DEVELOPMENT OF SMART ANTI-THEFT SYSTEM BASED ON IOT USING ESP32-CAM INTEGRATED WITH TELEGRAM BOT

AIMAN AKMAL BIN MOHD AINI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEVELOPMENT OF SMART ANTI-THEFT SYSTEM BASED ON IOT USING ESP32-CAM INTEGRATED WITH TELEGRAM BOT

AIMAN AKMAL BIN MOHD AINI

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronics Engineering Technology (Industrial Electronics) with Honours

Faculty of Electronics and Computer Engineering and Technology

Universiti Teknikal Malaysia Melaka

2025

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Development of Smart Anti-Theft System based on IoT using ESP32-CAM Integrated with Telegram Bot

Sesi Pengajian : 2024

Saya **AIMAN AKMAL BIN MOHD AINI** mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hak milik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (/)

☐	SULIT*	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia sebagaimana yang termaktub dalam AKTA RAHSIA RASMI 1972)
☐	TERHAD*	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
√	TIDAK TERHAD	

Disahkan oleh:

.... (TANDATANGAN PENULIS)
Alamat Tetap:

.... (COP DAN TANDATANGAN PENYELIA)
SITI HARYANTI BINTI HJ HAIROL ANUAR
Pensyarah
Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK)
Universiti Teknikal Malaysia Melaka (UTeM)

Tarikh: 25th January, 2025

Tarikh: 25th January, 2025

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this report entitled "Development of Smart Anti-Theft System based on IoT using ESP32-CAM Integrated with Telegram Bot" is the result of my own research except for quotes as cited in the references.

Signature : ....

Student Name : Aiman Akmal Bin Mohd Aini

Date : 25th January, 2025

APPROVAL

I declare that I have read this thesis and in my opinion, this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronics Engineering Technology (Industrial Electronics) with Honours.

Signature	:	....
Supervisor Name	:	Puan Siti Haryanti Binti Hairol Anuar
Date	:	25th January, 2025

Signature	:	....
Co-Supervisor Name (if any)	:	NOT APPLICABLE
Date	:	....

DEDICATION

This project is a heartfelt tribute to my cherished father, Mohd Aini Bin Harun, and my incredible mother, Mazuen Binti Abdul Ghani. Their unwavering support and steadfast belief in my potential have illuminated every step of my academic journey. The love they have showered upon me and the countless sacrifices they have made resonate deeply within my heart. They have not only shaped my character but have built the very foundation upon which every word written and every discovery made stands. I am eternally grateful to them for their boundless encouragement and for instilling in me the courage to pursue my dreams. To my beloved family, you are my guiding stars, and I owe you an immeasurable debt of gratitude that words alone cannot express.

ABSTRACT

The objective of this research is to design a smart anti-theft system utilizing IoT (Internet of Things) technology, specifically by combining the ESP32-CAM with a Telegram Bot. This system is created to address the limitations of traditional security systems by providing real-time monitoring and quick notifications. It aims to offer an affordable and scalable solution for detecting unauthorized access or movements. The system includes a user-friendly interface for setup and remote monitoring via Telegram. The process involves project planning, data collection, hardware and software setup, system deployment, testing, and report generation. By integrating ESP32-CAM and Telegram Bot, the system's capabilities are improved. Research findings show the successful deployment of the smart anti-theft system, enabling remote monitoring and immediate alerts for suspicious activities. This system signifies a significant improvement in security measures through IoT technology, effectively strengthening security and preventing theft incidents.

ABSTRAK

Objektif penyelidikan ini adalah untuk mereka bentuk sistem anti-kecurian pintar yang menggunakan teknologi IoT, khususnya dengan menggabungkan ESP32-CAM dengan Bot Telegram. Sistem ini dicipta untuk menangani batasan sistem keselamatan tradisional dengan menyediakan pemantauan masa nyata dan pemberitahuan cepat. Ia bertujuan untuk menawarkan penyelesaian yang berpatutan dan berskala untuk mengesan akses atau pergerakan yang tidak dibenarkan. Sistem ini merangkumi antara muka mesra pengguna untuk persediaan dan pemantauan jarak jauh melalui Telegram. Proses ini melibatkan perancangan projek, pengumpulan data, persediaan perkakasan dan perisian, penggunaan sistem, pengujian dan penjanaan laporan. Dengan mengintegrasikan ESP32-CAM dan Telegram Bot, keupayaan sistem dipertingkatkan. Hasil penyelidikan menunjukkan kejayaan penggunaan sistem anti-kecurian pintar, membolehkan pemantauan jarak jauh dan amaran segera untuk aktiviti yang mencurigakan. Sistem ini menandakan kemajuan yang ketara dalam langkah-langkah keselamatan melalui teknologi IoT, dengan berkesan mengukuhkan keselamatan dan mencegah kejadian kecurian.

ACKNOWLEDGEMENT

First and foremost, I want to sincerely thank Puan Siti Haryanti Binti Hj Hairol Anuar, my project supervisor, for all of his advise and great skills, which have been crucial in helping to make this project successful.

I also want to thank my family and friends for their unfailing encouragement and support. Their confidence in my ability gave me the will to achieve.

Further appreciation is owed to Universiti Teknikal Malaysia Melaka for supplying the facilities and tools required to see this project through to completion.

Last but not least, I want to thank me for believing in me, I want to thank me for doing all this hard work, I want to thank me for having no days off, I want to thank me for never quitting, I want to thank me for always being a giver and trying to give more than I receive. I want to thank me for trying to do more rights than wrong, I want to thank me for just being me at all times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IoT	-	Internet of Things
PIR	-	Passive infrared
ESP	-	Espressif Systems
API	-	Application Programming Interface
MEM	-	Micro-electromechanical systems
LCD	-	Liquid crystal display
MQ2	-	Message queue(gas sensor)
VoIP	-	Voice over Internet Protocol
CCTV	-	Closed circuit television
MIT	-	Massachusetts Institute of Technology
NodeMCU	-	Node Microcontroller Unit
WiFi	-	Wireless Fidelity

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This chapter describes the project background, problem statement, objectives, and scope of work in developing smart anti-theft system based on IoT using ESP32-CAM integrated with Telegram Bot.

1.1 Crime Rate in Melaka

According to Meor Riduwan Meor Ahmad[1], the crime rate in Melaka witnessed an increase of 2.44 percent surge between January and October 2023, as the number of reported cases reached 1,177, in contrast to 1,149 during the corresponding period in 2022 [1]. Additionally, for house break-ins and theft shows of 2.69 percent increasing in rate from 334 cases in 2022 to 343 cases in 2023 as depicted in Figure 1.1.

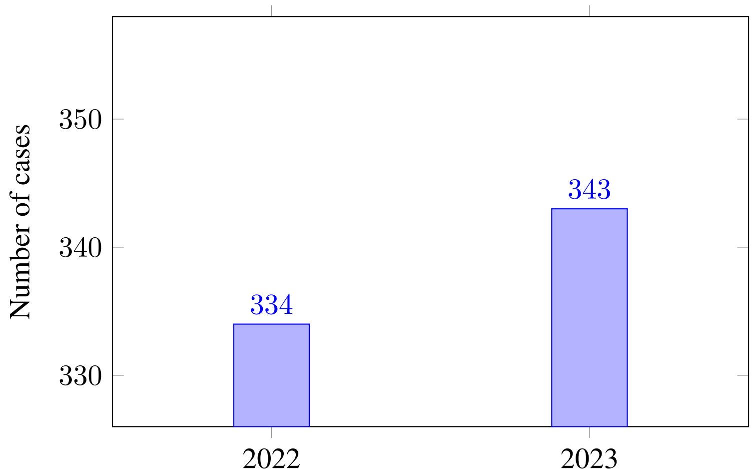

Figure 1.1: Comparison of Burglary Cases.

1.2 Problem Statement

The increasing rate of theft and burglary poses a significant threat to property and personal safety. Traditional security systems often lack real-time monitoring capabilities and fail to provide timely alerts to prevent theft incidents. There is a need for a more advanced and proactive approach to address this issue.

1.3 Project Objectives

The main aim of this project is to prevent and lowering the rate of theft. Specifically, the objectives are as follows:

1. To develop an IoT-based anti-theft detection system that can monitor and detect unauthorized entry or movement in a closed space like rooms.

2. To design a user-friendly interface for system configuration and remote monitoring via Telegram messaging platform.

3. To evaluate the effectiveness of the anti-theft system in providing timely notifications and preventing unauthorized access.

1.4 Scope of Work

The project scope includes the integration of ESP32 camera module, PIR sensor, and Telegram API to develop an effective anti-theft detection system. The ESP32 camera module will be used to capture images or videos upon detecting motion, while the PIR sensor will detect human presence within the monitored area. ESP32 camera also will facilitate data processing, control logic, and communication with other components and the Telegram messaging platform.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a broad overview of the project related to the topic in this report. Besides, the relevant literature is critically discussed and presented later in this chapter.

2.1 Introduction

The anti-theft detection system is an Internet of Things system intended to improve home and business security systems. An effective antitheft solution is created by this system, which integrates ESP32 camera module, and sensor technologies. The goal is to identify unlawful activity by strategically placing sensors in the target area like closed space such as motion detectors (PIR sensor). With their mobile devices, users may instantly access information, which speeds up their decision-making. In addition to allowing for remote monitoring, cloud-based architecture allows for data analysis and storage in order to provide even more security.

Reducing the danger of theft and ensuring the protection of precious assets is made possible by the suggested IoT-based anti-theft detection system, which is scalable and effective.

2.2 Past Related Project Research

2.2.1 Laptop Anti-Theft System with Tracking and Image Capture Device Based on Internet of Things Technology

According to research paper from R.Santika in 2023, the laptop is a versatile, portable tool that helps people with their job [2]. Thus, a laptop's storage contains a lot of crucial information and papers. But because the laptop cannot be traced, determining when it is taken is difficult. Even if the laptop may be found using default software, the program requires an internet connection in order to function. As a result, the owner can only locate their laptop while it is online. To increase security, a system that tracks the laptop's whereabouts and state is therefore required. This study presented an Internet of Things (IoT)-based system that can monitor and record the conditions surrounding the laptop. A camera, tracking device, warning system, and sensor for pressure are all part of the suggested setup. Figure 2.1 shows the flow of the system.

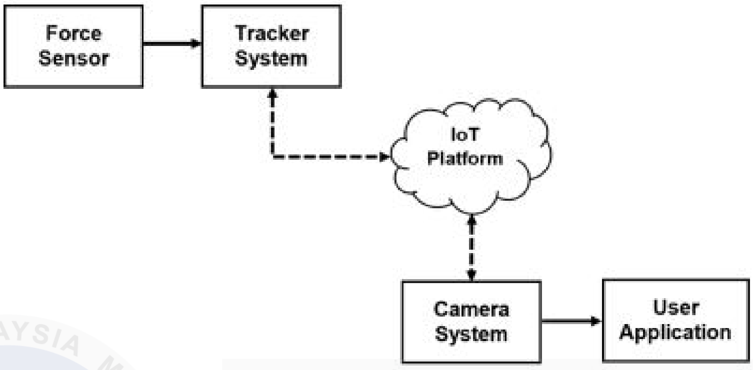

Figure 2.1: Block diagram of the system.

2.2.2 Smart Anti-Theft Water Metering System

With the use of cloud computing services, water flow sensor technology, and the Internet of Things (IoT), this study proposes an innovative method to the design and development of a particular context-based smart water meter system and deployment technique. The study from E.Effah in 2022 address two major issues which are water theft prevention and the issue of opaque water meter readings [3]. With the ability to identify leaks and water theft instantly, the suggested smart water meter system is an advancement in technology that can help with water consumption control. It also makes remote water meter reading and invoicing more transparent. Sent to the cloud server platform ThingSpeak, the water flow sensor provided real-time monitoring of water use statistics. Figure 2.2 shows how the system works.

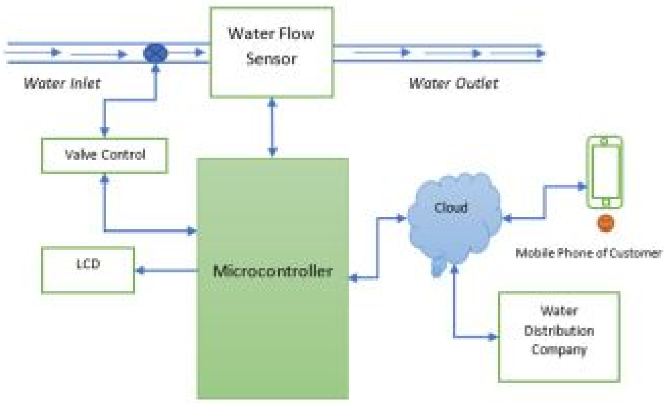

Figure 2.2: Block diagram of the system.

2.2.3 Smart Home Security System using IoT and ESP8266

From P.K.Sattaru in 2023, the paper discuses about home security is essential for safeguarding belongings and ensuring the safety of occupants, whether during workdays or vacations [4]. Manual monitoring is impractical and raises the risk of theft, burglary, and accidents involving elderly individuals or children. This research addresses the need for technological support in home security by introducing a smart home security system that leverages IoT and the ESP8266 microprocessor. The proposed system enhances security by allowing remote monitoring of unusual activities around the home at any time and from any location. Recent technological advancements have facilitated the development of advanced home security systems to address the increasing occurrences of home burglaries and fire accidents. This study presents a new model designed to meet these requirements, utilizing components such as surveillance cameras and sensors to create a comprehensive smart home security solution. In figure 2.3 shows how the system work.

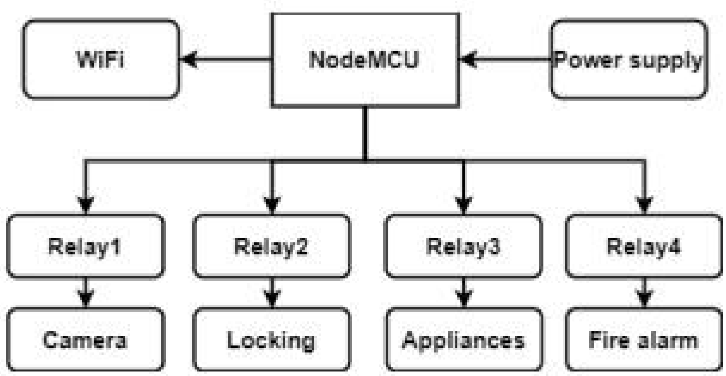

Figure 2.3: Block diagram of the system.

2.2.4 Low-Cost Compact Theft-Detection System using MPU-6050 and Blynk IoT Platform

This study describes a system that offers a small, smart surveillance system. According to A.Karnik in 2020, the Internet of Things, or IoT, has been a dominant force in several domains of applications in recent years [5]. People are becoming more connected to their devices as 5G technology emerges and devices become smarter. An MEM sensor that can identify angular disruptions is a gyroscope. The idea is to use a gas cutter or physical means to detect when the door is opened or knocked down. Using an ESP8266, the system is linked to the user's Wi-Fi network. This system may be installed on doors, stores, automobiles, and other surfaces because of its small form factor. The system has an alarm system that not only notifies the user via the Blynk mobile application but also alerts the neighbours. The suggested system is a transportable smart home theft detection system. Figure 2.4 illustrates the block diagram of the system.

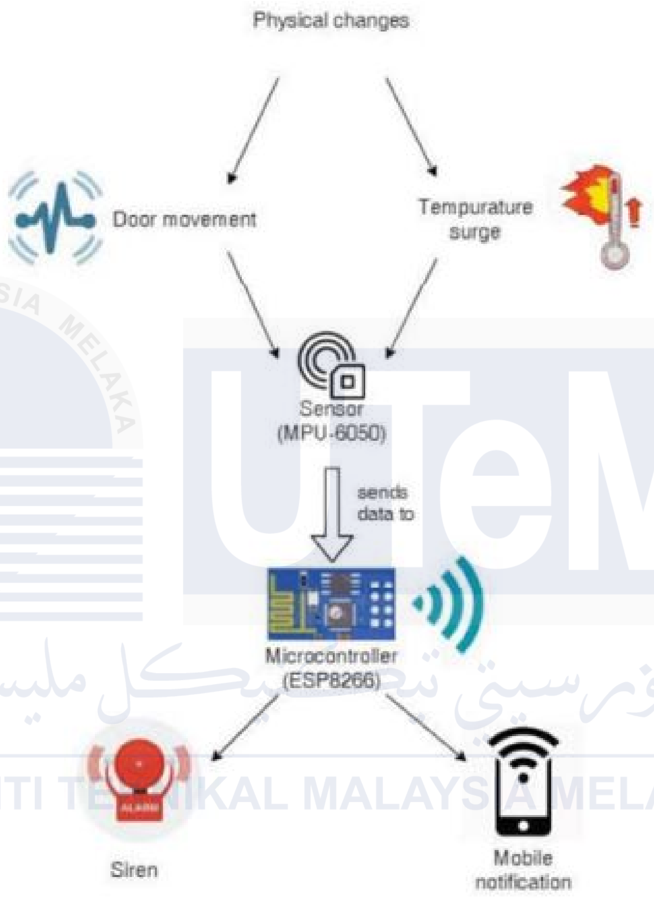

Figure 2.4: Block diagram of the system.

2.2.5 Home Security And Anti-Theft System

The purpose of a home security system is to protect your house and its occupants from potential threats such as gas leaks, broken pipes, and burglaries. As per discussed from T.Rani in 2022 , the purpose of this project is to develop a wireless, smart, internet of Things(IoT) home security system that can notify the owner via the Internet in the event of a fire, gas leak, or intrusion and sound an alarm [6]. The user may manage lighting and other smart gadgets through the system, which also notifies them in the event of a gas leak or fire. The security system's microcontroller will serve as an interface between its various parts, including the video doorbell, smart lights and devices, smoke and gas leakage sensor (MQ2), fingerprint scanner, LCD screen, buzzer, Reed Sensor, keypad and Wi-Fi module for online and app-based communication. Remote control of the doors, lighting, security cameras, and other smart devices is available to the user. Proper device authentication, identification, and encryption helps prevent data theft and privacy theft. The system works being showed in the figure 2.5.

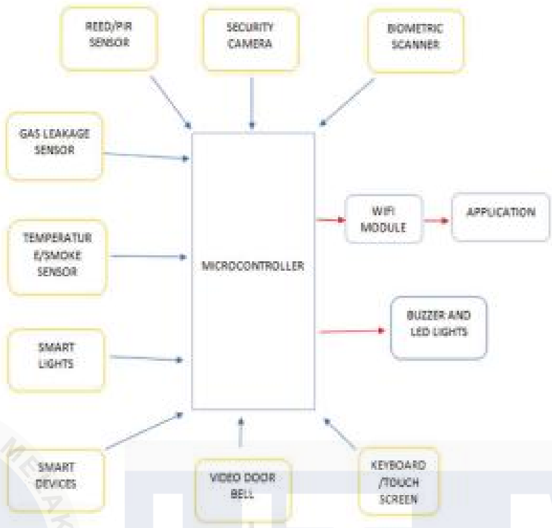

Figure 2.5: Block diagram of the system.

2.2.6 Smart Home Automation with Smart Security System over the Cloud

The article from G.Verma in 2023 discusses a smart home automation system utilizing ESP32 and ESP32-CAM modules integrated with IoT cloud technology [7]. This system incorporates sensors, actuators, and cameras for remote control, monitoring, and security in smart homes. It aims to enhance convenience, energy efficiency, and safety for homeowners. Technological advancements have facilitated the automation of appliances, with IoT technology enabling remote connectivity and operation via the Internet. The system is designed to be user-friendly, easy to install, and efficient in regulating household appliances. It emphasizes a cost-effective and reliable Internet of Things-based home control monitoring system. Figure 2.6 illustrates the block diagram of the system.

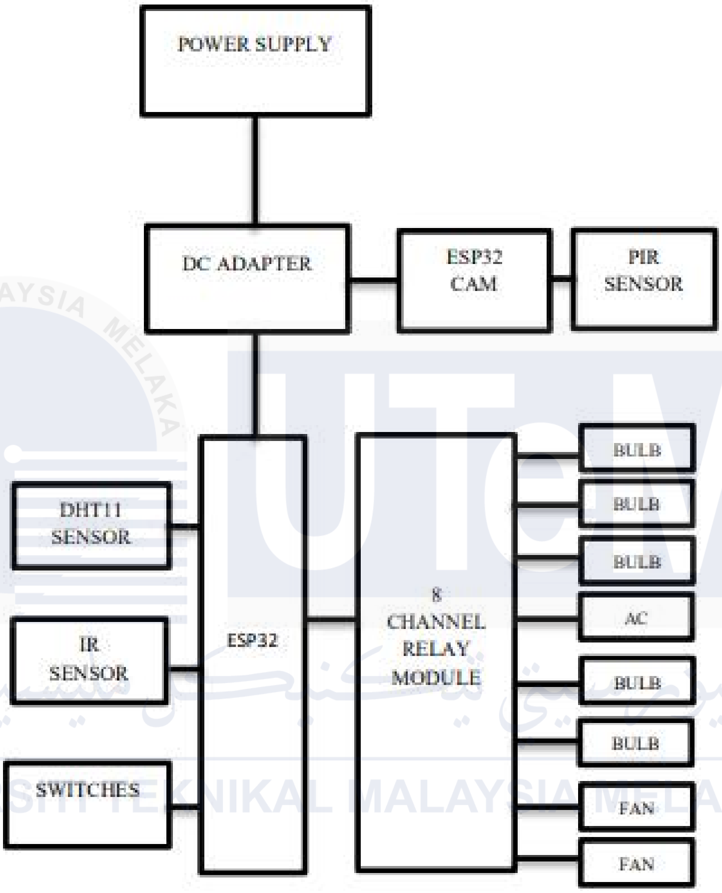

Figure 2.6: Block diagram of the system.

2.2.7 Independent, Integrated, Reconfigurable IOT based Home Security System

The Internet of Things (IoT) connects real-world devices through the Internet to enhance home automation, safety, and comfort. According to K.S.Keerthi in 2022, this project presents an IoT-based home security system featuring a door/window sensor, fire alarm, smart safe, and smart eye devices [8]. The door sensor monitors movement through gateways using fundamental physics. The fire alarm module alerts users to accidental kitchen fires with an emergency alarm and a VoIP call. The smart safe employs biometric authentication for secure access to valuables, while the smart eye, a motion-triggered camera module, offers user-controlled locking. All security-related data is stored in real-time on either an independent or dependent cloud, ensuring robust and flexible data management solutions. This integrated approach aims to provide a comprehensive, affordable, and efficient home security system leveraging modern IoT capabilities. Figure 2.7 shows how the system works.

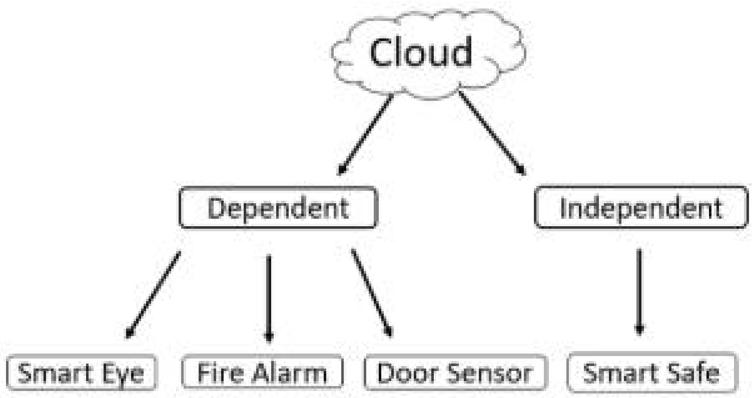

Figure 2.7: Block diagram of the system.

2.2.8 Smart Home Security with Dual Modes

Based on the report from M.N.M. Razali in 2020 , they focuses on designing and testing a CCTV system integrated with IoT technology to enhance home security [9]. The system includes a mobile application for user control and monitoring, as well as alarm notifications for neighbors. The study compares IP Cameras with IoT microcontrollers like ESP32 and Arduino, emphasizing the need for notification-based systems to improve home security. The hardware components include sensors, servos, microcontrollers, and cameras, while the software is developed using the MIT App Inventor. Future improvements may involve enhancing sensor efficiency and adding advanced features like firefighting technology. Figure 2.8 illustrates the system.

Figure 2.8: Block diagram of the system.

2.2.9 IoT-based Emergency Alert System Integrated with Telegram Bot

From M.I.M. Abu.Zaid in 2023,the research is about the IoT-based emergency alert system that operates by using gas and sound sensors to monitor the environment for unusual levels of gases or sounds [10]. When the sensors detect deviations beyond pre-set thresholds, they send signals to the Arduino microcontroller for processing. The Arduino then analyzes the sensor data and activates emergency alert devices, such as a siren and emergency light, through a relay. At the same time, the NodeMCU communicates with the Arduino to send real-time notifications to the user's smartphone using the Telegram app. This enables users to receive immediate alerts about potential emergencies, even when they are away from home. Users can interact with the system through the Telegram app, allowing for remote control to manage alerts and respond to emergencies promptly. By integrating IoT technology with Telegram, the system provides users with advanced monitoring capabilities, proactive alert notifications, and remote accessibility, enhancing home security and safety measures. Figure 4.9 shows overall work of the system.

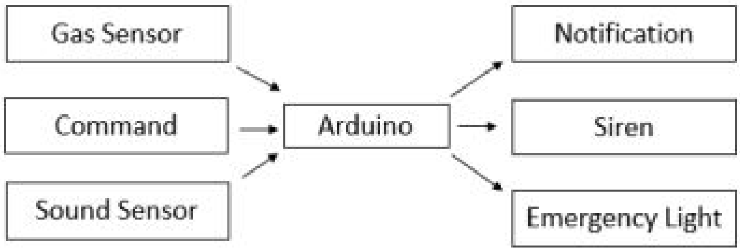

Figure 2.9: Block diagram of the system.

2.2.10 Chatting with Arduino Platform through Telegram Bot

According to J.C. de Oliveira in 2016, the integration between Telegram and the Arduino platform involves using Telegram Bots and the Telegram Bot API [11]. Telegram Bots are machine users capable of performing tasks similar to human users and can have artificial intelligence features. Figure 3.6 shows the working flow for the system. By integrating the Arduino platform with Telegram Bots, users can remotely communicate with their hardware prototypes through the Telegram application. This allows users to create hardware prototypes using sensors and motors and connect them to the Internet using shields like the Ethernet Shield or WiFi Shield. Through proper programming, users can interact with their hardware prototypes via Telegram Bots, enabling tasks like retrieving sensor data or controlling devices from anywhere in the world. This integration expands the possibilities of IoT applications in daily life by allowing users to create and communicate with their machines using the same tool they use to communicate with other human beings.

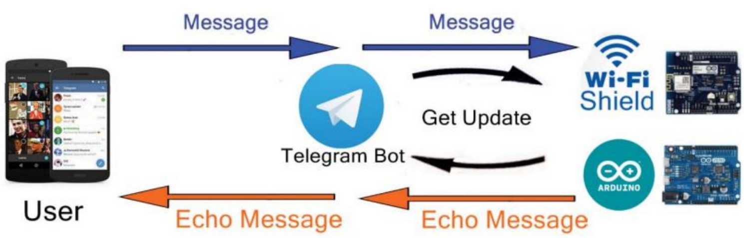

Figure 2.10: Block diagram of the system.

2.3 Known Techniques of Development of Smart Anti-Theft System based on IoT using ESP32-CAM Integrated with Telegram Bot

All the previous techniques as discussed above are summarized in Table 2.1.

Table 2.1: Summary of the previously proposed techniques

Authors (Year)	Proposed Technique	Advantage(s)	Disadvantage(s)
Karnik (2020) *et al.*, [5]	To develop a low-cost theft-detection system using the MPU-6050 gyroscope sensor and Blynk IoT platform. The system is mounted on the door to detect any physical disturbances like opening or cutting. When a disturbance is sensed, notifications are sent to the user and a siren alerts neighbors.	• Can have a compact design. • Can ultilize a low-cost project.	• Unable to function while out of range because of Internet connection. • Unable to have a stable motion detection system.
Effah (2022) *et al.*, [3]	To develop a smart water meter system usinf a wate flow sensor, IoT, and cloud services to enable remote meter reading, enhance billing transparency, and detect water theft.	• Provides real-time monitoring of water usage. • System can be configured to send automatic alerts to utilities or consumers.	• Unable to save cost because it is a bit expensive. • Requires ongoing maintenence, updates an technical supports.
Santika (2023) *et al.*, [2]	To develop a comprehensive securiity solution utilizing Internet of Things(IoT) technology to track and capture the condition around the laptop. The system integrates a sensor pressure, tracking device, camera, and warning system to enhance security measures.	• Enables real-time tracking of the laptop's location using GPS. • Anti-theft systems allow for remote locking of the laptop.	• Unable to save cost because it is a bit expensive. • Requires technical expertise, particularly in integrating and managing IoT devices and software.
Rani (2022) *et al.*, [6]	To develop a low-cost security system with remote monitoring capabilities. The proposed technique involves using an Arduino microcontroller connected to various sensors such as PIR sensor, MQ2(gas and smoke), and WiFi modules.	• Able to maximise a low-budget project. • Has the ability to sound an alarm or notify the owner via Internet.	• Relies on Internet connection. • Requires technical expertise, particularly in integrating and managing IoT devices and software.
Sattaru (2023) *et al.*, [4]	This system is designed to enhance home security and networking capabilities by monitoring and managing activites such as surveillance, automation, motion detection, and gas leaking.	• Uses biometrics, a password, and a special key to unlock the door. • Capable of setting off an alarm or sending an online notification to the owner.	• Uses Internet connection to operate. • Requires skilled technicians, especially in matters od controlling and integrating IoT software and devices.

Continued from previous page . . .

Authors (Year)	Proposed Technique	Advantage(s)	Disadvantage(s)
Verma (2023) *et al.*, [7]	The project involves developing a smart home automation system utilizing IoT cloud technology, ESP32, and ESP32-CAM modules to enable remote control, monitoring, and security functions within a smart home environment.	• Cost effective by using ESP32 and ESP32-CAM modules. • Can optimize energy usage by automating the operation of appliances.	• Requires Internet connection. • Requires ongoing maintenence, updates an technical supports.
Keerthi (2022) *et al.*, [8]	The IoT-based home security system integrates devices such as a door/window sensor, fire alarm, smart safe, and smart camera to provide comprehensive home protection. All security data is stored in real time on a cloud.	• Can be access anywhere by using smartphone or any devices like it. • Have real-time monitoring.	• Requires a big cost to implement a system with multiple devices. • Requires skilled technicians, especially in matters od controlling and integrating IoT software and devices.
Razali (2020) *et al.*, [9]	To develop an advanced security system using IoT technology, sensors, microcontrollers, cameras, and a mobile app. The system has a Calibration Mode for equipment testing and a Security Mode for live monitoring.	• Have dual modes. • Have potential for future enhancement.	• Requires a big cost to implement a comprehensive system. • Requires skilled technicians, especially in matters of controlling and integrating IoT software and devices.
Abu.Zaid (2023) *et al.*, [10]	The IoT Emergency Alert System utilizes sensors to detect gas and sound levels. When triggered, it sends notifications to the user's smartphone via Telegram for improved home security.	• Have integration with Telegram Bot enables users to interact with the system. • Provides real-time alerts that can be accessed everywhere.	• Requires internet connection to use it. • Requires skilled technicians, especially in matters of calibrating the sensors, Arduino and the NodeMCU.
Oliveira (2016) *et al.*, [11]	The project seamlessly links Telegram with Arduino through the use of Telegram Bots, granting users the ability to remotely control hardware prototypes.	• Have security features of Telegram, which is data encryption. • Have wide IoT concepts can be inspired by this project.	• Dependency on Internet connection. • Requires expertise in developing the system.

2.4 Summary

The second chapter of the report provides a comprehensive literature review on smart anti-theft systems and IoT technology. It begins with introducing the topic, emphasizing the importance of anti-theft detection systems powered by Arduino-based Internet of Things solutions for enhancing home and business security. The chapter discusses various related projects. These include a laptop anti-theft system with tracking and image capture device based on IoT, a smart anti-theft water metering system, a smart home security system using IoT and ESP8266, a low-cost compact theft-detection system using MPU-6050 and the Blynk IoT platform, a home security and anti-theft system, smart home automation with smart security system over the cloud, an independent, integrated, reconfigurable IoT-based home security system, and smart home security with dual modes. These projects showcase the diverse applications and technologies in the field, providing essential knowledge and insights for the development of the proposed smart anti-theft system in the current project.

CHAPTER 3

METHODOLOGY

This chapter describes the methodology and design steps to develop smart anti-theft system based on IoT using ESP32-CAM integrated with Telegram Bot and determining whether the project is processing smoothly in this area. This section provides a short explanation of the steps that need to be taken, in addition to an explanation of the primary components.

3.1 Description of Methodology - Sustainable Development

Our main goal is to ensure that all project requirements are met will be covered by the techniques and approches used with a brief overview of the actions that need to be followed. Figure 3.1 illustrates the top level of the system. ESP32-CAM act as the main brain of this system, controlling and commanding PIR sensor. To design the proposed system, depicted in Figure 3.7 the system commences with user inputs. When unauthorized movements or suspicious activities are detected within the monitored area by PIR sensor, ESP32-CAM will captures the image and videos, providing visual evidence of security breaches. Therefore, the captured media, along with relevant information, is sent to a predefined Telegram account for remote monitoring and alerting purposes.

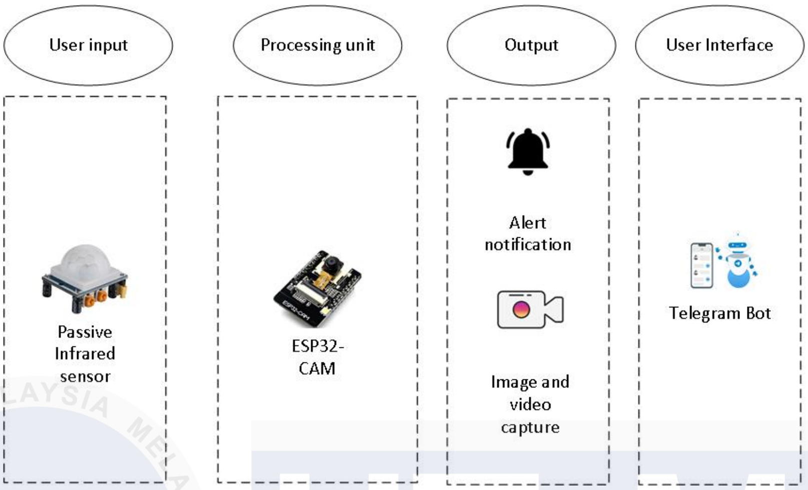

Figure 3.1: Top level block diagram of smart anti-theft system.

3.2 Hardware

3.2.1 ESP32-CAM

Figure 3.2 is a powerful tool for developing Internet of Things (IoT) applications, thanks to its combination of a camera and Wi-Fi capabilities. The ESP32-CAM module has an integrated ESP32-S system-on-chip, which combines Wi-Fi, Bluetooth, and a dual-core processor. It is equipped with a camera sensor that can capture images up to 2 megapixels in resolution. This module is perfect for various uses like surveillance, face recognition, and image processing.

In comparison, ESP8266 is missing some features and does not have a camera interface, so the ESP32-CAM is a better option for projects needing image capture. The ESP32 also has a faster processor, more GPIO pins, and Bluetooth connectivity. Refer to Table 3.1 for ESP32-CAM specifications.

Figure 3.2: Module of ESP32-CAM.

Table 3.1: Specification of ESP32-CAM

No.	Specification	Detail
1.	Processor	ESP32-DOWD
2.	RAM	Internal 512kB + External 4mB PSRAM
3.	WiFi protocol	ESP32-DOWD
4.	WiFi mode	WPA/WPA2/WPA-2 Enterprise/WPS
5.	Bluetooth	Bluetooth v4.2 BR/EDR and BLE
6.	Power Supply	5V
7.	GPIO pins	10(available)
8.	Dimensions	40.5mm x 27mm x 4.5mm

3.2.2 Passive Infrared Sensor

In Figure 3.3, there is a Passive Infrared (PIR) sensor. This sensor can detect infrared (IR) light that comes from objects in its field of view. It works without emitting any energy for detection. PIR sensors are often used in security alarms and automatic lighting systems to detect general movements, like that of people or animals. However, they cannot give detailed information about the moving object. The sensor detects changes in the amount of IR radiation, which varies with the temperature and surface characteristics of the objects in front of it. When an object moves across the sensor's field of view, it triggers a change in the output voltage, activating the detection mechanism. These sensors are commonly used in applications like motion-activated lights, burglar alarms, and occupancy detection systems.

Figure 3.3: Module of PIR sensor.

3.2.3 AA Alkaline Battery

Figure 3.4 shows AA alkaline batteries that serve as the main power source for the system. The selection is based on factors such as widespread availability, cost-effectiveness, and straightforward replacement. In contrast to rechargeable batteries, AA alkaline batteries deliver stable voltage and eliminate the need for a charging infrastructure, thereby streamlining both the design and upkeep of the system. These batteries are especially appropriate for applications characterized by low to moderate power demands, exemplified by the smart anti-theft system. In this configuration, a set of four AA batteries delivers a nominal voltage of 6V, making it suitable for the ESP32-CAM and its related components. This arrangement guarantees consistent performance without the necessity for regular recharging or specialized charging devices.

Figure 3.4: Module of AA alkaline battery.

Table 3.2: List of hardwares.

No.	Component	Price	Quantity	Total
1.	ESP32-CAM	RM26.90	1	RM26.90
2.	PIR sensor	RM5.90	1	RM5.90
3.	AA Alkaline battery	RM8.90	1	RM8.90
4.	Total Spent			RM34.90

3.3 Software

3.3.1 Arduino IDE

In this project, the code will be upload to the ESP32-CAM using the Arduino Software (IDE). This software is a powerful tool for programming and uploading code to Arduino boards. First, the IDE allows you to write programs in the text editor. These programs are saved with the .ino file extension. With the Arduino programming language, users can create personalized features and manage sensors, motors, and other components. After completing the code, it can be uploaded to the Arduino board. The IDE communicates with the board through USB, allowing consistent program transfer to the hardware. Additionally, the IDE offers debugging tools for error identification. It can display error messages, monitor output through the serial monitor, and troubleshoot issues. Lastly, the Arduino IDE can be used offline, making it suitable for individuals with unreliable or no internet access.

Figure 3.5: Arduino IDE software.

3.3.2 Telegram Bot

The Telegram Bot is an important aspect in the smart anti-theft system based on IoT using ESP32-CAM. It acts as a bridge between the hardware components and the user interface, enabling seamless communication. Unlike regular Telegram accounts, the Telegram Bot is operated by software rather than a human user. Its main function is to facilitate message transmission, command reception, and user interaction in a conversational manner. In this system, the Telegram Bot is programmed to receive alerts and notifications from the hardware components, particularly the ESP32-CAM, when security breaches are detected. Whenever unauthorized movements or suspicious activities occur, the Telegram Bot promptly sends real-time alert messages to the user's Telegram account.

The ESP32-CAM module is capable of sending alerts in the form of text notifications, images, and videos, providing visual evidence of security breaches. In addition, the Telegram

Bot enables users to remotely interact with the system. Users can request access to live camera feeds for real-time monitoring. This two-way communication feature enhances user engagement and control over the anti-theft system, empowering them to promptly respond to security threats. In summary, the Telegram Bot serves as a user-friendly interface that facilitates seamless communication, alert dissemination, and remote monitoring in the smart anti-theft system.

Figure 3.6: Telegram Bot as in Telegram software.

3.4 Flow Chart

Based on the figure 3.7, the flowchart outlines the operational process of a smart anti-theft system using the ESP32-CAM, which integrates motion detection, image capture, and Telegram notifications. The system begins with the initialization of the ESP32-CAM.

First, it verifies whether the Telegram bot feature is active. If it is, the system checks if the PIR (Passive Infrared) sensor is powered on. If the PIR sensor is active, the security feature is enabled, and the system starts monitoring for motion. When motion is detected, a notification is sent to the user via Telegram, and a picture is captured and transmitted to the user for verification.

The system also takes into account the state of the flash feature. If the flash is enabled, it activates during picture capture to improve image quality in low-light conditions. If the flash is disabled, the system directly checks the PIR sensor for motion detection and follows the same notification and image capture process.

This process is cyclical, meaning the system continuously monitors the PIR sensor and processes alerts until it is turned off or reaches an endpoint. This design ensures ongoing surveillance and efficient communication with the user through Telegram notifications.

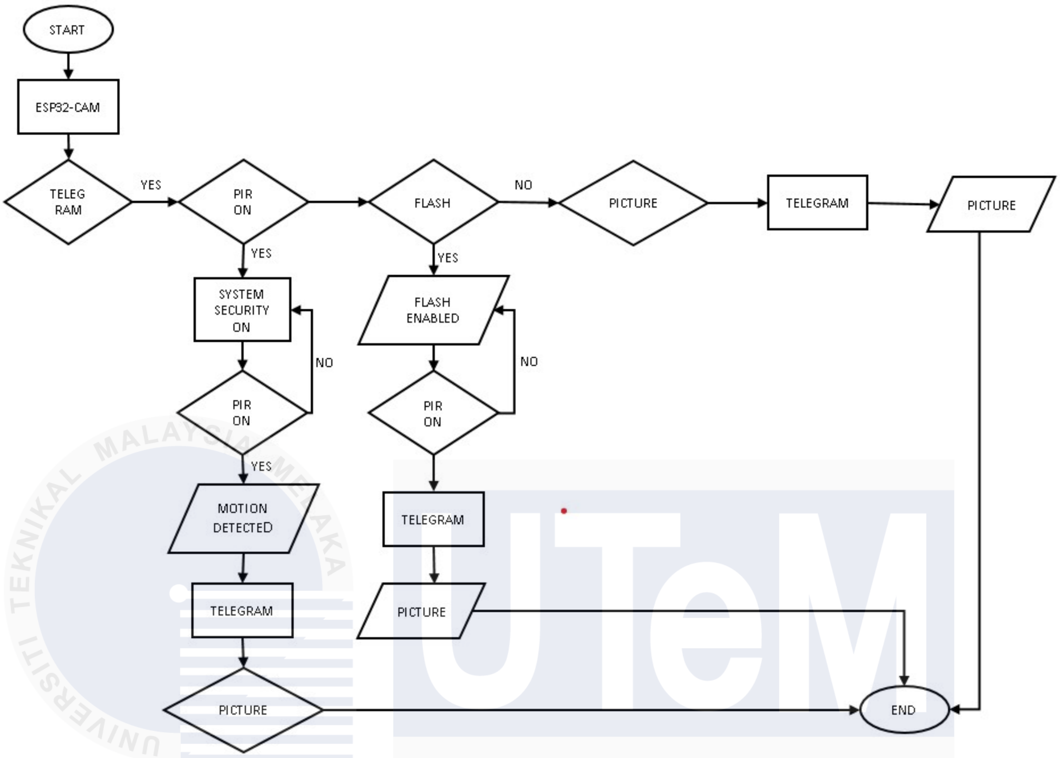

Figure 3.7: Flow chart of the system.

3.5 Gantt Chart

In Projek Sarjana Muda I (PSM I), the focus was on the literature review, the definition of the objective, the definition of the problem statement, the definition of the scope of work, the definition of methodology, and preliminary results. Figure 3.8 depicts the timeline of project implementation in PSM I. The literature review mainly focuses on ESP32-CAM, PIR sensors, and previous techniques on smart anti-theft system.

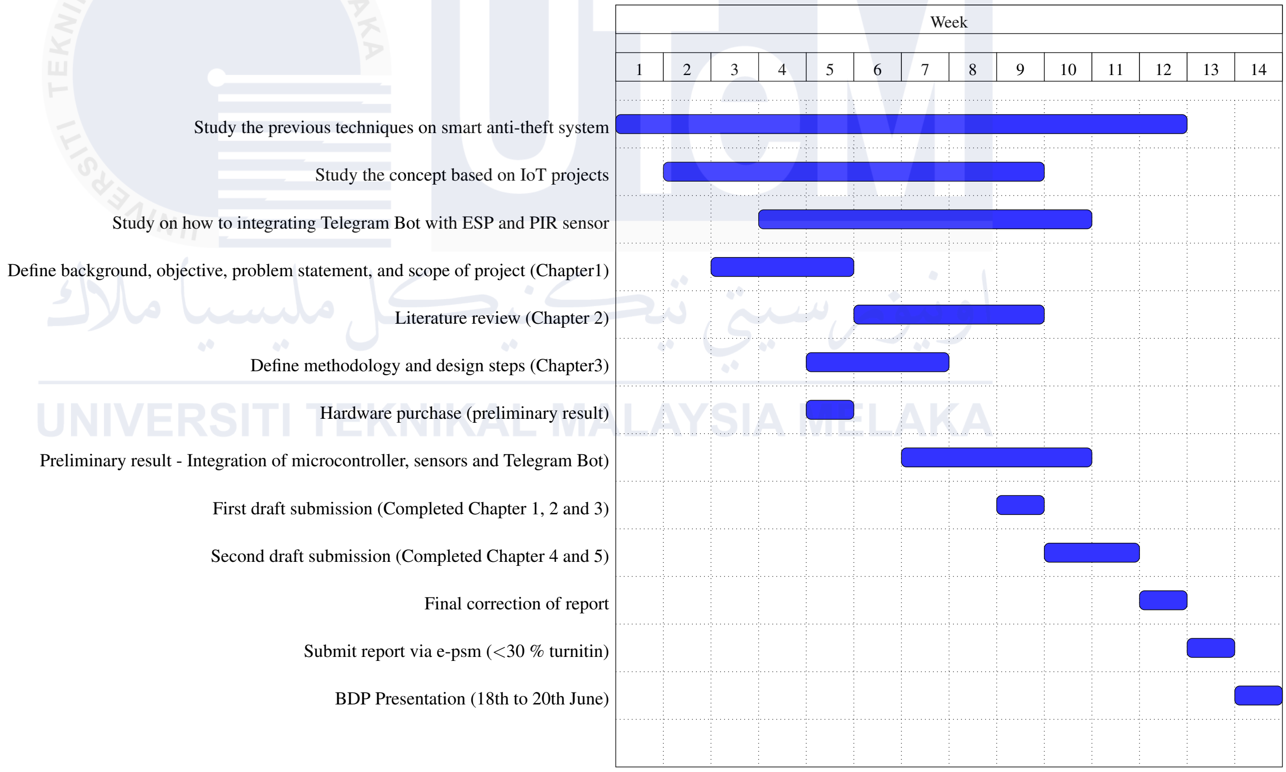

Figure 3.8: Timeline for PSM I

In PSM II, the focus is on hardware integration, development of the , testing and validation of the proposed system. Figure 3.9 depicts the time-line for project implementation in PSM II.

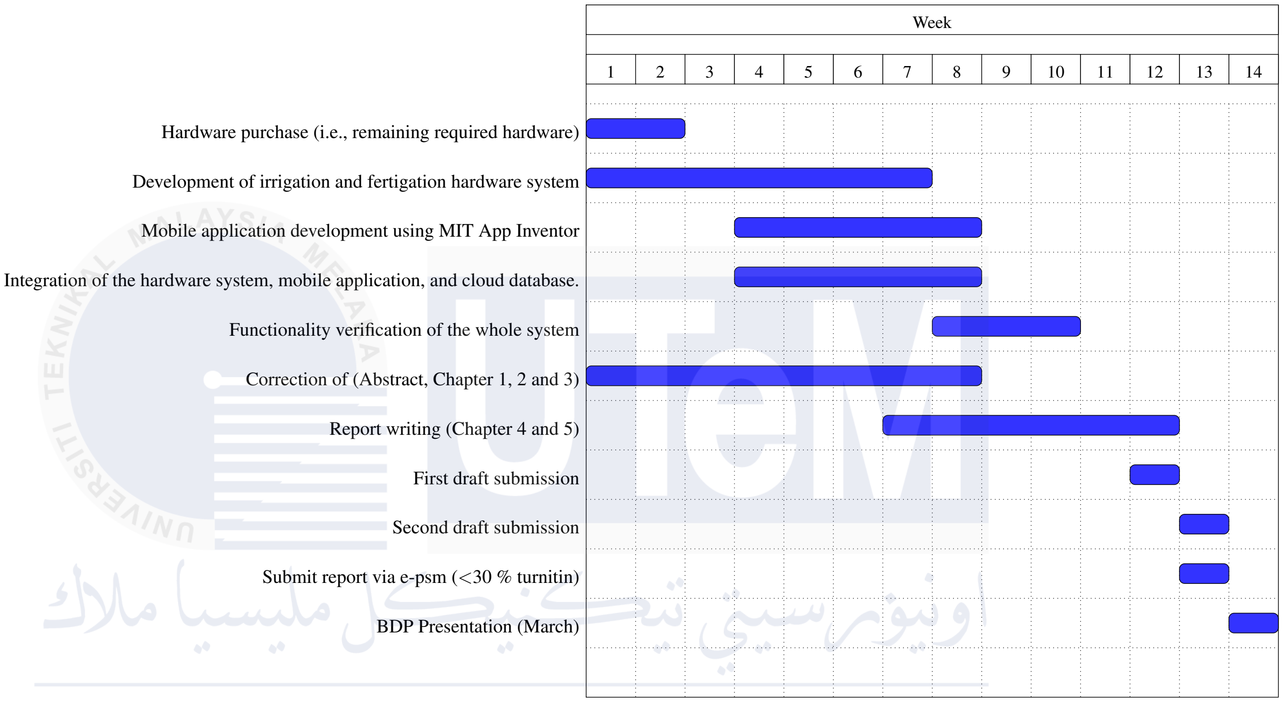

Figure 3.9: Timeline for PSM II

3.6 Summary

In this chapter, a comprehensive overview is presented regarding the methodologies, hardware, and software components utilized in the creation of the intelligent anti-theft system. The methodology is specifically designed to ensure a thorough and organized approach to creating a highly effective security solution. By using the ESP32-CAM and PIR sensor, the system has strong detection capabilities, and the integration with the Telegram Bot allows for immediate user alerts and remote monitoring. The project is divided into two phases to facilitate comprehensive development and testing, ensuring that the end product meets all objectives. This meticulous approach provides the system's efficiency in real-time monitoring and alerts, greatly enhancing security measures in closed space such as rooms.

CHAPTER 4

RESULTS AND DISCUSSION

This chapter presents the results and analysis on the development of smart anti-theft system.

4.1 Design of Smart Anti-Theft System in Telegram

As mentioned in Section 3.3.2, Telegram is used to design the interface for the smart anti-theft system. Figure 4.1 depicts the user interface of developed smart anti-theft system which prioritizes simplicity and functionality. The arrangement of the interactional function makes it simple to navigate and quickly access key functionality. User may take a photo by pressing the ”Photo” command. User also can armed the system to turn on the PIR sensor by simply pressing the ”Arm” command. Moreover, the interface also have the function to control the flash of the ESP32-CAM and know the state of the system whether the system have turn the PIR sensor on or off.

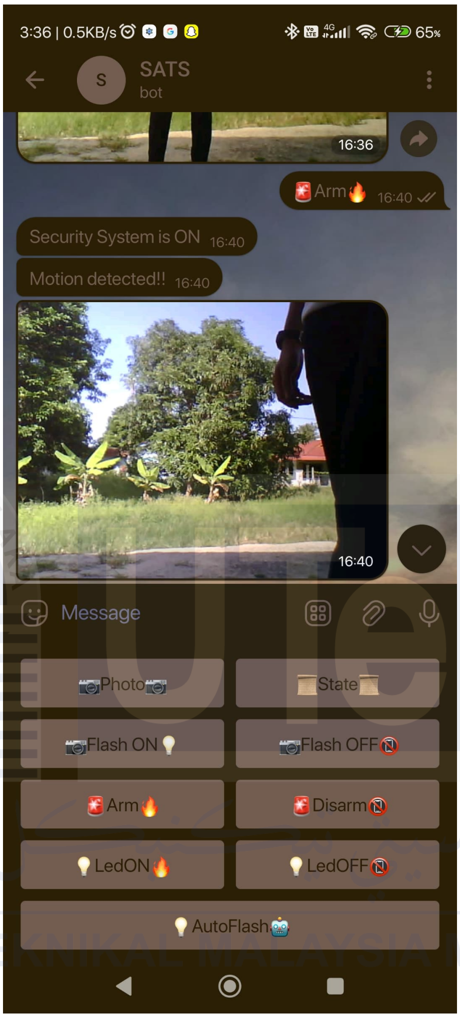

Figure 4.1: Main interface of the system in Telegram.

Figure 4.2 depicts the base coding for commands or states of the smart anti-theft system to control in Telegram. Each string is linked to an emoji and accompanying text to improve user interaction and readability. For example, commands like "Photo" trigger a photo capture, while ("Arm") and "Disarm" respectively activate and deactivate the system. Similarly, "LedON" and "LedOFF" control the LED light, while "AutoFlash", "Flash ON", and "Flash OFF" manage the camera flash settings.

```
const String photo_str = "📷Photo📷";
const String state_str = "📜State📜";
const String arm_str = "🚨Arm🔥";
const String disarm_str = "🚨Disarm📵";
const String ledon_str = "💡LedON🔥";
const String ledoff_str = "💡LedOFF📵";
const String autoflash_str = "💡AutoFlash🤖";

const String flashOn_str = "📷Flash ON💡";
const String flashOff_str = "📷Flash OFF📵";
```

Figure 4.2: Strings to represent specific commands for the smart anti-theft system.

4.2 Security Measures

To ensure strong security, the system has multiple layers of protection designed to safeguard both user data and device functionality.

1. Authentication tokens

    - Authentication tokens serve as the first line of defense, ensuring that only authorized users can interact with the system. These unique identifiers verify each user's identity, preventing unauthorized access and malicious commands. Without a valid token, the system will not process any requests, maintaining a secure operational environment. Figure **??** shows the coding for authentication tokens

```
// Replace with your network credentials
const char* ssid = "RemiNote";
const char* password = "1sampai8";

// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you
String chatId = "5057903264";

// Initialize Telegram BOT
String BOTtoken = "8000092277:AAGNtm6s1M9glL0XAzJZKuNTKUsdhrLTOZU";
```

Figure 4.3: Coding for authentication tokens.

2. Encryption

   - Encryption is employed to secure all communications between the system and the Telegram bot. By using advanced cryptographic protocols, the system ensures that sensitive data, such as captured images and alerts, cannot be intercepted or accessed by third parties. This encryption layer is crucial for maintaining the privacy and integrity of the transmitted information.

4.3 Sending Photo to Telegram

To send the photo to Telegram, figure 4.4 shows the main loop function in Arduino sketch, designed to handle essential system functionalities such as sending photos, detecting motion, and responding to messages via a Telegram bot.

The first section checks if the ‘sendPhoto‘ flag is set, which indicates a request to take a photo. When ‘sendPhoto‘ is true, the system prepares to send the photo by calling the ‘sendPhotoTelegram()‘ function and then resets the ‘sendPhoto‘ flag to prevent repeated execution. This ensures that photo capture and transmission occur only when specifically requested.

The second section deals with motion detection. If the system is armed (indicated by ‘armed‘ being true) and motion is detected (when ‘motionDetected‘ is true), it sends a ”Motion detected!!” message via the Telegram bot using ‘bot.sendMessage()‘. Additionally, it logs the detection to the serial monitor and triggers the ‘sendPhotoTelegram()‘ function to send a photo of the detected motion.

The final section manages periodic updates from the Telegram bot. It checks if enough time has passed since the last bot operation using ‘millis()‘ and ‘botRequestDelay‘. If sufficient time has elapsed, it retrieves new messages from the bot with ‘bot.getUpdates()‘ and processes them in a loop using ‘handleNewMessages()‘. The ‘lastTimeBotRan‘ variable is updated to ensure proper timing for future requests, maintaining efficient communication with the Telegram bot. This combination of features makes the system responsive and user-interactive.

```
void loop() {
  if (sendPhoto) {
    Serial.println("Preparing photo");
    sendPhotoTelegram();
    sendPhoto = false;
  }

  if (armed && motionDetected) {
    bot.sendMessage(chatId, "Motion detected!!", "");
    Serial.println("Motion Detected");
    sendPhotoTelegram();
    motionDetected = false;
  }

  if (millis() > lastTimeBotRan + botRequestDelay) {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
      Serial.println("got response");
      handleNewMessages(numNewMessages);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
  }

}
```

Figure 4.4: Arduino sketch for handle sending photos, detecting motion and responding to messages via Telegram bot.

4.4 Expected Result

The section will cover integrating the ESP32-CAM and PIR sensor to improve intrusion detection accuracy. It will also discuss the ESP32-CAM's image capturing capabilities, including resolution and quality adjustments.

Additionally, the section will explore setting up a Telegram Bot for instant notifications of security breaches. Instructions on creating an API key and configuring parameters will be provided for real-time alerts with captured images and timestamps. Table 4.1 shows the expected result.

Table 4.1: Expected result.

Component and Software	Input/Output	Action
PIR sensor	Triggered by suspicious movement	Signalling ESP32-CAM to take picture or record the video.
ESP32-CAM	Triggered by PIR sensor	Take picture or record video.
Telegram	By user	Real-time monitoring and immediate notification.

4.5 Integration of Hardware and Telegram

The mobile application interfaces seamlessly with hardware components using the capabilities of the ESP32-CAM module as a web server, which utilizes a Wi-Fi network for real-time communication and data synchronization. This system aims to provide an advanced anti-theft solution, ensuring efficient monitoring and quick notifications.

This prototype in figure 4.5a incorporates key hardware elements, including the ESP32-CAM, a PIR motion sensor, and AA alkaline batteries. The ESP32-CAM serves as the central controller, capable of capturing high-resolution images and transmitting them via a Telegram bot for instant alerts. The system architecture allows users to monitor and control operations remotely through a user-friendly Telegram interface. Commands such as activating motion detection, capturing images, and toggling the flash can be executed efficiently.

The integration of the ESP32-CAM's image-capturing capability with real-time notifications through Telegram creates a robust security solution. The PIR sensor complements this setup by detecting movement and triggering alerts, making the system suitable for home and business security applications. This solution balances cost-effectiveness and scalability while maintaining high accuracy and efficiency in intrusion detection.

(a) The prototype of the smart anti-theft system.

(b) The user interface in Telegram.

Figure 4.5: The integration of the hardware and Telegram.

4.6 Data Analysis

This report evaluates the effectiveness of a smart anti-theft system using IoT technology, specifically the ESP32-CAM, PIR motion detector and a Telegram Bot. It examines key aspects such as motion detection accuracy, image quality, notification reliability and power consumption analysis. The analysis highlights the system's strengths, identifies potential issues, and provides recommendations for improvement, serving as a foundation for assessing its practical performance.

4.6.1 Data Description

The analysis presented in this chapter is organized into five principal performance metrics, each illustrating the operational capabilities and efficiency of the smart anti-theft system.

1. Motion detection accuracy
   - Evaluates the accuracy of the PIR sensor in detecting movement in various environmental conditions
2. Image quality
   - Evaluates the sharpness and definition of images obtained from the ESP32-CAM under various lighting scenarios.
   - Metrics encompass clarity scores, which are evaluated on a scale ranging from 1 to 10, as well as noise levels, represented as a percentage.
3. Notification delivery
   - Evaluates the time taken from the start of a trigger event to the subsequent delivery of a notification via the Telegram Bot.
   - Data is captured through various types of networks, including Wi-Fi and mobile data.
4. Power Consumption Analysis
   - Estimation of battery life based on current draw and battery capacity.

- Assess the current and voltage usage of the ESP32-CAM and PIR sensor in various states, including idle, active, and image transmission.

4.6.2 Data Result

This section presents the data result of the smart anti-theft system.

4.6.2.1 Motion detection accuracy

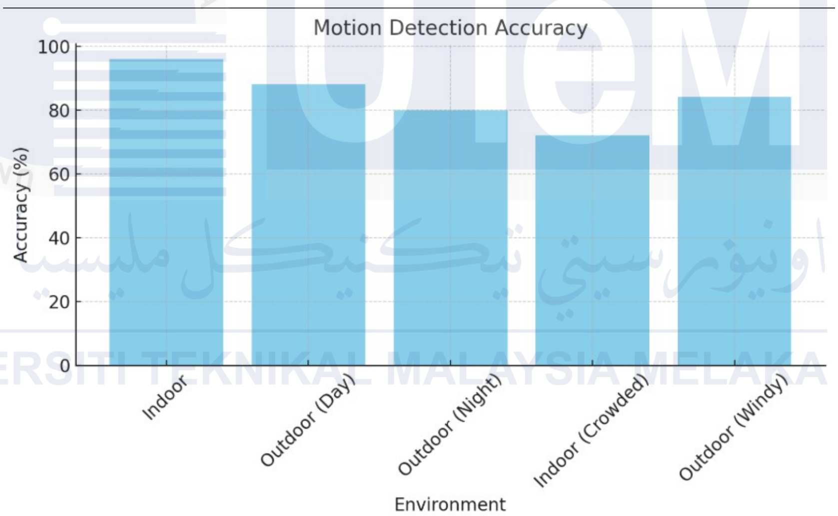

Figure 4.6: Figure shows motion detection accuracy chart.

The motion detection system works best in controlled indoor environments, achieving an impressive accuracy of 96%. This high accuracy is due to minimal external factors like temperature changes and movement. However, in crowded indoor spaces, accuracy drops to about 72% because overlapping heat signatures and rapid movements can confuse the sensor. Outdoors at night, accuracy further declines to around 80%, influenced by reduced visibility and fluctuating temperatures.

These variations highlight how sensitive the PIR sensor is to environmental changes, especially with unpredictable movement or heat sources. While it performs well in stable conditions, its limitations in challenging environments suggest a need for enhancements, such as better motion filtering or additional sensors, to maintain reliable detection across different settings.

4.6.2.2 Image quality

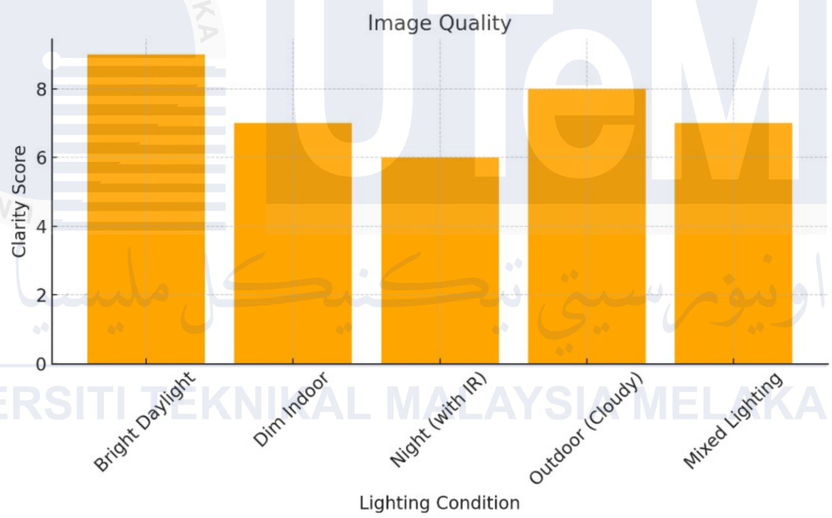

Figure 4.7: Figure shows image quality chart.

The ESP32-CAM produces fantastic image quality in bright lighting, scoring a solid 9 for clarity. This means that the pictures it takes are sharp and detailed, making it perfect for both indoor and outdoor settings when there's plenty of light. However, when the lighting conditions aren't as bright—like in dim or mixed lighting—the clarity takes a bit of a hit, with scores dropping to 7. This drop likely happens because the camera struggles to balance exposure and manage shadows.

When it comes to nighttime, the clarity score drops even further to 6, mainly due to heightened noise levels in low-light situations. While the ESP32-CAM can still snap decent photos at night, it really needs some ambient light to work well. To enhance its performance in these darker settings, adding features like infrared lighting or noise reduction software could really make a difference, helping to produce clearer images regardless of the lighting conditions.

4.6.2.3 Notification Delivery Delay

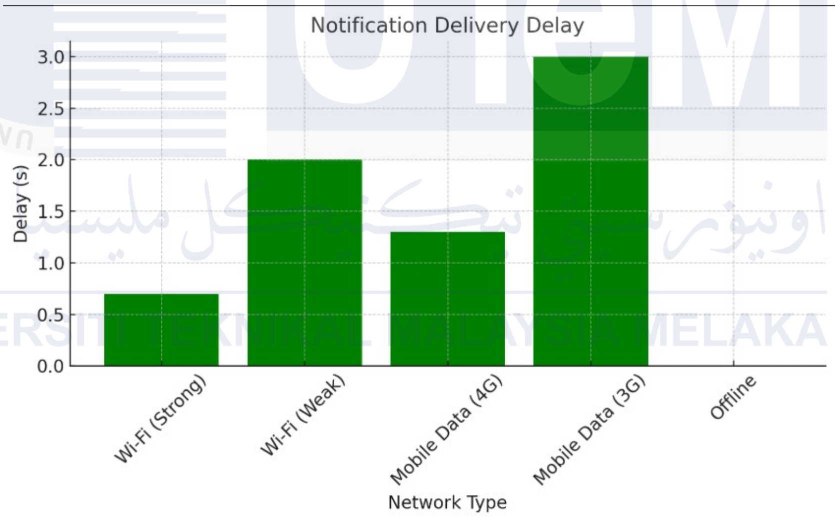

Figure 4.8: Figure shows notification delivery delay chart.

The notification delivery system operates efficiently under strong Wi-Fi conditions, achieving a minimal delay of only 0.7 seconds between motion detection and user notification. This allows for nearly instantaneous alerts, enabling rapid responses. However, as network strength decreases, the delays become more noticeable. For instance, weak Wi-Fi connections or 3G mobile data may result in delays of up to 3 seconds, which can impact the timeliness of responses in critical situations.

Additionally, offline conditions present a significant limitation, as the system cannot send notifications without an internet connection. This reliance underscores the need for a backup mechanism, such as local alarms or SMS notifications, to ensure users receive alerts even during connectivity issues. While the notification system is reliable in most scenarios, addressing its dependency on stable internet could further enhance its overall reliability.

4.6.2.4 Power Consumption Analysis

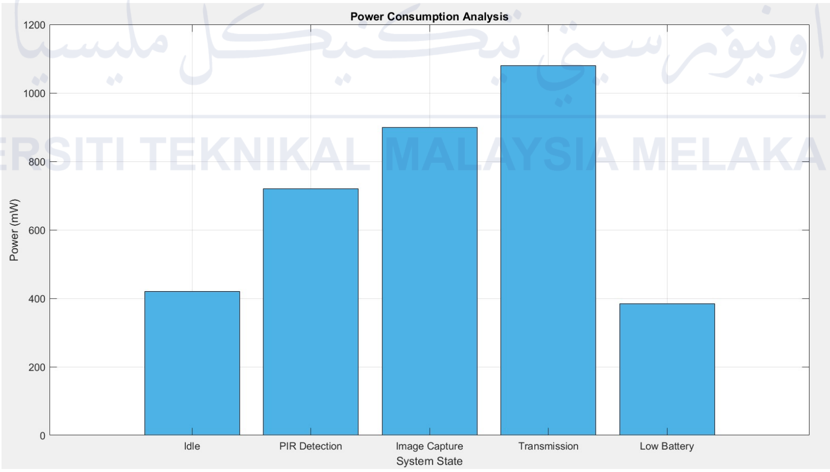

Figure 4.9: Figure shows power consumption analysis.

Power consumption analysis is essential for evaluating the performance and efficiency of Internet of Things (IoT) systems, including smart anti-theft devices. This involves measuring the energy use of the ESP32-CAM and its components, like the PIR sensor, in various states:

idle, motion detection, image capture, and data transmission. The ESP32-CAM shows significantly higher power consumption during image capture and data transmission compared to idle, affecting battery life and usability. The strategy that use to conserve the battery is by put the switch that can turn the the smart anti-theft system on and off.

This analysis helps predict battery life in different scenarios and optimize the system for longer use. Strategies such as enabling power-saving modes, lowering image resolution, and reducing data transmission frequency can effectively decrease energy consumption and extend battery life. This is crucial for battery-powered devices, ensuring they operate longer without frequent recharging or replacements while balancing performance and energy efficiency.

4.7 Summary

The analysis of the system's performance shows where it excels and where it could use some improvement. In controlled indoor settings, the motion detection accuracy is impressive at 96%. However, when it encounters crowded areas or outdoor environments, that accuracy drops because of challenges like overlapping heat signatures and changing temperatures. This suggests that we might need to look into better motion filtering to handle these dynamic conditions more effectively.

When it comes to image quality, the ESP32-CAM performs well in bright lighting, scoring a solid 9 for clarity. However, in dim or nighttime conditions, the quality suffers due to noise and low light. Adding infrared lighting or noise reduction features could really help improve the image capture in these situations.

Notification delivery is fairly quick with strong Wi-Fi connections, averaging about 0.7 seconds. This means you can expect timely alerts. On the downside, if the Wi-Fi signal is weak or if you're using mobile data, the delays increase, and you won't receive notifications at all if you're offline. This really highlights how much the system depends on a stable internet connection. A good solution might be to implement a backup feature, like SMS alerts, to fill in those gaps.

As for power consumption analysis, it indicates that the system's energy efficiency varies depending on its operational state. For example, the system uses minimal power in idle mode, but significantly more during image capture and data transmission. This fluctuation directly affects battery life, particularly in high-activity scenarios. Implementing optimizations, such as sleep mode or configurations that consume less power, could extend battery life, making the system more practical for long-term use.

Overall, although the system has a solid foundation, addressing its power consumption and enhancing its performance in other areas could increase its versatility and reliability.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This project successfully developed a smart anti-theft system utilizing IoT technology, specifically integrating the ESP32-CAM with a Telegram bot for enhanced surveillance and communication. The primary objectives outlined in Section 1.3 were achieved as follows.

The first objective is achieved by developing an IoT based anti-theft detection system using ESP32-CAM. The goal was achieved through the development of a system that has the ability to capture images and promptly notify the user via Telegram.

The second objective is achieved by designing a user-friendly interface for system configuration and remote monitoring. Through the integration of the Telegram bot, users gained the capability of instant communication, enabling them to remotely monitor their premises and receive immediate alerts if any unauthorized access or suspicious activities occurred.

The third objective is achieved by evaluating the effectiveness of the system. Comprehensive testing confirmed the system's efficiency in detecting unauthorized activities and recording images, providing a strong security solution.

5.2 Future Works

The existing project has laid a strong foundation for an intelligent anti-theft system, but there are several areas that offer potential for further exploration and progress:

1. Future research may focus on combining image processing algorithms and machine learning to improve the accuracy of identifying suspicious behaviors, including investigating new methods and technologies to enhance surveillance system detection capabilities and performance.

2. Integrating of supplementary sensors, such as those for sound or vibration, has the potential to improve the system's capacity in identifying various forms of intrusions.

3. Developing a mobile app with enhanced control and customization options improves user experience by offering personalized interactions and convenient access on smartphones and tablets.

5.3 Project Commercialization

To successfully enter the market, it is crucial to conduct thorough market research for the smart anti-theft system project. By identifying the target customers and understanding their needs, we can customize the product features to better meet market demands. In addition, creating multiple prototypes and gathering feedback from initial users will provide valuable insights for further improvements.

It is essential to conduct a thorough cost analysis to ensure the competitiveness and quality of the product. It is also crucial to collaborate with security companies and seek funding from investors or government grants for innovative technology solutions. Developing a comprehensive marketing and sales strategy, including online marketing, trade show participation, and demonstration videos, will effectively reach potential customers and establish the smart anti-theft system as a valuable product in the security market.

REFERENCES

[1] Ahmad, "Jenayah indeks Melaka meningkat 2.44 peratus," 2023.

[2] R. Santika, B. A. Pramudita, N. F. Puspita, and S. Sumaryo, "Laptop anti-theft system with tracking and image capture device based on internet of things technology," in *2023 International Conference on Data Science and Its Applications (ICoDSA)*, 2023, pp. 53–58.

[3] E. Effah, G. E. Y. Okai, and J. K. Aidoo, "Smart anti-theft water metering system," in *2022 IEEE/IET International Utility Conference and Exposition (IUCE)*, 2022, pp. 1–5.

[4] P. K. Sattaru, K. V. Burugula, R. Channagiri, and S. Kavitha, "Smart home security system using iot and esp8266," in *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2023, pp. 469–474.

[5] A. Karnik, D. Adke, and P. Sathe, "Low-cost compact theft-detection system using mpu-6050 and blynk iot platform," in *2020 IEEE Bombay Section Signature Conference (IBSSC)*, 2020, pp. 113–118.

[6] T. Rani, S. Sakthy, P. Kalaichelvi, T. Vignesh, and M. Priyadharshan, "Home security and anti-theft system," in *2022 1st International Conference on Computational Science and Technology (ICCST)*, 2022, pp. 130–133.

[7] G. Verma, S. Pachauri, A. Kumar, D. Patel, A. Kumar, and A. Pandey, "Smart home automation with smart security system over the cloud," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023, pp. 1–7.

[8] K. S. Keerthi, A. A. Yadwad, K. S. Kumar, Amandeep, and A. M, "Independent, integrated, reconfigurable iot based home security system," in *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2022, pp. 27–34.

[9] M. N. M. Razali, N. A. Abdul-Kadir, and J. Kasim, "Smart home security with dual modes," in *2020 IEEE Student Conference on Research and Development (SCOReD)*, 2020, pp. 453–458.

[10] M. I. M. Abu.Zaid, R. Abdullah, S. I. Ismail, and N. N. S. N. Dzulkefli, "Iot-based emergency alert system integrated with telegram bot," in *2023 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, 2023, pp. 126–131.

[11] J. C. de Oliveira, D. H. Santos, and M. P. Neto, "Chatting with arduino platform through telegram bot," in *2016 IEEE International Symposium on Consumer Electronics (ISCE)*, 2016, pp. 131–132.

APPENDICES

Appendix A: Full coding for smart anti-theft system.

```

#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include <Wire.h>

const int hour_to_flash = 20; // hour to enable autoflash
const int hour_till_flash = 6; // hour to enable autoflash

#include <TimeLib.h>
bool checkHour = false; // check hour to automatic enable the flash
unsigned long offset_days = 3;    // 3 days
unsigned long t_unix_date1, t_unix_date2;


int armed = true;

const String photo_str = "Photo";
const String state_str = "State";
const String arm_str = "Arm";
const String disarm_str = "Disarm";
const String ledon_str = "LedON";
const String ledoff_str = "LedOFF";
const String autoflash_str = "AutoFlash";
```

```
const String flashOn_str = "Flash ON";
const String flashOff_str = "Flash OFF";

const String keyboardJson = "[[\"" + photo_str + "\", \"" + state_str + "\"],[\"" + flashOn_str + "\"



// Replace with your network credentials
const char* ssid = "RemiNote";
const char* password = "1sampai8";

// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you
String chatId = "5057903264";

// Initialize Telegram BOT
String BOTtoken = "8000092277:AAGNtm6s1M9glL0XAzJZKuNTKUsdhrLTOZU"; // put your bot token here

bool sendPhoto = false;

bool flashEnabled = false;


WiFiClientSecure clientTCP;

UniversalTelegramBot bot(BOTtoken, clientTCP);

//CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
```

```
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

#define FLASH_LED_PIN 4
bool flashState = LOW;

// Motion Sensor
bool motionDetected = false;

int botRequestDelay = 1000;   // mean time between scan messages
long lastTimeBotRan;     // last time messages' scan has been done

void handleNewMessages(int numNewMessages);
String sendPhotoTelegram();


// Indicates when motion is detected
static void IRAM_ATTR detectsMovement(void * arg) {
  //Serial.println("MOTION DETECTED!!!");
  motionDetected = true;
}
```

```

void setup() {
  checkHour = false;
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  Serial.begin(115200);

  pinMode(FLASH_LED_PIN, OUTPUT);
  digitalWrite(FLASH_LED_PIN, flashState);
  gpio_hold_dis(GPIO_NUM_4);




  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for api.telegram.org
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  WiFi.hostname("AimanCamera");
  Serial.println();
  Serial.print("ESP32-CAM IP Address: ");
  Serial.println(WiFi.localIP());

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
```

```
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  //init with high specs to pre-allocate larger buffers
  if (psramFound()) {
    config.frame_size =  FRAMESIZE_VGA;//FRAMESIZE_UXGA;
    config.jpeg_quality = 10;  //0-63 lower number means higher quality
    config.fb_count = 4;//2;
  } else {
    config.frame_size = FRAMESIZE_VGA;//FRAMESIZE_UXGA;
    config.jpeg_quality = 10;//12;  //0-63 lower number means higher quality
    config.fb_count = 2;//1;
  }

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
```

```
    ESP.restart();
  }

  // Drop down frame size for higher initial frame rate
  //sensor_t * s = esp_camera_sensor_get();
  //s->set_framesize(s, FRAMESIZE_CIF);  // UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA

  sensor_t * s = esp_camera_sensor_get();
  //initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1);//flip it back
    s->set_brightness(s, 1);  //up the blightness just a bit
    s->set_saturation(s, -2); //lower the saturation
    s->set_sharpness(s, 2);   //Up THe sharpness
  }
  //drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_SXGA);



  // PIR Motion Sensor mode INPUT_PULLUP
  //err = gpio_install_isr_service(0);
  err = gpio_isr_handler_add(GPIO_NUM_13, &detectsMovement, (void *) 13);
  if (err != ESP_OK) {
    Serial.printf("handler add failed with error 0x%x \r\n", err);
  }
  err = gpio_set_intr_type(GPIO_NUM_13, GPIO_INTR_POSEDGE);
  if (err != ESP_OK) {
    Serial.printf("set intr type failed with error 0x%x \r\n", err);
  }
```

```
}

void loop() {
  if (sendPhoto) {
    Serial.println("Preparing photo");
    sendPhotoTelegram();
    sendPhoto = false;
  }

  if (armed && motionDetected) {
    bot.sendMessage(chatId, "Motion detected!!", "");
    Serial.println("Motion Detected");
    sendPhotoTelegram();
    motionDetected = false;
  }

  if (millis() > lastTimeBotRan + botRequestDelay) {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
      Serial.println("got response");
      handleNewMessages(numNewMessages);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
  }

}

String sendPhotoTelegram() {
  const char* myDomain = "api.telegram.org";
  String getAll = "";
  String getBody = "";

```

```
if (flashEnabled) {
  digitalWrite(FLASH_LED_PIN, HIGH);
}

camera_fb_t * fb = NULL;
fb = esp_camera_fb_get();
if (!fb) {
  Serial.println("Camera capture failed");
  delay(1000);
  ESP.restart();
  return "Camera capture failed";
}

Serial.println("Connect to " + String(myDomain));

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

  String head = "--aiman\r\nContent-Disposition: form-data; name=\"chat_id\"; \r\n\r\n" + chatId +
  String tail = "\r\n--aiman--\r\n";

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

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

  uint8_t *fbBuf = fb->buf;
```

```
    size_t fbLen = fb->len;
    for (size_t n = 0; n < fbLen; n = n + 1024) {
      if (n + 1024 < fbLen) {
        clientTCP.write(fbBuf, 1024);
        fbBuf += 1024;
      }
      else if (fbLen % 1024 > 0) {
        size_t remainder = fbLen % 1024;
        clientTCP.write(fbBuf, remainder);
      }
    }

    clientTCP.print(tail);

    esp_camera_fb_return(fb);

    int waitTime = 10000;   // timeout 10 seconds
    long startTimer = millis();
    boolean state = false;

    while ((startTimer + waitTime) > millis()) {
      Serial.print(".");
      delay(100);
      while (clientTCP.available()) {
        char c = clientTCP.read();
        if (state == true) getBody += String(c);
        if (c == '\n') {
          if (getAll.length() == 0) state = true;
          getAll = "";
        }
        else if (c != '\r')
          getAll += String(c);
        startTimer = millis();
```

```
      }
      if (getBody.length() > 0) break;
    }
    clientTCP.stop();
    Serial.println(getBody);
  }
  else {
    getBody = "Connected to api.telegram.org failed.";
    Serial.println("Connected to api.telegram.org failed.");
  }

  if (flashEnabled) {
    delay(2000);
    digitalWrite(FLASH_LED_PIN, LOW);
  }
  return getBody;
}

void handleNewMessages(int numNewMessages) {
  Serial.print("Handle New Messages: ");
  Serial.println(numNewMessages);

  for (int i = 0; i < numNewMessages; i++) {
    // Chat id of the requester
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != chatId) {
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);
```

```
telegramMessage message = bot.messages[i];
int now_hour = getHourFromTelegram(message); // get hour
if ((checkHour) && (now_hour >= hour_to_flash &&  now_hour <= hour_till_flash)) {
  flashEnabled = true;
}


String from_name = bot.messages[i].from_name;
const String welcome = "Welcome, " + from_name + ". Choose from one of the following options\n";
if (text == photo_str) {
  sendPhoto = true;
  Serial.println("New photo  request");
}
else if (text == ledon_str) {
  bot.sendMessageWithReplyKeyboard(chat_id, "LED state set to ON", "", keyboardJson, true);
  //bot.sendMessage(chat_id, "LED state set to ON", "");
  digitalWrite(FLASH_LED_PIN, HIGH);
}
else if (text == ledoff_str) {
  bot.sendMessageWithReplyKeyboard(chat_id, "LED state set to OFF", "", keyboardJson, true);
  //bot.sendMessage(chat_id, "LED state set to OFF", "");
  digitalWrite(FLASH_LED_PIN, LOW);
}
else if (text == arm_str) {
  armed = true;
  bot.sendMessageWithReplyKeyboard(chat_id, "Security System is ON", "", keyboardJson, true);
  //bot.sendMessage(chat_id, "Security System is ON", "");
}
else if (text == disarm_str) {
  armed = false;
  bot.sendMessageWithReplyKeyboard(chat_id, "Security System is OFF", "", keyboardJson, true);
  //bot.sendMessage(chat_id, "Security System is OFF", "");
```

```
    }
    else if (text == autoflash_str) {
      checkHour = !checkHour;
      if(checkHour){
        bot.sendMessageWithReplyKeyboard(chat_id, "Flash will be automatically enabled after 20:00",
        //bot.sendMessage(chat_id, "Flash will be automatically enabled after 20:00", "");
      }else{
        bot.sendMessageWithReplyKeyboard(chat_id, "Automatic flash disabled", "", keyboardJson, true)
        //bot.sendMessage(chat_id, "Automatic flash disabled", "");
      }
      flashEnabled = false;
    }
    else if (text == flashOn_str) {
      flashEnabled = true;
      bot.sendMessageWithReplyKeyboard(chat_id, "Flash enabled for photos", "", keyboardJson, true);
      //bot.sendMessage(chat_id, "Flash enabled for photos", "");
    }
    else if (text == flashOff_str) {
      flashEnabled = false;
      bot.sendMessageWithReplyKeyboard(chat_id, "Flash disabled for photos", "", keyboardJson, true);
      //bot.sendMessage(chat_id, "Flash disabled for photos", "");
    }
    else if (text == state_str) {
      String flashEnabledStr = "";
      String motionEnabledStr = "";


      if (flashEnabled) flashEnabledStr = "yes";
      else  flashEnabledStr = "no";

      if (armed)  motionEnabledStr = "yes";
      else  motionEnabledStr = "no";

```

```

      String stat = "Connected to: " + String(ssid) +". Rssi: " + String(WiFi.RSSI()) + "\nip: " +  W
      bot.sendMessageWithReplyKeyboard(chat_id, stat, "", keyboardJson, true);
    }
    else {

      //String welcome = "Welcome, " + from_name + ".\n";
      //welcome += "Use the following commands to control your outputs.\n\n";
      //welcome += "/photo to take a picture \n";
      //welcome += "/arm to arm the security system \n";
      //welcome += "/disarm to disarm the security system \n";
      //welcome += "/ledon to turn flash light always on\n";
      //welcome += "/ledoff to turn flash light always off\n";
      //welcome += "/autoflash to turn on flash automatically at 20:00\n";
      //welcome += "/state to request current GPIO state and security system state \n";
      //bot.sendMessage(chat_id, welcome, "Markdown");



      bot.sendMessageWithReplyKeyboard(chat_id, welcome, "", keyboardJson, true);
    }
  }
}

int getHourFromTelegram(telegramMessage message) {
  String now_date = message.date; // get now date

  unsigned long offset_days = 3;    // 3 days
  unsigned long t_unix_date1 = strtol(now_date.c_str(), NULL, 10);;
  int now_hour = hour(t_unix_date1);
  return now_hour + 8;  // current hour in GMT+8
}
```