

The DFD Level-0 diagram provides a high-level overview of the major processes within the system and their interactions with external entities. It represents the flow of data from detection to processing, storage, and notification, ensuring the system operates effectively and meets user requirements.

The diagram includes three key external entities. The Intruders Detection Board serves as the hardware component responsible for detecting intrusions or events. It sends collected intrusion data to the system for processing. The Application is a storage platform where processed data is saved for future access or analysis. The User is the end-user who interacts with the system by receiving real-time notifications and alerts about detected intrusions.

The system consists of two main processes. The 1.0 Process Data function receives intrusion data from the detection board and analyzes it to identify significant events. This process ensures that only relevant and meaningful data is passed to the next stage. The 2.0 Store and Notify

process handles two critical tasks: storing processed data in the application and delivering notifications to the user to keep them informed about security events promptly.

The data flow in the diagram begins with the Intruders Detection Board, which sends intrusion data to 1.0 Process Data for analysis. The processed data is then passed to 2.0 Store and Notify, where it is sent to the Application for storage and simultaneously to the User as a real-time notification.

This diagram effectively outlines the flow of information within the system, highlighting how data is detected, processed, stored, and communicated to the user, ensuring seamless functionality and user awareness.

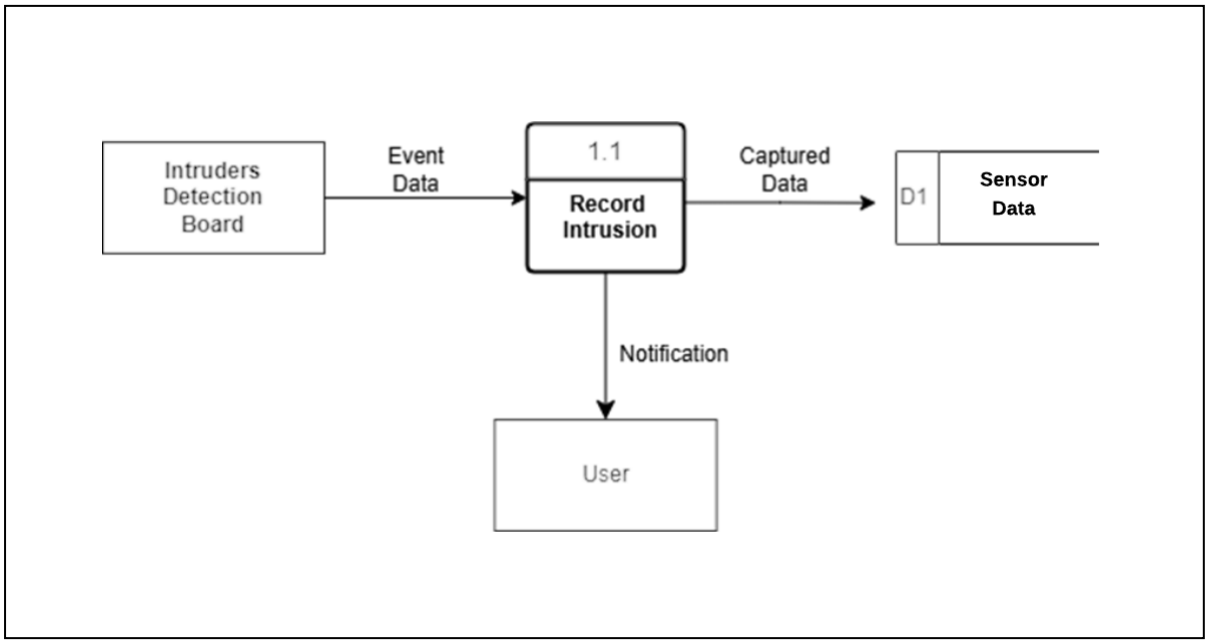

Figure 3.10: DFD Level-1

The Data Flow Diagram (DFD) illustrates the process of recording intrusions within a security system. The Intruders Detection Board serves as the input source, detecting intrusion events and sending the corresponding event data to the central process, labeled 1.1, “Unauthorized Person." This process is responsible for logging the details of the detected intrusion. The relevant captured data is then stored in a data store (D1) for future reference. Simultaneously,

the system sends a notification to the user to alert them of the detected intrusion. This diagram highlights how the system efficiently monitors intrusions, records important data, and notifies users for immediate action.

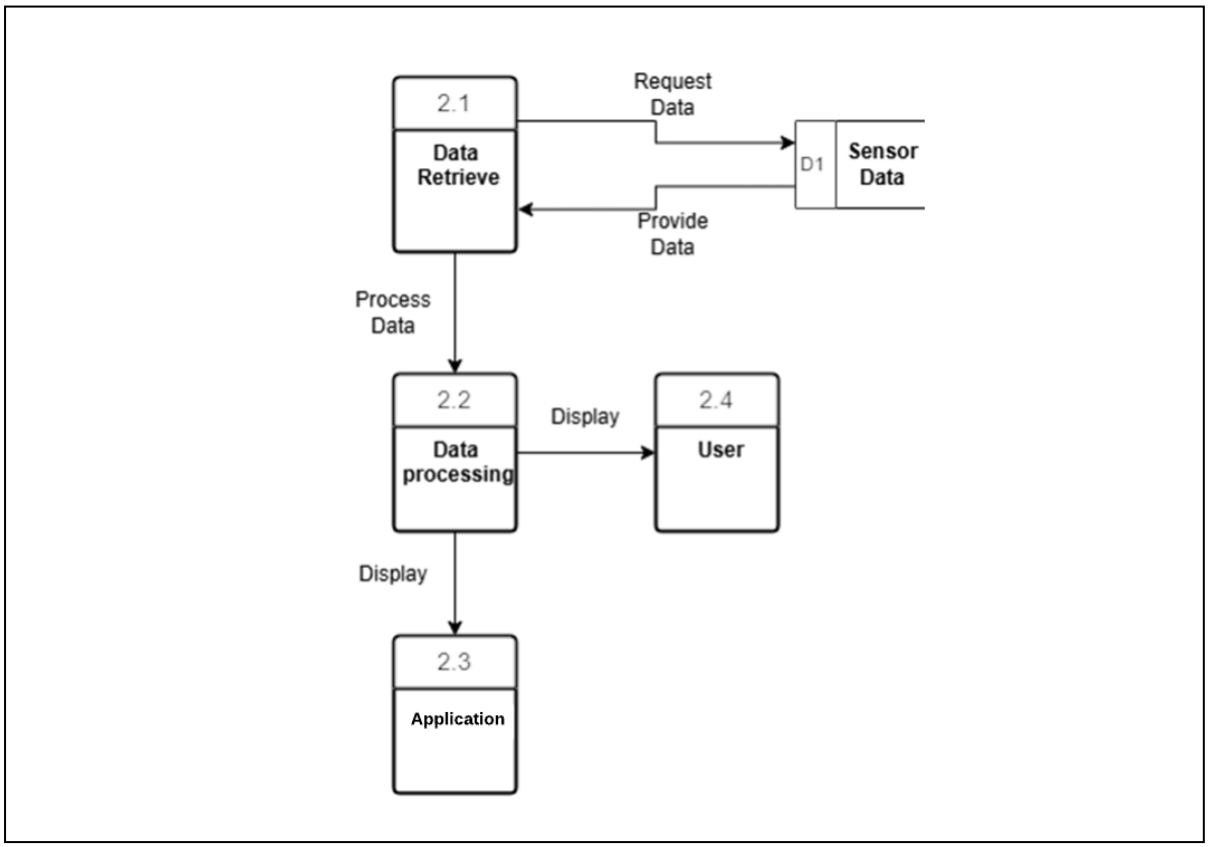

Figure 3.11: DFD Level-2

This diagram illustrates the detailed breakdown of DFD Level-2 in the system, which focuses on storing data and notifying the user. It builds upon the high-level view provided in DFD Level-0, showing the sub-processes involved in managing notifications and securely storing intrusion event data.

The process begins with 2.1 Receive Processed Data, which acts as the entry point for this stage. This sub-process is responsible for receiving processed data from earlier stages of the system, ensuring a seamless flow of information. Once the data is received, it moves to 2.2 Notify User, where notifications are sent to the user in real-time. These notifications alert the

user about detected intrusions or security events via channels such as mobile applications, SMS, or other communication methods.

Simultaneously, the system prepares the processed data for storage in 2.3 Prepare for Storage. This sub-process ensures the data is organized, formatted, and optimized for secure storage. After preparation, the data is transferred to 2.4 Store Data, where it is securely saved in the data store labeled D1. This storage unit serves as the repository for all processed event logs, allowing for future retrieval and analysis.

The data flows between these sub-processes maintain a balance between real-time communication and long-term data management. Users are promptly informed about security events while the system retains accurate and organized records for future use. This detailed view of Process 2.0 highlights the system's ability to manage data effectively and ensure both immediate notifications and reliable data storage.

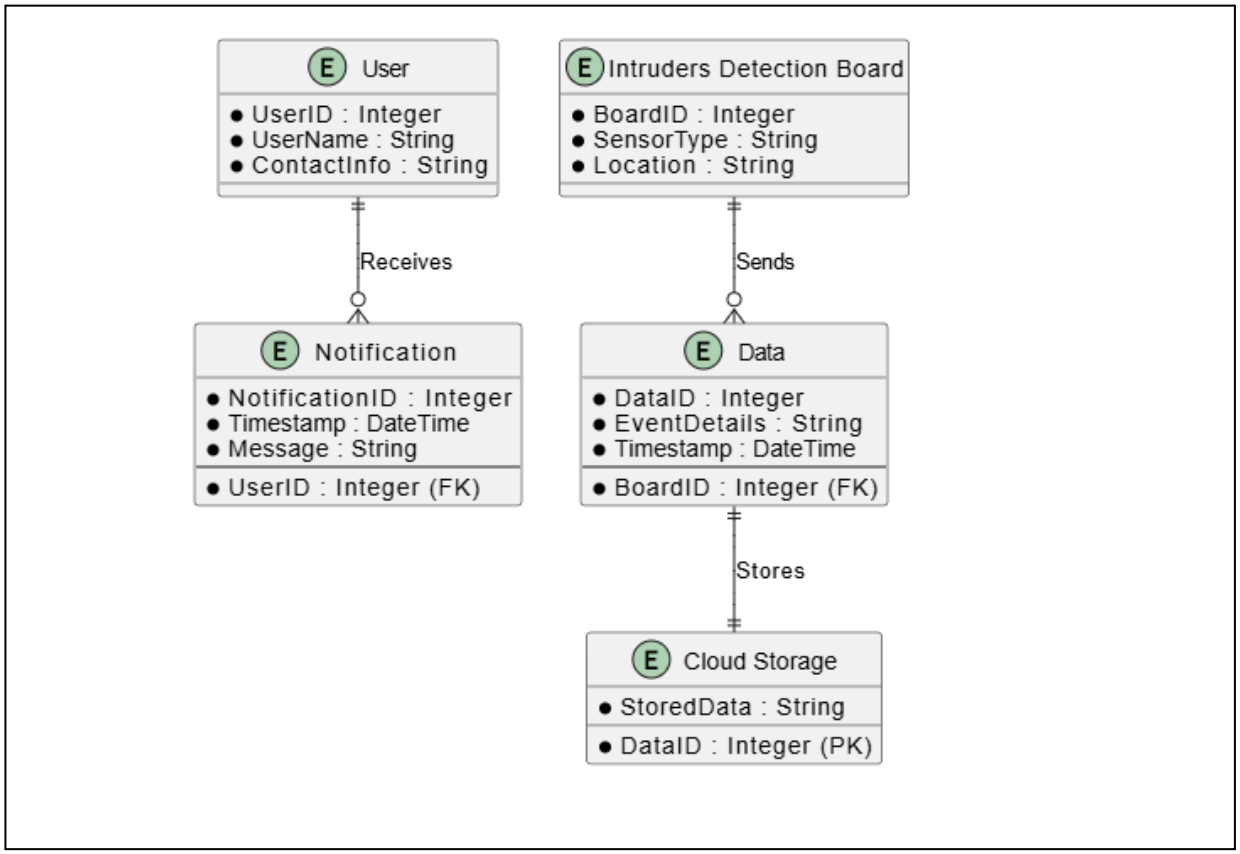

Figure 3.12: ER-Diagram

This Entity-Relationship Diagram (ERD) illustrates the relationships between the key entities in the system and their roles. The User entity represents individuals who receive notifications about detected intrusions. Each user is uniquely identified by UserID and is linked to multiple notifications through the Notification entity, which contains information such as the message, timestamp, and user association.

The Intruders Detection Board represents the hardware responsible for detecting events. It is uniquely identified by BoardID and includes details like the sensor type and location. The detection board generates event data, represented by the Data entity. The Data entity contains attributes such as EventDetails and Timestamp and links back to the detection board through the BoardID.

The processed data is stored in Application Storage, which securely saves the event information for future access. The DataID serves as the primary key linking the data to the application. Notifications generated from this data are sent to the users to alert them of any detected intrusion.

This diagram provides a clear and efficient representation of how data flows between users, detection hardware, and the application storage system. It highlights the seamless integration of components to ensure real-time notifications and reliable data storage.

3.9 Chapter Summary

This chapter highlights the adoption of The Agile Model for developing the system, emphasizing its simplicity and systematic approach. The iterative nature of the agile model ensures that each development step is organized and adaptable, allowing for adjustments based on evolving requirements. Key aspects of the chapter include the requirement analysis, which identified the system's needs, such as integrating hardware components like the ESP32-CAM and sensors and providing real-time notifications, reliable data storage, and seamless user interaction.

Additionally, the design phase detailed the system's architecture using diagrams, such as the context diagram and DFDs Level-0, Level-1 and Level-2, to illustrate data flow and interactions between components. These diagrams outline processes like intrusion detection, user notification, and data storage in the application. The chapter also introduced User Acceptance Testing (UAT) as a crucial step to validate the system's functionality, with further development and testing to be discussed in later chapters using real-world data.

CHAPTER 4: IMPLEMENTATION

4.1 Introduction

This chapter will focus on the processes involved in developing the prototype as well as the hardware development for users to detect the intruders. The entire implementation of this system was according to the requirements analysis and design mentioned in the previous chapter. Hence, there will be the development of the functionalities and flow of the entire system. The hardware connection and tool requirements, installation of tools will be further discussed on this chapter.

4.2 System and Tool Requirements

The system and tool requirements needed for the connection are shown below:

- Operating System (Windows 8 / 10 / 11)
- Arduino IDE
- Telegram (Platform for notification)
- Stable Internet Connection

4.3 Software Installation and Configuration

The primary installation and configuration of the system required by the system will be discuss in the following points.

4.3.1 Arduino IDE

The Arduino IDE is a platform that acts as a bridge for communication between the microcontroller and the PC. The programming code will be coded in a logical way for the microcontroller to function normally and be able to upload to it through a USB cable. This software is easy to install and code.

Arduino IDE can be downloaded and installed from the Microsoft Store on every Windows computer. Just type Arduino IDE in the search bar of the Microsoft Store. To download and

install, click on the “get” button. After clicking on the get button, it will download and install automatically.

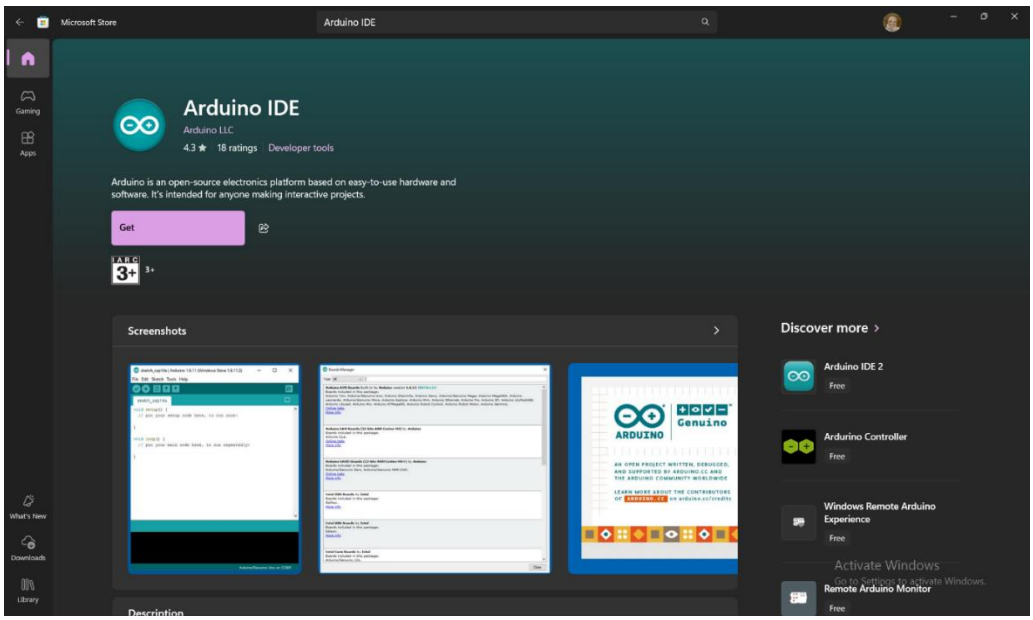

Figure 4.1: Microsoft Store

When the installation is completed, Arduino IDE will appear on the start menu where developer can open the software and start working on the configuration.

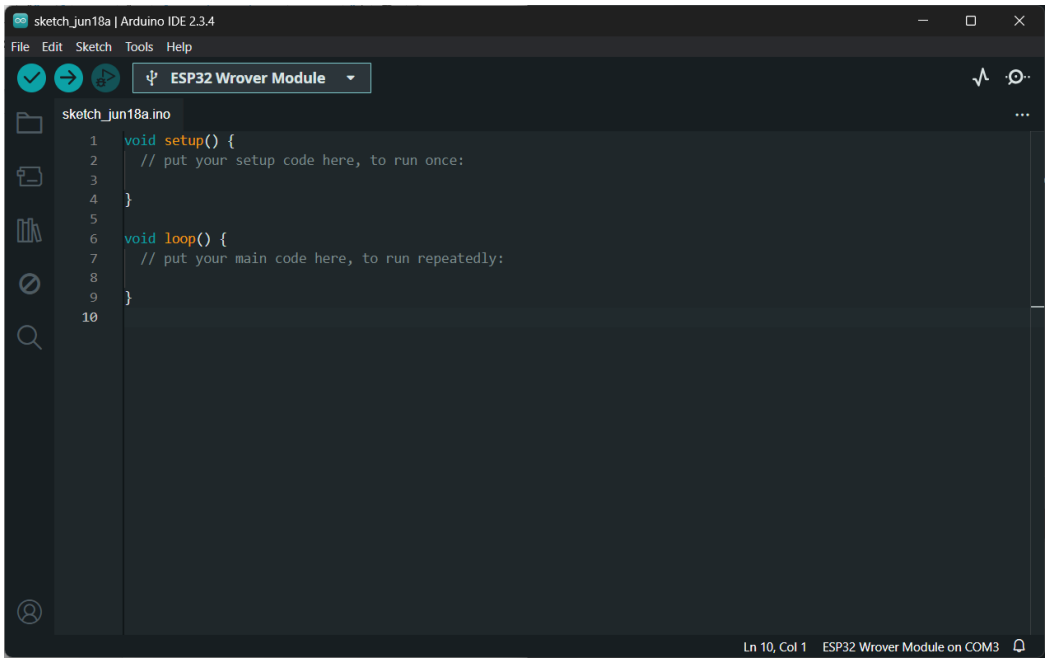

Figure 4.2: Arduino IDE

As the microcontroller being used ESP32 Wrover Module. Special configurations will be needed for successful in uploading the codes for the microcontroller to function. Click on File at the window bar > Preferences as shown below.

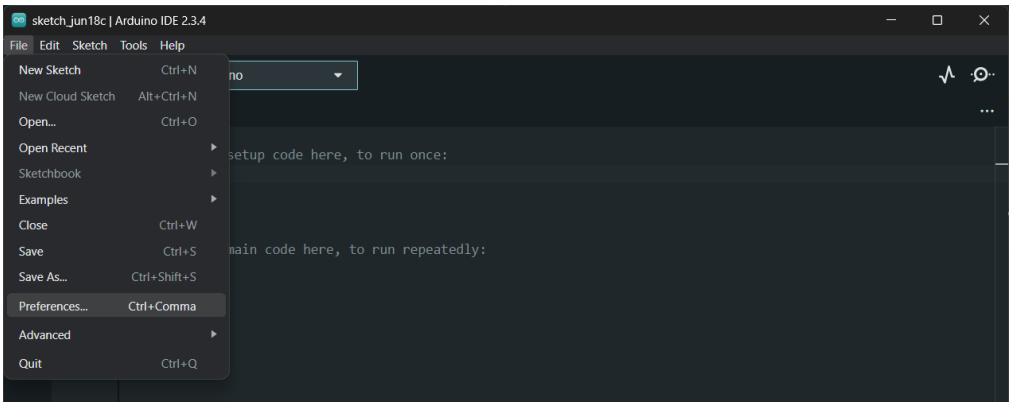

Figure 4.3: Preferences in Arduino IDE

Key in or paste the link " https://dl.espressif.com/dl/package_esp32_index.json , https://espressif.github.io/arduino-esp32/package_esp32_index.json" into the Additional Boards Manager URLs: and click OK.

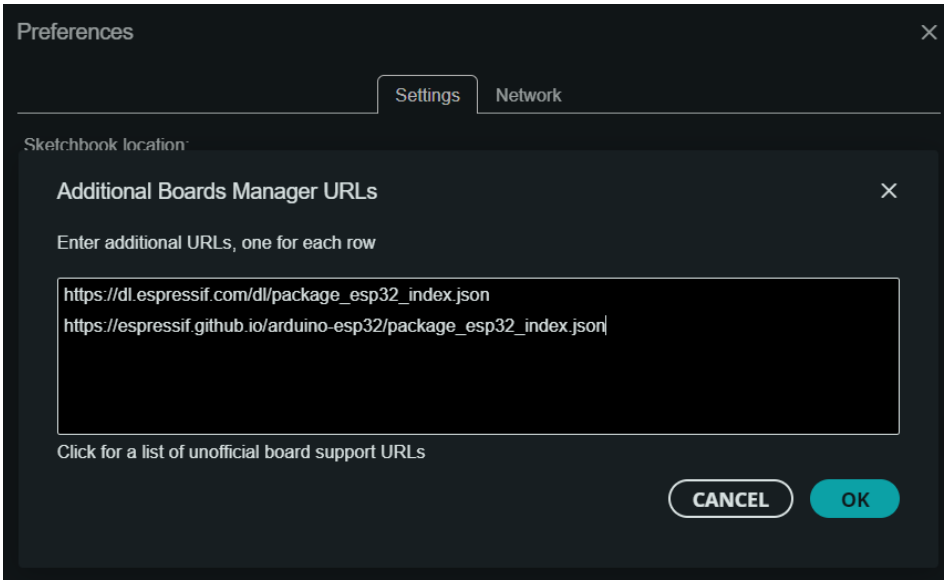

Figure 4.4: Add in link under Additional Boards Manager URLs

Next, click on the Tools on the window bar of Arduino IDE > Board:… > Boards Manager as shown in the figure below.

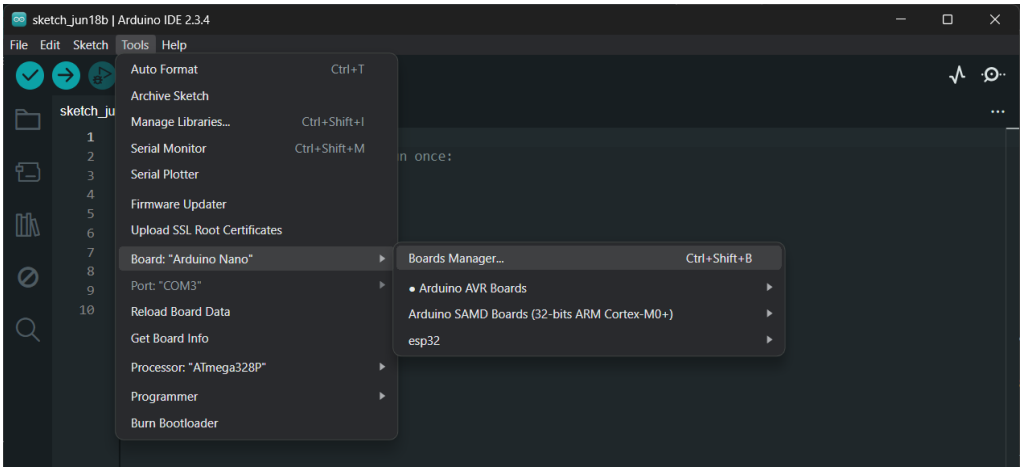

Figure 4.5: Enter into Boards Manager

Then type ESP32 in the search bar as shown below. After searching, install the board of ESP32 by Espressif System.

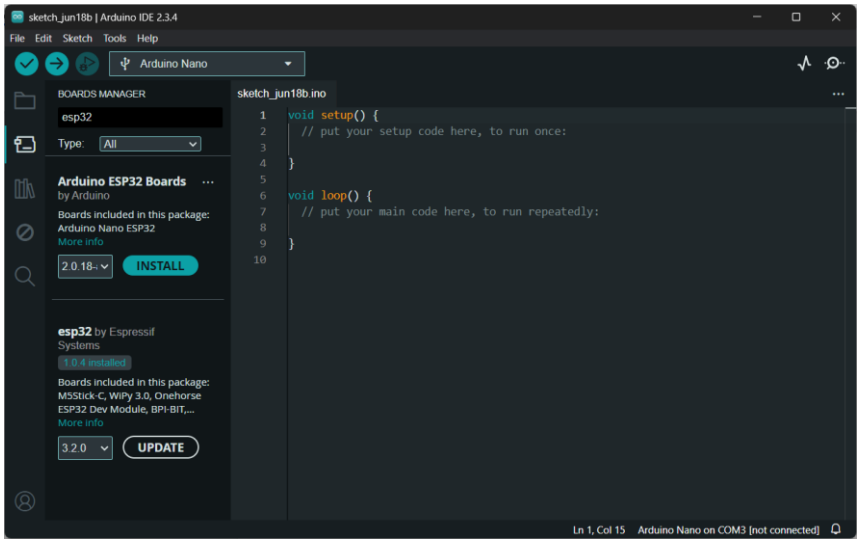

Figure 4.6: Installation for ESP32

Next, open Library Manager and search for Universal Telegram and choose "UniversalTelegramBot" by Brian Lough, and install it.

The UniversalTelegramBot library allows an Arduino-based project to interact with Telegram, a widely used messaging application. It simplifies the process of sending messages, images, and other updates to a Telegram bot directly from the Arduino project. One of the key features of this library is its ability to send text messages to a Telegram bot, which is particularly useful for sending alerts or notifications, such as informing the user when an intruder is detected by the system. Additionally, the library enables the sending of photos, allowing images captured by the camera module (e.g., the ESP32-CAM) to be sent directly to a Telegram chat. This enhances the security system by providing real-time images in case of an event. Furthermore, the library supports receiving commands from users, enabling the system to interact with users via Telegram, offering two-way communication. In the context of this project, the UniversalTelegramBot library is essential for sending timely alerts, such as notifications for face recognition or unauthorized access, and for integrating the IoT system with a messaging service, ensuring real-time communication and control.

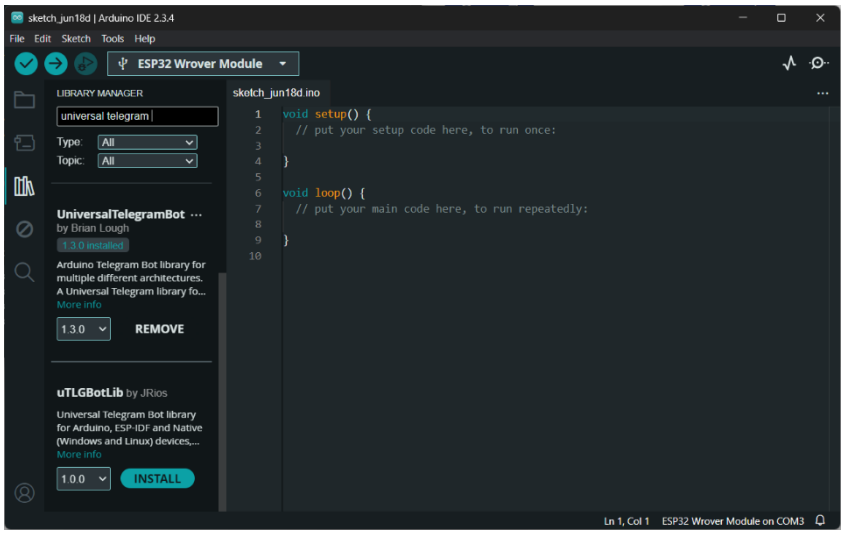

Figure 4.7: Install Telegram Bot in Arduino IDE

To complete the part of ESP32, click on Tools > Board:… > select ESP32 > Select ESP32 Wrover Module as shown in the figure below.

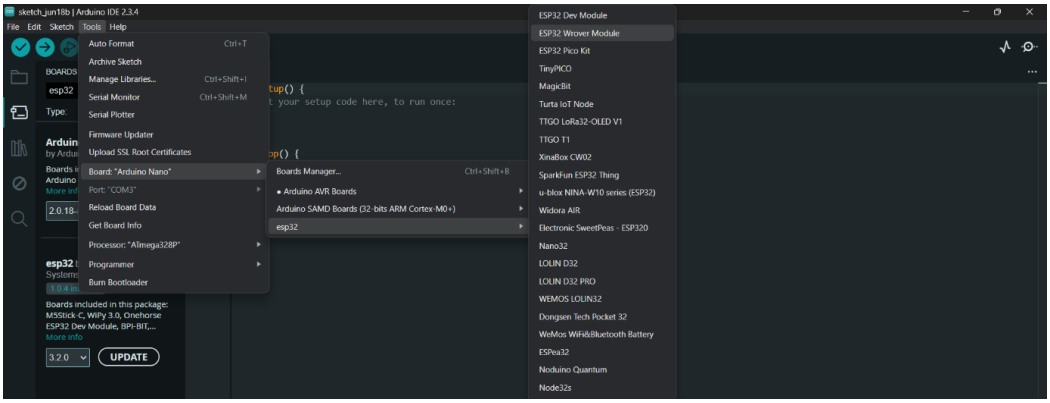

Figure 4.8: Selecting a board for file uploading

4.3.2 Telegram (Platform for notification)

In this system, Telegram will act as a platform for communication between the microcontroller and users. The function used will be the alert messaging function. To use this function, certain configurations will be needed. These steps to complete the configuration will be discussed in the following points.

First, type Telegram in the search bar in the Microsoft Store and click “get” to install.

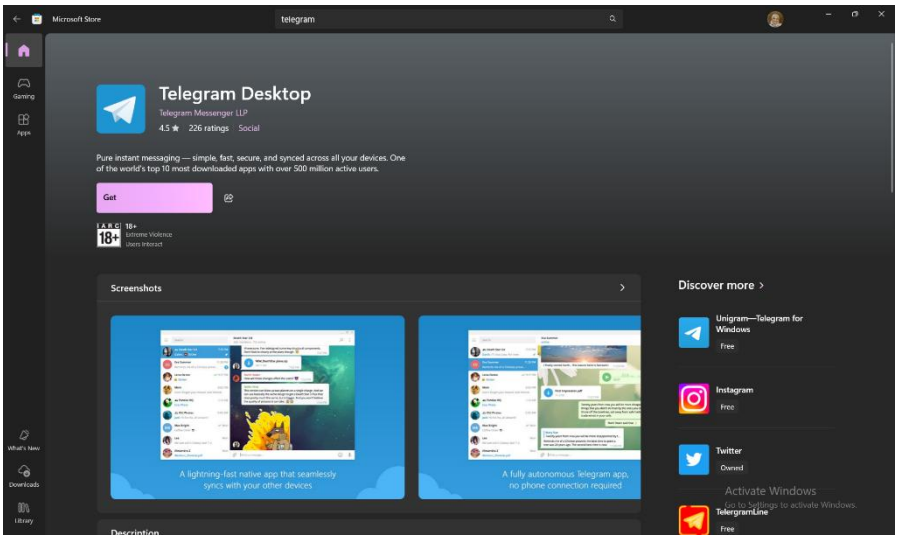

Figure 4.9: Microsoft Store

When Telegram is successfully installed, search in the Telegram bar “BotFather” to create chat chatbot for the alert notifications. Choose “newbot”, then create a name for the bot, and the bot is successfully created. Telegram sends the API key for the bot, that API key is used for configuration in Arduino IDE.

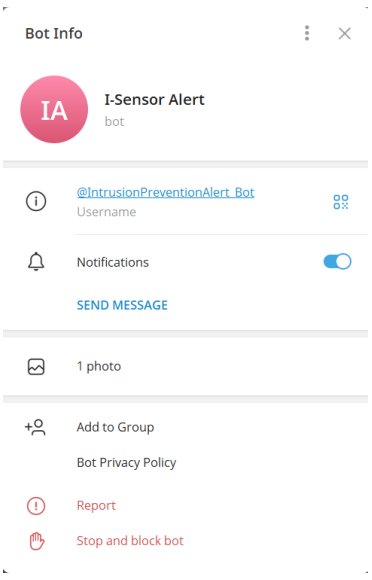

Figure 4.10: I-Sensor Alert Bot

4.4 Hardware Installation

4.4.1 ESP32-CAM

The microcontroller that is used in this system will be the ESP32-CAM, where the camera module comes together. It is simple and straightforward to connect. Furthermore, a magnetic door sensor, buzzer and a micro servo motor will be connected to this microcontroller for the functioning of the system. The sensors have been discussed in the previous chapter; hence, the focus will be on ESP32-CAM and the connectivity between it and the sensors. Figure 4.11 shows the actual image of ESP32-CAM, and Figure 4.12 displays the diagram of the pins of ESP32-CAM.

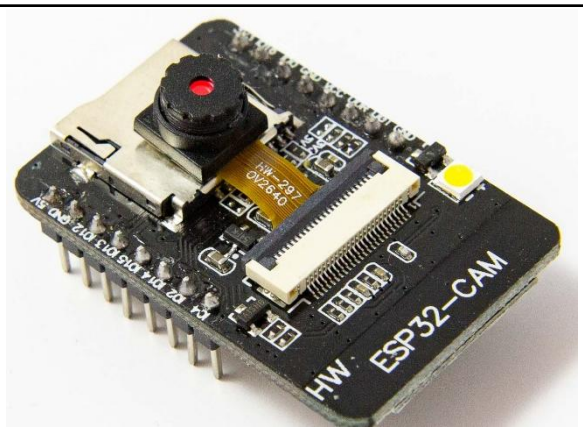

Figure 4.11: ESP32-CAM

Figure 4.12: ESP32-CAM Pins

Figure 4.11 shows the connectivity of the entire hardware component. The magnetic sensor is connected to GPIO13, which has 2 outputs on it, Ground and Signal. The Ground output is connected to GND, and the Signal output is connected to GPIO13. For the micro servo motor as shown in Figure 4.12, there are 3 outputs which are control, VCC, and GND. The connectivity for control is connected to GPIO2, for VCC output connected to 5V, and for GND connected to GND. The buzzer is soldered with ESP32-CAM. The push button is connected to GPIO15 and GND. A USB to microUSB 2.0 cable is connected for the power supply.

Figure 4.13: Micro Servo Motor

Figure 4.14 represents the finalized version of the hardware connectivity diagram, showcasing the complete and fully integrated setup of all components and how they interact with each other within the system. This diagram illustrates the precise connections and configurations of the hardware, providing a clear overview of how the different elements are linked together to ensure the proper functioning of the system.

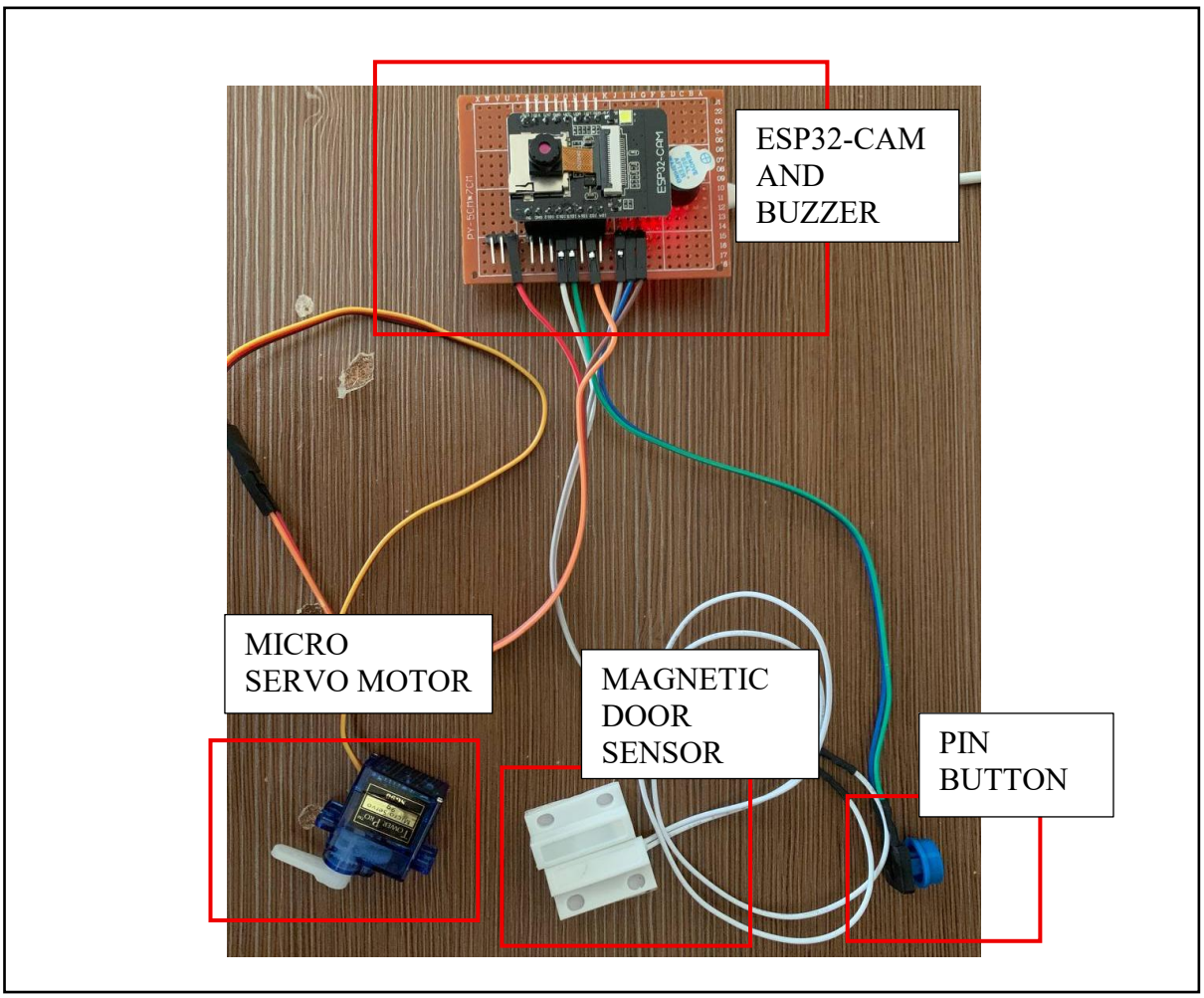

Figure 4.14: Completed hardware connection

4.4.2 Wi-Fi Connectivity Configuration

As Wi-Fi is embedded in ESP32-CAM, there is no hardware configuration needed. However, certain codes will need to be included in the Arduino IDE for the Wi-FI to function. The coding part will be explained as below.

First of all, declaring SSID, Password, API key Telegram, and ID Telegram. The API key Telegram and ID will be the medium of communication through which data will be sent to. The figure shows the syntax of the declaration.

```
const char* WIFI_SSID = "Fana";
const char* WIFI_PASS = "nazifaaa";
const String BOT_TOKEN =
"8103958477:AAF7bzCdmhY0pz4Ly8Z3sxNevhsP5zrjhgs";
const String CHAT_ID = "1399117593";
```

Figure 4.15: SSID, Password, API Key and ID for Wifi

```
void connectToWiFi() {
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  Serial.print("Connecting to WiFi");

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
```

Figure 4.16: Wi-Fi Connecting and checking

To further check the Wi-Fi connection, the user connecting and notification sending process will be added as the system requires connecting to the user for notification sending. If it fails to connect, one of the factors would be Wi-Fi connectivity failure. Then, Figure 4.17 would be a notification sent to the Telegram Bot. If the notification is successfully sent to the Telegram Bot, it confirms that the Wi-Fi connectivity is working.

```
Serial.println("Connecting to " + String(telegramDomain));


  if (clientTCP.connect(telegramDomain, 443)) {
    Serial.println("Connection successful");


    String head = "--RandomNerdTutorials\r\nContent-Disposition:
form-data; name=\"chat_id\"; \r\n\r\n" +
                  CHAT_ID + "\r\n--
RandomNerdTutorials\r\nContent-Disposition: form-data;
name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type:
image/jpeg\r\n\r\n";
    String tail = "\r\n--RandomNerdTutorials--\r\n";


    uint16_t imageLen = fb->len;
    uint16_t extraLen = head.length() + tail.length();
    uint16_t totalLen = imageLen + extraLen;


    clientTCP.println("POST /bot" + BOT_TOKEN + "/sendPhoto
HTTP/1.1");
    clientTCP.println("Host: " + String(telegramDomain));
    clientTCP.println("Content-Length: " + String(totalLen));
    clientTCP.println("Content-Type: multipart/form-data;
boundary=RandomNerdTutorials");
    clientTCP.println();
    clientTCP.print(head);
```

Figure 4.17: Code for Telegram checking

4.4.3 Servo Control Connectivity

The servo control code is used to open and close the door in the system using a servo motor. A servo motor is a special type of motor that can rotate to a specific angle based on the input given. The code first sets up the servo by defining the pin it's connected to and the frequency at which it operates. It uses PWM (Pulse Width Modulation) to control the angle of the servo. The function setServoAngle takes an input angle (from 0 to 180 degrees) and converts it into a PWM signal, which tells the servo motor how far to rotate. When the door needs to be opened, the servo is moved to a 90-degree position, and when it needs to be closed, the servo goes back

```
void initializeServo() {
  ledcSetup(SERVO_PWM_CHANNEL, SERVO_PWM_FREQ,
SERVO_PWM_RESOLUTION);
  ledcAttachPin(SERVO_PIN, SERVO_PWM_CHANNEL);
  setServoAngle(0); // Initialize to closed position
}
void setServoAngle(int angle) {
  // Convert angle (0-180) to PWM duty cycle (500-2400µs)
  int pulseWidth = map(angle, 0, 180, 500, 2400); // Microseconds
  // Calculate duty cycle: (pulseWidth * 2^resolution) /
(1,000,000 / frequency)
  uint32_t duty = (pulseWidth * (1 << SERVO_PWM_RESOLUTION)) /
(1000000 / SERVO_PWM_FREQ);
  ledcWrite(SERVO_PWM_CHANNEL, duty);
}
```

Figure 4.18: Code for Servo Control

4.4.4 Magnetic Switch Connectivity

The magnetic switch in the system acts as a security feature to detect if intruders is trying to open the door without authorization. This switch is placed on the door and checks whether it's being tampered with. The code reads the status of the magnetic switch connected to a specific pin. If the switch is triggered (meaning the door is being opened or someone is trying to break in), it turns on a buzzer to alert the user and sends a notification through Telegram to inform them of the suspicious activity. The system also makes sure to reset the alert if the door is closed or no longer tampered with, and it turns off the buzzer. This ensures that the door stays secure and the user is always notified of any potential break-ins. Below is the figure 4.18 snippet code for the magnetic switch.

```
void handleSecuritySwitch() {
  int securitySwitch = digitalRead(SW_SECURITY_PIN);
  if (securitySwitch == HIGH) {
    if (!doorOpen) {
      digitalWrite(BUZZER_PIN, HIGH);

      if (securityAlertCounter == 0) {
        securityAlertCounter = 1;
        // Uncomment to send alert message
        sendTelegramNotification("Someone is trying to break
in");
        Serial.println("Someone is trying to break in");
      }
    }
  } else {
    if (doorOpen) {
      doorOpen = false;
    }
    securityAlertCounter = 0;
```

Figure 4.18: Code for Magnetic Switch

4.4.5 Face Enroll Configuration

This code is designed to set up an ESP32 camera module for face enrollment. First, the camera model is selected, in this case, the AI Thinker camera module, by defining CAMERA_MODEL_AI_THINKER. Other camera models are available but are commented out. The camera is connected to the ESP32 through specific pins, which are set in the

camera_pins.h file. These pins control the data, clock, and other functions of the camera. Then, the Wi-Fi credentials (SSID and password) are provided to connect the ESP32 to the internet. The code waits until the Wi-Fi connection is established, and once it's connected, it prints a message on the serial monitor to let us know the device is ready.

Next, the camera is initialized using the esp_camera_init() function. If there's an error with the camera setup, it stops and prints an error message. After initialization, the camera sensor settings are adjusted, like flipping the image vertically to make sure it's aligned properly. If the sensor is an OV3660, some additional settings are adjusted to improve image brightness and saturation. The camera's frame size is set to QVGA for a faster frame rate, which helps with smoother streaming.

Once the camera is set up and the Wi-Fi connection is stable, the camera's IP address is displayed on the serial monitor. This IP address will be used to connect to the camera and start streaming. The startCameraServer() function is called to start the camera server, which would likely handle tasks like capturing images and performing face enrollment, although this function isn't fully defined in the code provided. Finally, the main loop has a short delay, waiting for a few seconds before any further actions are performed, but most of the important steps happen in the setup phase.

```
  esp_err_t err = esp_camera_init(&config);

  if (err != ESP_OK) {

    Serial.printf("Camera init failed with error 0x%x", err);

    return;

}
```

Figure 4.19: Camera Initialization

```
    startCameraServer();

        Serial.print("Camera Ready! Use 'http://");
        Serial.print(WiFi.localIP());
        Serial.println("' to connect");
    }
```

Figure 4.20: Starting camera and IP address snippet

4.5 Chapter Summary

In summary, this chapter focuses on the implementation of the I-Sensor Alert: IoT-Based Security for Intruder Alert system, describing the process in a detailed manner. It covers the installation and configuration steps that are crucial for the smooth and efficient operation of the system. The chapter explains how each component was set up and how they work together to provide an effective security solution. By following the steps outlined, the system functions as intended, offering reliable monitoring and alerting features to detect potential intruders.

CHAPTER 5: TESTING

5.1 Introduction

After the implementation of hardware and system, testing and running through the entire process will be conducted to ensure the functionality and the flow of the system. This is a crucial process where developer will find out errors and bugs throughout the process. In addition, the quality of the system will be ensured through testing before releasing the product or system to global users. The testing conducted will be mainly on functional testing.

5.1.1 Unit Testing

The testing will be conducted by breaking down the developed system into smaller parts as a unit, where detailed testing will be conducted for each part as listed below.

5.1.1.1 Hardware Testing

In hardware testing, we will break down the parts into several cases to ensure the quality and functionality of the hardware. The test cases that are being carried out for each module are as below:

- Internet Connectivity Module
- Camera Recognition Module
- Servo Motor Module
- Buzzer and Security Switch Module

Table 5.1: Test case for Internet Connectivity Module

Project Name:	I-Sensor Alert: IoT-Based Security for Intruder Alert System	Designed by:	Fatin Nazifa binti Shahrin
Module:	Internet Connectivity Module	Designed Date:	18 June 2025
Test Title:	WiFi Connection Stability	Pre-requisite:	-

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
1.	Verify that the system connects to the WiFi network.	• Power on the system and monitor the serial output. • Check if the system successfully connects to the specified Wi-Fi network.	• The system should successfully connect to the Wi-Fi network and print "Wi-Fi connected" in the serial monitor.	• System connects to WiFi successfully	Pass
2.	Verify WiFi stability during operation.	• Perform several system operations (face recognition, door control) while monitoring the Wi-Fi connection. • Check if the system maintains a stable connection without dropping out.	• The system should maintain a stable Wi-Fi connection throughout the operations.	• Wi-Fi connection remains stable during operations	Pass

Table 5.2: Test case for Camera Recognition Module

Project Name:	I-Sensor Alert: IoT-Based Security for Intruder Alert System	Designed by:	Fatin Nazifa binti Shahrin
Module:	Camera Recognition Module	Designed Date:	18 June 2025
Test Title:	Camera Initialization and Image Capture	Pre-requisite:	-

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
1.	Verify camera captures an image	• Trigger the face recognition function (either by presenting a face or using another input). • Ensure the camera captures the image and processes it without error.	• The camera should capture a clear image and process it correctly.	• The image is captured and processed successfully	Pass
2.	Verify camera resolution and image quality.	• Capture an image of a person from a specific distance • Inspect the image for clarity and ensure the face is visible and well-defined	• The image should be clear enough for face recognition, with no blurriness or distortion.	• Image quality is satisfactory	Pass

Table 5.3: Test case for Servo Motor Module

Project Name:	I-Sensor Alert: IoT-Based Security for Intruder Alert System		Designed by:	Fatin Nazifa binti Shahrin	
Module:	Servo Motor Module		Designed Date:	18 June 2025	
Test Title:	Servo Motor Movement and Control		Pre-requisite:	-	
No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
1.	Verify servo motor movement when the door opens.	• Trigger the door open function either through face recognition or a manual switch. • Monitor the servo motor to ensure it rotates to the correct angle (90 degrees for open).	• The servo motor should rotate from the closed position (0 degrees) to the open position (90 degrees).	• The servo rotates to the correct angle	Pass

2.	Verify servo motor resets to closed position after the door closes.	• After the door is opened for 5 seconds, ensure the servo motor returns to the closed position (0 degrees).	• The servo should return to the closed position after the predefined interval	• The servo returns to the closed position.	Pass

Table 5.4: Test case for Buzzer and Magnetic Switch Module

Project Name:	I-Sensor Alert: IoT-Based Security for Intruder Alert System		Designed by:	Fatin Nazifa binti Shahrin	
Module:	Buzzer and Magnetic Switch Module		Designed Date:	18 June 2025	
Test Title:	Buzzer and Security Switch Functionality		Pre-requisite:	-	
No.	**Test Case**	**Test Procedure**	**Expected Result**	**Actual Result**	**Pass/Fail**
1.	Verify buzzer sounds when security switch is activated.	• Activate the security switch (SW_SECURITY_PIN) by setting it high to simulate an intruder. • Monitor if the buzzer sounds as expected.	• The buzzer should sound when the security switch is triggered.	• The buzzer sounds.	Pass

2.	Verify buzzer stops when the security switch is deactivated.	• Deactivate the security switch (SW_SECURITY_PIN) by setting it low. • Ensure the buzzer turns off.	• The buzzer should stop when the security switch is turned off.	• The buzzer stops	Pass

5.1.1.2 Notification Testing

The test cases under notification testing are as below:

- Face Recognition Module
- Unauthorized Access Module

Table 5.5: Test case for Face Recognition Module

Project Name:	I-Sensor Alert: IoT-Based Security for Intruder Alert System	Designed by:	Fatin Nazifa binti Shahrin
Module:	Face Recognition Module	Designed Date:	18 June 2025
Test Title:	Send Notification for Successful Face Recognition	Pre-requisite:	-

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
1.	Verify that a notification is sent when a recognized user successfully enters.	• Ensure that an enrolled face (e.g., "ali") is presented in front of the camera. • The system should recognize the face and trigger the door opening mechanism. • Verify that the system sends a notification to Telegram with the correct username and timestamp.	• A notification with the username and timestamp should be sent to Telegram, indicating that the user has successfully entered.	• The notification includes the correct timestamp and user details	Pass

Table 5.6: Test Case for Unauthorized Access Module

Project Name:	I-Sensor Alert: IoT-Based Security for Intruder Alert System	Designed by:	Fatin Nazifa binti Shahrin
Module:	Unauthorized Access Module	Designed Date:	18 June 2025
Test Title:	Send Notification for Unauthorized Access Attempt	Pre-requisite:	-

No.	Test Case	Test Procedure	Expected Result	Actual Result	Pass/Fail
1.	Verify that a notification is sent when an unauthorized access attempt is detected.	• Present an unrecognized face (e.g., an unknown person) to the camera. • After 3 failed attempts, the system should send a notification indicating an unauthorized attempt and include a photo.	• A notification is sent to Telegram, indicating that an unauthorized person tried to enter, along with a photo of the person.	• The notification includes both the unauthorized access message and a photo	Pass

5.2 Chapter summary

Chapter 5 focuses on testing the IoT-based security system, I-Sensor Alert, to make sure it works properly before being released. After building the system, it is essential to test every part to find and fix any issues. The testing is divided into two main areas: Unit Testing and Notification Testing.

For Unit Testing, the system is split into smaller parts, and each part is tested separately. This includes checking the hardware, like ensuring the Wi-Fi connection is stable, the camera captures clear images, and the servo motor moves the door correctly. The buzzer and security switch are also tested to make sure they work as expected.

In Notification Testing, the system tests if notifications are sent when a user enters or when there is an unauthorized access attempt. For example, if the system recognizes a face, it should send a notification with the user's details. If someone tries to enter without being recognized, the system should send a notification about the failed attempt, including a photo of the person.

All tests in the chapter show that the system works well, confirming that the system is ready for use. This chapter highlights the importance of testing to ensure everything functions correctly before the system is launched.

CHAPTER 6: CONCLUSION AND FUTURE WORKS

6.1 Introduction

In this chapter, we will summarize the achievements and outcomes of the project, highlighting the objectives that were met and the challenges encountered along the way. Additionally, we will reflect on the limitations of the project and propose areas for future improvements and developments. The project focused on creating an advanced, modular IoT-based security system with real-time alerts, ensuring that users can easily monitor and control the system through mobile devices.

6.2 Objective Achievements

Table 6.1: Achievement of Objectives.

Objectives	Achievements
To develop a user-friendly interface that allows easy monitoring and control of the security system through mobile devices.	A mobile interface was created for easy monitoring and control of the security system, allowing access to features like face recognition and door control from anywhere.
To design a modular IoT security system that can be easily installed, enabling straightforward deployment without extensive modifications.	The system was designed to be easily installed and expanded, with simple integration of components like cameras and sensors.
To develop a real-time alert system for immediate and alert intruder detection to authorized personnel when an intruders entry occurs	A real-time alert system was implemented to notify authorized personnel instantly when an intruder is detected, using face recognition and security switches to trigger alarms and send Telegram alerts.

6.3 Project Limitations

Although the project successfully met its objectives, some specific limitations impacted its overall performance and functionality:

- Limited Face Recognition Database

The system currently supports a limited number of registered faces. With only a small number of users (5 enrolled faces), the system performs well. However, if the system were to be scaled to support more users, the face recognition performance could degrade due to limitations in processing power and memory, impacting both speed and accuracy.

- Dependence on WiFi for Real-time Notification

The real-time alert system relies heavily on an internet connection for sending notifications through Telegram. In cases of poor or unstable WiFi connections, delays in notifications could occur, which might compromise the system's ability to notify authorized personnel on time.

- Face Recognition Performance in Low Light

The face recognition module performed well under standard lighting conditions but struggled with poor lighting or extreme brightness. In low light, the camera’s ability to capture clear images of the face was hindered, which could reduce the system's effectiveness in real-world environments where lighting conditions are unpredictable.

6.4 Future Works

While the current project meets its objectives, several improvements and additions can be made to enhance the system:

- Improved Face Recognition

Future work should focus on improving the face recognition algorithm to handle different lighting conditions and angles better. Incorporating advanced machine learning techniques, such as deep learning, could significantly improve recognition accuracy.

- Offline Mode

Developing an offline mode for the security system that allows basic functionality without the need for Wi-Fi connectivity could help in areas with unreliable internet access. This could involve local storage for images and alerts that sync once the connection is restored.

- Integration with Other IoT Devices

Expanding the system to integrate with other IoT devices, such as smart locks, motion detectors, or even smoke alarms, would provide a more comprehensive security solution.

6.5 Conclusion

In conclusion, this project successfully developed a modular IoT-based security system that integrates face recognition and real-time alerts for enhanced security. The system was designed to be user-friendly, easily deployable, and capable of providing immediate alerts when unauthorized access is detected. While the project met its objectives, there are areas for future improvements, such as enhancing the accuracy of face recognition, providing offline functionality, and optimizing power consumption. With these future developments, the system could become even more reliable and scalable, offering an advanced and flexible solution for home and business security.

References

Andreas, N., Aldawira, C. R., Putra, H. W., Hanafiah, N., Surjarwo, S., & Wibisurya, A. (2019). Door Security System for Home Monitoring Based on ESP32. *Procedia Computer Science*, *157*, 673–682. https://doi.org/10.1016/j.procs.2019.08.218

Anitha, A. (2017). Home security system using internet of things. *IOP Conference Series Materials Science and Engineering*, *263*, 042026. https://doi.org/10.1088/1757-899x/263/4/042026

Fireteanu, V. (2020). Agile Methodology Advantages when delivering Internet of Things projects. *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. https://doi.org/10.1109/ecai50035.2020.9223172

Lovecek, T., Velas, A., Kampova, K., Maris, L., & Mozer, V. (2013). Cumulative probability of detecting an intruder by alarm systems. *IEEE*. https://doi.org/10.1109/ccst.2013.6922037

Maiti, C., E, V., & Muthuswamy, S. (2024). A Low Cost on-Device Intruder Detection System for Smart Home Environment. *IEEE*, 1–6. https://doi.org/10.1109/cict64037.2024.10899599

Moghavvemi, M., & Seng, N. L. C. (2004). Pyroelectric infrared sensor for intruder detection. *IEEE*. https://doi.org/10.1109/tencon.2004.1415018

Simatupang, J. W., & Tambunan, R. W. (2022). Security Door Lock Using Multi-Sensor System Based on RFID, Fingerprint, and Keypad. *IEEE*, 453–457. https://doi.org/10.1109/gecost55694.2022.10010367