

UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior

Bachelor's Degree in Computer Science and Engineering



Trabajo de Fin de Grado

Sistema de Detección de Intrusión para Dispositivos Empotrados de
Bajo Consumo

Bachelor Thesis

Intrusion Detection System for Low Consumption Embedded Devices

Author: Irina Camacho Sánchez

Tutor: Francisco Javier García Blas



This work is licensed under Creative Commons **Attribution – Non Commercial – Non Derivatives**

ABSTRACT

The popularization of smartphones, tablets and personal computers is steering the digital transformation towards the necessities of end users, aiming to provide ease in their everyday tasks. As technology moves forward, the possibility to offer users total management of their home using a single device is becoming more popular. More devices are entering the market every day, such as lights, motors, intruder detection systems, etc. Nowadays, the deployment of affordable devices has become a necessity, given the current economic state.

With the assist of microcontrollers, and all the new components that are being introduced regularly in the market, it has become possible for users to develop their own systems for their home, and as such, decide upon the different technologies they require based on their needs.

This Bachelor Thesis details the process of obtaining an affordable approach to an intruder detection system. The system will be deployed in a Raspberry Pi, one of the most commonly used microcomputers in the world; and will use an Arduino UNO connected to a PIR sensor for the motion detection. With the help of the Telegram service, we will deploy a Telegram bot that will provide a message service to the user with a friendly and easy to use interface. This application will provide the end user similar functionalities as those systems that are already in the market.

KEYWORDS

Raspberry Pi, Arduino, Passive Infra-Red Sensor (PIR), Telegram API, Telegram Bot.

RESUMEN

La popularización de los Smartphones, tablets y ordenadores personales está dirigiendo la transformación digital hacia las necesidades del usuario final, tratando de proporcionar a los usuarios facilidad en la realización de sus tareas cotidianas. A medida que la tecnología avanza, la posibilidad de ofrecer a los usuarios la opción de controlar completamente de su hogar a través de un único dispositivo se hace cada vez más popular. Cada día aparecen más dispositivos en el mercado, desde luces, motores de movimiento y sistemas de detección de intrusos. Hoy en día, la instalación de dispositivos asequibles se ha convertido en una necesidad, dado el estado económico actual.

Con la ayuda de los microcontroladores y todos los nuevos componentes que se introducen regularmente en el mercado, los usuarios pueden desarrollar sus propios sistemas para su hogar y, como tal, decidir sobre las diferentes tecnologías que emplear de acuerdo a sus distintas necesidades.

Este Trabajo de Fin de Grado detalla el proceso para obtener una alternativa económica a los sistemas de detección de intrusos. El sistema se implementará en una Raspberry Pi, una de las microcomputadoras más utilizadas en el mundo; y utilizará un Arduino UNO conectado a un PIR para capturar la detección de movimiento. Con la ayuda de la API de Telegram, se implementará un Telegram bot que proporcionará al usuario una interfaz amigable y fácil de usar. Esta aplicación proporcionará al usuario final funcionalidades casi idénticas a los sistemas que ya existen en el mercado.

PALABRAS CLAVE

Raspberry Pi, Arduino, Passive Infra-Red Sensor (PIR), Telegram API, Telegram Bot.

ACKNOWLEDGEMENTS

After putting the final full-stop, I find I need to thank several people for everything they have done throughout these years.

First, to my parents, for all their support. For their understanding over endless nights confined in my bedroom, staring at my computer in frustration. For their help, and encouragement all this time.

To my “petarda”, my little sister, without whom, I would have given up many years ago. If not for her taking my clothes from my wardrobe without my permission, the idea for this project, would have never come to life.

To Javi, my thesis supervisor, for helping and guiding me throughout this project, coping with my endless questions.

To Xabe, my library rat. For all her days crying with me through the exam periods. For all her time, for keeping a smile on my face throughout the hardest days of the university. To Paula, for supporting me no matter what. For being there when I needed her the most, even when most people left. For being the best friends one could ever ask for.

Finally, to the best Computer Engineer I know, my Weekend Warrior. For all your infinite patience, for staying up on the phone with me for hours, encouraging me to keep going forward. Nacho, if I take something good from all these years, that is you.

“When you mark the sky and the sun comes through, know your greatest days are ahead of you.”

Table of Contents

1. INTRODUCTION.....	1
1.1. MOTIVATION	1
1.2. OBJECTIVES.....	2
1.3. DOCUMENT STRUCTURE.....	2
2. STATE OF THE ART.....	4
2.1. INTRUSION SYSTEMS.....	4
2.1.1. SYSTEMS IN THE MARKET.....	6
2.1.2. BUILT LOW-COST SYSTEMS	7
2.1.3. OTHER PROJECTS.....	8
2.2. EMBEDDED SYSTEMS	9
2.2.1. ARDUINO	10
2.2.2. RASPBERRY PI	11
2.2.3. ESP8266	12
2.3. COMMUNICATION PROTOCOLS IN EMBEDDED SYSTEMS.....	13
2.3.1. UART PROTOCOL	13
2.3.2. SPI PROTOCOL	14
2.3.3. I ² C PROTOCOL.....	15
2.4. INSTANT MESSAGING APPLICATIONS.....	16
3. SYSTEM ANALYSIS.....	18
3.1. DEFINITION OF THE SYSTEM	18
3.2. USE CASES.....	19
3.3. SYSTEM REQUIREMENTS	22
3.3.1. FUNCTIONAL REQUIREMENTS	23
3.3.2. NON-FUNCTIONAL REQUIREMENTS	27
3.4. TRACEABILITY MATRIX	30
4. SYSTEM DESIGN.....	31
4.1. DETAILED DESIGN.....	31
4.1.1. COMMUNICATION PARADIGM.....	31
4.1.2. CONNECTION TYPE.....	32
4.1.3. CONCURRENCY.....	32
4.1.4. STATE	32
4.1.5. NAMING.....	32
4.1.6. SECURITY	32
4.1.7. MESSAGE SEQUENCE.....	33
4.1.8. MESSAGE FORMAT.....	34
4.2. SYSTEM ARCHITECTURE.....	35
4.3. FLOWCHART	36
4.4. COMPONENT DIAGRAM	38
4.5. SEQUENCE DIAGRAM	41
4.6. CLASS DIAGRAM	45
5. IMPLEMENTATION.....	47

5.1. SYSTEM COMPONENTS	47
5.2. ARDUINO	48
5.3. RASPBERRY PI	49
5.4. CAMERA CONFIGURATIONS	50
5.5. ARDUINO RASPBERRY PI CONNECTION	50
5.6. SCRIPT IMPLEMENTATION	52
5.7. BOT CREATION.....	54
5.8. RUNNING THE SCRIPT.....	55
6. EVALUATION.....	56
6.1. UNIT TESTING	57
6.2. INTEGRATION TESTING.....	58
6.3. SYSTEM TESTING	59
6.4. TEST TRACEABILITY MATRIX	61
7. MANAGEMENT	62
7.1. PLANNING	62
7.2. BUDGET	65
7.2.1. DIRECT COSTS	65
7.2.2. INDIRECT COSTS	67
7.3.2. RISK COSTS.....	67
7.3.3. TOTAL COSTS	67
8. LEGAL AND SOCIO-ECONOMIC ENVIRONMENT	68
8.1. OPEN-SOURCING THE PROJECT	68
8.2. LEGAL ASPECTS	68
8.2.1. LEY DE SEGURIDAD PRIVADA	68
8.2.2. GENERAL DATA PROTECTION REGULATION	70
8.3. INTELECTUAL PROPERTY.....	71
8.3.1. GPL – Compatible Free Software License.....	71
8.3.2. MIT License.....	71
8.3.3. Apache.....	72
8.3.4. CDDL-1.0 - Common Development and Distribution License	72
8.3.5. LICENSE OF THE PROJECT	73
8.4. SOCIO-ECONOMIC IMPACT	73
9. CONCLUSIONS.....	76
9.1. PERSONAL CONCLUSIONS.....	76
9.2. FUTURE WORK.....	78
9.2.1. IMAGE SECURITY	78
9.2.2. AUTOMATIC FILE DELETION IN RPi.....	78
9.2.3. WIRELESS SYSTEM	78
9.2.4. CAMERA NIGHT-VISION AND ROTATION	79
9.2.5. FACE RECOGNITION SOFTWARE	79
Bibliography.....	80

ANNEX A: USER MANUAL.....	82
Software installation	82
Raspberry Pi.....	82
Arduino.....	83
Mobile devices.....	84
Software operation.....	85
ANNEX B: ARTÍCULO 7 LEY DE SEGURIDAD PRIVADA.....	87
ANNEX C: ARTÍCULO 427 LEY DE SEGURIDAD PRIVADA	88
ANNEX D: GENERAL DATA PROTECTION REGULATION [24]	89

INDEX OF FIGURES

FIGURE 1: UART PROTOCOL COMMUNICATION [12]	14
FIGURE 2: SPI COMMUNICATION PROTOCOL [13]	14
FIGURE 3: I2C COMMUNICATION PROTOCOL [14]	15
FIGURE 4: APPLICATION DESIGN	18
FIGURE 5: USE CASES 1	21
FIGURE 6: USE CASES 2	22
FIGURE 7: PASSIVE CLIENT - SERVER MESSAGE PASSING	33
FIGURE 8: ACTIVE CLIENT - SERVER MESSAGE PASSING	33
FIGURE 9: MODEL VIEW CONTROLLER (MVC) SCHEME	35
FIGURE 10: FLOWCHART	37
FIGURE 11: COMPONENT REPRESENTATION	38
FIGURE 12: COMPONENT DEPENDENCY REPRESENTATION	38
FIGURE 13: SUBSYSTEM REPRESENTATION	39
FIGURE 14: COMPONENT DIAGRAM	40
FIGURE 15: SEQUENCE DIAGRAM BOT CREATION	41
FIGURE 16: START SEQUENCE DIAGRAM	42
FIGURE 17: HELP SEQUENCE DIAGRAM	42
FIGURE 18: CLEAR SEQUENCE DIAGRAM	42
FIGURE 19: ALARM OFF SEQUENCE DIAGRAM	42
FIGURE 20: ALARM ON SEQUENCE DIAGRAM	43
FIGURE 21: PICTURE REQUEST SEQUENCE DIAGRAM	43
FIGURE 22: MOTION DETECTION SEQUENCE DIAGRAM	44
FIGURE 23: CLASS DIAGRAM	46
FIGURE 24: ARDUINO CIRCUIT	48
FIGURE 25: ARDUINO-RASPBERRY CIRCUIT	51
FIGURE 26: GANTT CHART	64

INDEX OF TABLES

TABLE 1: PRIVATE COMPANIES INTRUSION DETECTION SYSTEMS CHARACTERISTICS	6
TABLE 2: BUILT INTRUSION DETECTION SYSTEM CHARACTERISTICS	7
TABLE 3: SIMILAR PROJECTS CHARACTERISTICS	8
TABLE 4: ARDUINO UNO TECHNICAL CHARACTERISTICS	10
TABLE 5: RASPBERRY PI TECHNICAL CHARACTERISTICS	11
TABLE 6: ESP8266 TECHNICAL CHARACTERISTICS	12
TABLE 7: USE CASE TEMPLATE.....	19
TABLE 8: USE CASE UC-01.....	19
TABLE 9: USE CASE UC-02.....	20
TABLE 10: USE CASE UC-03.....	20
TABLE 11: USE CASE UC-04.....	20
TABLE 12: USE CASE UC-05.....	20
TABLE 13: USE CASE UC-06.....	20
TABLE 14: USE CASE UC-07.....	21
TABLE 15: USE CASE UC-08.....	21
TABLE 16: REQUIREMENT TEMPLATE.....	22
TABLE 17: FUNCTIONAL REQUIREMENT FR-01	23
TABLE 18: FUNCTIONAL REQUIREMENT FR-02.....	24
TABLE 19: FUNCTIONAL REQUIREMENT FR-03.....	24
TABLE 20: FUNCTIONAL REQUIREMENT FR-04.....	24
TABLE 21: FUNCTIONAL REQUIREMENT FR-05.....	24
TABLE 22: FUNCTIONAL REQUIREMENT FR-06.....	24
TABLE 23: FUNCTIONAL REQUIREMENT FR-07.....	25
TABLE 24: FUNCTIONAL REQUIREMENT FR-08.....	25
TABLE 25: FUNCTIONAL REQUIREMENT FR-09.....	25
TABLE 26: FUNCTIONAL REQUIREMENT FR-10.....	25
TABLE 27: FUNCTIONAL REQUIREMENT FR-11	25
TABLE 28: FUNCTIONAL REQUIREMENT FR-12.....	26
TABLE 29: FUNCTIONAL REQUIREMENT FR-13.....	26
TABLE 30: FUNCTIONAL REQUIREMENT FR-14.....	26
TABLE 31: FUNCTIONAL REQUIREMENT FR-15.....	26
TABLE 32: FUNCTIONAL REQUIREMENT FR-16.....	26
TABLE 33: FUNCTIONAL REQUIREMENT FR-17.....	27
TABLE 34: FUNCTIONAL REQUIREMENT FR-18.....	27
TABLE 35: NON-FUNCTIONAL REQUIREMENT NFR-01	27
TABLE 36: NON-FUNCTIONAL REQUIREMENT NFR-02.....	27
TABLE 37: NON-FUNCTIONAL REQUIREMENT NFR-03.....	28
TABLE 38: NON-FUNCTIONAL REQUIREMENT NFR-04.....	28
TABLE 39: NON-FUNCTIONAL REQUIREMENT NFR-05.....	28
TABLE 40: NON-FUNCTIONAL REQUIREMENT NFR-06.....	28
TABLE 41: NON-FUNCTIONAL REQUIREMENT NFR-07	28
TABLE 42: NON-FUNCTIONAL REQUIREMENT NFR-08.....	28
TABLE 43: NON-FUNCTIONAL REQUIREMENT NFR-09.....	29
TABLE 44: NON-FUNCTIONAL REQUIREMENT NFR-10.....	29
TABLE 45: NON-FUNCTIONAL REQUIREMENT NFR-11	29
TABLE 46: NON-FUNCTIONAL REQUIREMENT NFR-12.....	29
TABLE 47: NON-FUNCTIONAL REQUIREMENT NFR-13.....	29

TABLE 48: NON-FUNCTIONAL REQUIREMENT NFR-14.....	29
TABLE 49: TRACEABILITY MATRIX	30
TABLE 50: FLOWCHART NOTATION.....	37
TABLE 51: SYSTEM COMPONENTS	47
TABLE 52: TEST TEMPLATE.....	56
TABLE 53: UNIT TEST UT-01	57
TABLE 54: UNIT TEST UT-02	57
TABLE 55: UNIT TEST UT-03	57
TABLE 56: UNIT TEST UT-04	57
TABLE 57: INTEGRATION TEST IT-01	58
TABLE 58: INTEGRATION TEST IT-02	58
TABLE 59: INTEGRATION TEST IT-03	58
TABLE 60: SYSTEM TEST ST-01	59
TABLE 61: SYSTEM TEST ST-02.....	59
TABLE 62: SYSTEM TEST ST-03.....	59
TABLE 63: SYSTEM TEST ST-04.....	59
TABLE 64: SYSTEM TEST ST-05.....	59
TABLE 65: SYSTEM TEST ST-06.....	60
TABLE 66: SYSTEM TEST ST-07.....	60
TABLE 67: SYSTEM TEST ST-08.....	60
TABLE 68: SYSTEM TEST ST-09.....	60
TABLE 69: SYSTEM TEST ST-10.....	60
TABLE 70: SYSTEM TEST ST-11	61
TABLE 71: TEST-REQUIREMENTS TRACEABILITY MATRIX.....	61
TABLE 72: HUMAN RESOURCES BUDGET.....	66
TABLE 73: MATERIAL COST BUDGET	66
TABLE 74: INDIRECT COSTS BUDGET	67
TABLE 75: MARKET PRICES COMPARISON TABLE.....	74
TABLE 76: SYSTEM PRICE TABLE	74
TABLE 77: ELECTRICITY COSTS TABLE	74
TABLE 78: SYSTEMS CHARACTERISTICS COMPARISON.....	77

1. INTRODUCTION

This first chapter provides a general overview of the project. It is divided into three parts: the first part includes an introduction to the project as well as the motivation behind its development; the second part will outline the objectives that we wish to achieve; and the later describes the structure that the document will follow.

1.1. MOTIVATION

Delinquent acts in Spain have increased 73% in the last eight years. Last year alone, over 40.000 homes were summited to burglary, according to the Ministerio de Interior [1].

The possibility to deploy a surveillance system, gives the user a higher sense of security. One of the most attracting qualities of the systems in the market, is the possibility to have your home under surveillance 24 hours a day. However, in order to have that opportunity, the user must pay to the companies that provide it, a starting amount varying from 300 euros to even 1,000, just for the installation; plus, maintenance.

With the popularization of Smartphones, the amount of people that own a Smartphone in Spain has increased, from 44% in 2012 - to 87% in the year 2017, according to Google's Consumer Barometer [2].

Home automation or domotics, and Internet of Things (IoT) are fields in technology, which can provide a household security system. Those technologies enable the possibility to control all the different aspects of a home. With the help of smartphones, that possibility is in the reach of your pocket, becoming more popular every day. So, if we are capable of controlling every single light in our home, why not produce a cheap alternative to security companies, and build an intrusion detection system that can be managed through our smartphone.

1.2. OBJECTIVES

The addressed objective is to create a security system that can be used in any household and ease the user the access to manage a complete and economic intrusion detection system.

The aim of the project is, to provide an economic substitute to modern security systems and to provide a software that can be managed by any user.

In order to achieve this main goal, a Raspberry Pi and an Arduino will be used, some of the most used micro-computers in the world. The possibilities that a Raspberry Pi offers are endless, from using it as a local server, to the case of building a security system. In order to detect an Intruder, an Arduino will be used, connected to a Passive Infra-Red Sensor (PIR). For the user to be capable of communicating and interacting with the system, the Telegram messaging application will be used.

The underlying objectives to be attained in order to reach the main goal will be:

- Present the user with the different functionalities that the system will have with both an interactive keyboard, and the option to insert it directly through a command in Telegram.
- Inform the users when a movement is detected in their home/office.
- Allow the user to turn on and off the alarm system from wherever they may be.
- Allow the user to take pictures at any time as long as the alarm system is turned on.
- Ease the user the option to remove the files generated by the camera by sending a simple command.

1.3. DOCUMENT STRUCTURE

This document is divided into different chapters that allow to explain all the information that is related to the development and building of the project. This information will be divided into main topics, and sub-topics. An overall vision on what each topic of the document will contain is shown below.

Over the first section, *Introduction*, the motivations to undergo this project, as well as the objectives defining what is desired to be accomplished are stated. This section also includes the structure that the document will follow.

The second section, *State of the Art*, presents a study of the current state of the technology that is used in the project, as well as a comparison of other systems similar to this one that are already in the market.

The third section, *Analysis*, details the scope of the project. It defines the different requirements that constrain the system, the different use cases that the system needs to be able to perform, and the tests required in order to ensure that the system is working according to the necessary specifications.

The fourth section, *Design*, will depict the system architecture, including an in-depth study of the pieces which are part of it. It will also show various diagrams, entailing how the software works.

The fifth section, *Implementation*, will explain the most relevant aspects from the implementation process.

The sixth section, *Evaluation*, will outline the tests done in the system, and if the system works correctly with all the requirements specified in the Analysis section.

The seventh section, *Management*, will detail the organization and the budget that is estimated to build the project.

The eighth section, *Legal and Socio-Economic Environment*, will talk about the **socio-economic impact** that this product can do in the market, and the **legal aspects** that are related to it, and affect it either directly or indirectly.

The ninth section, *Conclusions and Future Work*, will summarize the undergone project, and the main ideas that can be inferred from its development, including future improvements and updates that can be included in the project.

Finally, the *Annex*, is a bonus section, which entails a user-installation manual, so that anyone who wishes to benefit from the system, can do so, following simple step-by-step instructions, including diagrams and pictures of the circuit. It also includes the various articles of the current laws that govern the development and use of the project.

2. STATE OF THE ART

This chapter will entail all the information obtained after performing a study of the different technologies already present, that could be useful for the later implementation of the project.

In order to do so, first, the intrusion detection systems in the market will be explained, including their different components. The different type of intrusion detection systems, and their characteristics will also be explained.

Second, embedded systems and their applications will also be explained. This part will also contain several microcontrollers and their characteristics, in order to later perform a decision on those that will be used for the implementation of the project.

Third, the most common communication protocols between embedded systems will be detailed, including their advantages and disadvantages.

Finally, instant messaging applications will be depicted, along with their characteristics and possible uses.

2.1. INTRUSION SYSTEMS

This section will detail intrusion detection systems and their different components. The sub-sections that will follow will explain the advantages, disadvantages and characteristics of the systems in the market.

Home Intrusion Detection Systems are programs that monitor unauthorized access to homes or offices, informing the owner when said access has been committed, along with image evidence, a message or an alert. These systems are networks of integrated electronic devices, normally equipped with one or more motion detection sensors, connected to a controller that, either sets off an alarm sound, triggers a camera in order to capture images of the intruder, or both.

What makes these systems so appealing is the possibility to interact with them from anywhere in the world due to the technology of automation. Whether the user has realized they left their home without setting the alarm, or whether they have just locked

the door behind them, these systems allow the user the possibility to have their home protected within reach.

Entry level intrusion detection systems normally use windows or doors, equipped with motion detectors to recognize security breaches. These breaches normally trigger an alarm sound, to alert other people that may be close by in order for them to call the police, and even inform the user of when a breach has taken place.

These systems have integrated surveillance networks, introducing cameras providing images and video, allowing the users to have ultimate control over monitoring their homes or offices. These cameras capture images and or video footage of the culprits entering the premises.

Typical home intrusion detection systems include:

- A control panel: computer system that arms or disarms the security system and communicates with all the different components in the network.
- Doors and or window sensors: these sensors provide two parts that are adjacent to one another, when the door or window is locked the two parts of the system are joined.
- Motion sensors: these components are normally equipped with infra-red sensors, which measure the infra-red levels of the surroundings and trigger the alarm when said level varies.
- Wired or wireless cameras: these can be used for several scenarios:
 - Observe property surroundings.
 - Observe property entry points.
 - Capture images and or video footage of a security breach.
- Alarms: these devices allow a high-decibel sound to be emitted in order to inform neighbors of the situation.

These systems are not only available to users through professional monitored companies, but also through built-in systems available for sale and other alternative projects.

2.1.1. SYSTEMS IN THE MARKET

Intrusion systems in the market are normally provided by Security Companies, each of which grant different services.

The main advantages of these systems are:

- **Monitored Home Alarms:** systems connected to a third party, which communicate with law-enforces to provide extra security.
- **Stickers:** companies provide stickers to inform passers-by and burglars that the system is connected to a security company and that breaches will be informed automatically. These stickers also provide passers-by with information about executing their image rights.
- **Back-up systems:** systems that allow power supply for up to 24 hours in case the power supply is down.
- **Application management:** the user is capable of monitoring the system from their mobile phone or computer.

The main disadvantages of these systems are:

- **Costs:** Simple systems consisting only of motion sensors and cameras can cost from 300 euros. If a user desires more sensors, this cost will rise.
- **Maintenance fees:** users must pay monthly for the system to be available for their use. These fees start at 50 euros a month.

Below, a table providing information of the services provided by the main companies in the Spanish market is presented – check symbols represent that the service is provided.

Company	Response Time	Maintenance	App control	Image capture	Video capture	Motion sensors
Securitas Direct	30 min	✓	✓	✓	✓	✓
Segur24	30 min	✓			✓	✓
Alartec	50 min	✓	✓		✓	
Visegur	60 min	✓	✓		✓	

TABLE 1: PRIVATE COMPANIES INTRUSION DETECTION SYSTEMS CHARACTERISTICS

2.1.2. BUILT LOW-COST SYSTEMS

Given the high cost of company-provided security systems, the market has established a new trend of lower-cost systems that allow users to have a security system with similar characteristics to those provided by private companies, avoiding the monthly maintenance fees.

The main advantages of these systems are:

- **Wireless systems:** most of these systems are comprised of wireless devices interconnected in the network.
- **Easy installation:** these systems provide a user guide for the user to install the system, with easy step-by-step instructions.
- **Costs:** fixed cost for the provided materials, these costs are normally lower than those from private companies.
- **Back-up systems:** systems that allow power supply for up to 24 hours in case the power supply is down.
- **Application management:** the user is capable of monitoring the system from their mobile phone or computer.

The main disadvantages are:

- **No monitoring:** these systems do not provide direct communication with law-enforces.
- **Fixed packages:** the packages sold are normally completely configured, therefore it does not allow for extra sensors, unless they are from the same vendor.

Below, a table providing the main characteristics of some of the most sold systems – check symbols represent that the service is provided.

System	Response Time	App control	Image capture	Video capture	Motion sensors
G5 Touch	User	✓	✓	✓	✓
Blaupunkt SA2700	User	✓	✓	✓	✓
Netgear Arlo Pro	User	✓	✓	✓	

TABLE 2: BUILT INTRUSION DETECTION SYSTEM CHARACTERISTICS

2.1.3. OTHER PROJECTS

The approaches to build a system similar to those that already exist in the market, have not been many. After performing a wide research, only two similar projects have been found.

System 1 (Eren Golge): Video-vigilance system based on a Raspberry Pi, that monitors what is happening in the place where the system is deployed. This system alerts the user via electronic mail and allows the monitoring through a personal computer [3].

System 2 (Ignacio Bartolomé Tabanera): Video-vigilance system based on a Raspberry Pi that monitors access to a residence or office. This system alerts the user via electronic mail and allows the monitoring through a web-page [4].

These two systems provide a high disadvantage, they do not allow the user an active access and control over their system. Users must perform the monitoring of their system through a web-page.

The main advantage of both systems is the cost, both systems include only a Raspberry Pi, a Motion Sensor and a camera, which allow users a very flexible price.

Below, a table providing the main characteristics of these systems – check symbols represent that the service is provided.

System	Response Time	App control	Image capture	Video capture	Motion sensors
System 1	User	X	✓	✓	✓
System 2	User	X	✓	✓	✓

TABLE 3: SIMILAR PROJECTS CHARACTERISTICS

2.2. EMBEDDED SYSTEMS

Within the era of information revolution, new technologies arise to serve the needs of users. One of the many advancements in the field is the development of embedded systems (also known as microcontrollers). Embedded systems were designed as small computers meant to be introduced into a product, as such, they are hardly ever seen by the users. "If we take any engineering product that needs control and if a computer is incorporated within that product to undertake the control, then we have an embedded system" [5].

Embedded systems are given such name due to the fact that they are normally a computer system with a dedicated function embedded as part of a larger device. Often, embedded systems include real-time computing.

The automation of tasks became simpler with the introduction of these devices in the market. It is very easy to find these systems in almost every household, even without the knowledge of the user, given that these devices have been implanted in the electrical appliances since the 1980's. As such, these devices now form a large part of home automation systems.

The applications of these systems are endless - they do not end with appliances - nowadays several embedded systems have entered the market to allow users to create their own applications, whether for their home, or even for school projects.

One of the many benefits of these so-called microcontrollers is the price, it allows you to have a small computer in a single integrated circuit. Microcontrollers vary in format, with most common being 8-bit, 16-bit and 32-bit. Microcontrollers contain one or more CPU's, memory and programmable peripherals (input and output); as such they can be used as an embedded system, allowing them to form part of circuits at a very economical price. 8-bit microcontrollers were selling at a rate of 2 Billion units per year from 1997 up until the early 2010's [6].

2.2.1. ARDUINO

Arduino is a development platform, with a set of microcontrollers with open-source electronic platform. Arduino boards are capable of reading inputs and transforming them into outputs. These microcontrollers can be programmed to read inputs (through sensors, buttons, and messages) and transform said information into outputs (activating a sensor, turning on a LED, or sending messages).

Arduino has its own programming language Arduino (which is based on Wiring) and their own Integrated Development Environment (IDE), the Arduino Software, based on Processing [7].

The main characteristics of the Arduino board are:

- Inexpensive: the most expensive Arduino board is worth 50 euros.
- Open-Source: the software is published as open-source, allowing developers to contribute to its development. The hardware is published under Creative Common license, allowing circuit designers to develop their own.
- Cross-Platform: runs on Windows, Linux and Mac-OS.

With all this, Arduino offers many boards, the most basic being Arduino UNO, with the following technical characteristics:

Operating Voltage	5V	Flash Memory	32 KB (ATmega328P)
Digital Input/Output Pins	14	SRAM	2 KB (ATmega328P)
PWM Digital I/O Pins	6	EEPROM	1 KB (ATmega328P)
Analog Input Pins	6	Clock Speed	16MHz
Length	68.6mm	I/O Pin Current	20mA
Width	53.4mm	3.3V Pin Current	50mA
Weight	25g		[8]

TABLE 4: ARDUINO UNO TECHNICAL CHARACTERISTICS

2.2.2. RASPBERRY PI

The Raspberry Pi is a low-cost microcontroller with an open-source electronic platform. It allows to be plugged into a computer monitor. The Raspberry Pi resembles many aspects from a desktop computer, it allows users to program, browse the internet and even play video games.

This microcontroller has the ability to interact with the outside world and has been part of many digital projects.

The main characteristics of the Raspberry Pi are:

- **Inexpensive:** the most expensive Raspberry Pi is worth 50 euros with all the basic accessories (SD card, power cable, and case).
- **Open-Source:** the software is published as open-source, allowing developers to contribute to its development. The hardware however is not completely open-source.
- **Cross-Platform:** runs on Windows and Linux.
- **Programming Languages:** Python and Scratch

The main operating system distribution for the Raspberry Pi is Raspbian, a Debian-based Linux distribution.

With all this, Raspberry Pi offers many boards, the most recent being Raspberry Pi 3 Model B+, with the following technical characteristics:

Operating Voltage	5V	Flash Memory	32 KB (ATmega328P)
GPIO Pins	40	SDRAM	1GB
CSI Camera Port		Wireless LAN	2.4GHz and 5GHz
DSI Display		Bluetooth	4.2 BLE
Length	82mm	Ethernet	Gigabit
Width	56mm	4 USB Ports	USB 2.0
Weight	50g		[9]

TABLE 5: RASPBERRY PI TECHNICAL CHARACTERISTICS

2.2.3. ESP8266

The ESP8266 WROOM Wi-Fi Module is a low-cost microcontroller, based on an ARMv7 processor, with an open-source electronic platform. Given its processor, this chip achieves extra-low power consumption. It has a Real-Time Operating System, with combined with the Wi-Fi module, allows 80% of its power to be available for the user programming and developments [10].

This microcontroller is one of the most widely used microcontrollers in the field of IoT (Internet of Things), home-automation and even Wearables. This is due to its compatibility with the TCP/IP protocol, allowing the microcontroller to be connected to any network.

The main characteristics of the ESP8266 are:

- Inexpensive: the most expensive chip is worth 5 euros.
- Open-Source: the software is published as open-source, allowing developers to contribute to its development. The hardware is published under Creative Commons license, allowing circuit designers to develop their own.
- Protocol support: IPv4, TCP, UDP, HTTP and FTP

ESP8266 offers the following technical characteristics:

Operating Voltage	3V	Network Protocol	IPv4, TCP, UDP, HTTP
Wi-Fi Module		External Interface	None
Encryption	WEP/TKIP/AES	Wi-Fi Frequency	2.4G – 2.5G
Security	WPA/WPA2	Width	5mm
Length	5mm	Weight	5g
Operating Current	80mA		[11]

TABLE 6: ESP8266 TECHNICAL CHARACTERISTICS

2.3. COMMUNICATION PROTOCOLS IN EMBEDDED SYSTEMS

Embedded systems make use of protocols in order to communicate with other embedded devices. Communication protocols are sets of formal rules that describe the exchange of data between devices, there are two type of protocols:

- Low-level protocols: these protocols define the physical and electrical standards to achieve communication between devices.
- High-level protocols: these protocols define the logical standards to achieve communication between devices, including data formatting, message syntax, message sequence, etc.

Communication protocols may be implemented by the hardware of a device, the software or both. Protocols follow well defined formats for the exchange of messages, every message expects a certain response pre-determined for a situation.

There exist several well-known communication protocols for embedded devices. These protocols will be defined in the sub-sections below.

2.3.1. UART PROTOCOL

Universal Asynchronous Receiver-Transmitter Protocol is a physical communication protocol used for serial asynchronous communication. Serial communication is the process of transmitting data bit by bit, sequentially, over a communication channel.

In order to communicate two devices through the UART protocol, only two wires are required. One connecting the Transmitting pin of the first device to the Receiving pin of the second device, and the second wire connecting the Receiving pin of the first device, to the Transmitting pin of the second device.

In the data transmission of the UART protocol, the data flows from the transmitting pin of the transmitting device, to the receiving pin of the receiving device. At the receiving device, the UART re-assembles the bit structure into bytes for reading. The UART protocol has no clock signal, apart from the initial and final bit of the communication.

Some embedded systems use this protocol as their communication protocol, given that their CPU has been previously programmed to read from the input pins, and transmit data through their output pins.

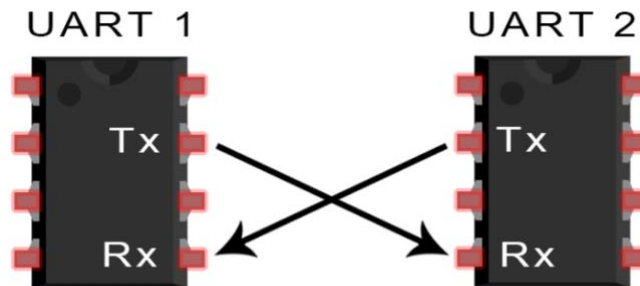


FIGURE 1: UART PROTOCOL COMMUNICATION [12]

2.3.2. SPI PROTOCOL

Serial Peripheral Interface (SPI) protocol is a communication protocol used to transmit data from peripherals to microcontrollers. Embedded devices that communicate using this protocol, normally communicate using a Master-Slave communication, allowing the Master to control multiple Slaves. Normally, the microcontroller of the systems acts as the master, and sends instructions to the slave part of the communication protocol.

This type of communication uses a clock signal to synchronize the data from the master into the slave. The steps followed for the communication are:

1. The Master sends a clock signal
2. The Master switches on the Slave
3. The Master sends the data bit-by-bit
4. If needed, the Slave replies the data bit-by-bit

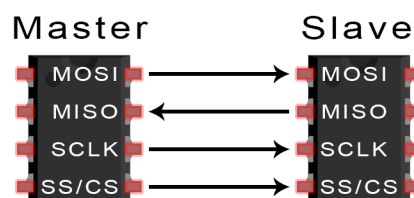


FIGURE 2: SPI COMMUNICATION PROTOCOL [13]

2.3.3. I²C PROTOCOL

Inter-Integrated Circuit (I²C) protocol, is a serial communication protocol most commonly used to transmit data from peripherals to microcontrollers. This protocol combines UART Protocol's main advantage (it needs only two wires for the communication to take place), and the main advantage from the SPI Protocol (it allows for a multiple Slave – single Master communication).

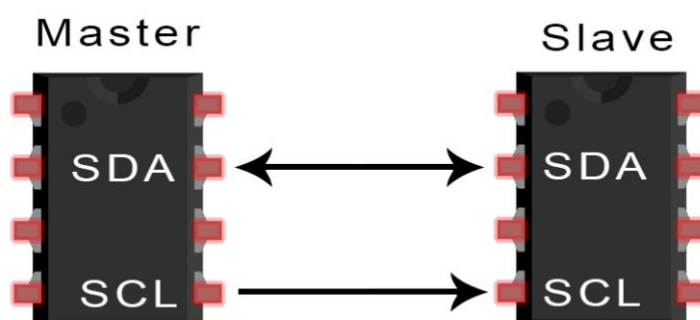


FIGURE 3: I2C COMMUNICATION PROTOCOL [14]

The two wires have two different functions:

- SDA (Serial Data): data transfer bit-by-bit.
- SCL (Serial Clock): clock signal.

This protocol transfers messages as frames of data, these frames are transmitted through the serial bus (SDA). Each message contains:

- Binary address of the slave
- Data frames for the data transmission
- Start and stop conditions
- Read and write bits
- ACK and NACK bits

The main advantage of this communication protocol is the inclusion of ACK and NACK bits, that allow the system to confirm that the data has been transmitted as desired.

2.4. INSTANT MESSAGING APPLICATIONS

Instant messaging applications increased in popularity as Smartphones became more accessible to users. These applications offer reasonable approximation of real-time message transmission between different parties over the Internet.

“Instant messaging has become a powerful and popular communication tool, used by individuals and businesses alike, as an effective means to transfer and relay information to other individuals and/or customers.” [15]

These applications, depending on their complexity, offer users different services:

- Message transmission
- Image/video transmission
- Cloud storage
- Private conversations (between two people)
- Group conversations (two or more people)
- Message encryption
- Acknowledgement of message receipt
- File transfers
- Check user connection

Instant Messaging Applications allow users to communicate with others by knowing their username or contact information (i.e. phone number). If a user wishes to communicate with another, all they have to do is open a new window to begin the communication.

These tools have allowed people all around the world to stay interconnected, and within reach of each other.

Some of the most popular messaging applications are:

- *WhatsApp*: free instant messaging application, connected to a phone number. Uses end-to-end encryption in their messaging protocol. Allows file transfer, image and video transfer, contact transfer, acknowledges message reception, and allows users to visualize the connection of other users.

- *Messenger*: Facebook's free instant messaging application, that connects to a user's phone number and/or Facebook profile. Uses end-to-end encryption in their messaging protocol. Allows file transfer, image and video transfer, contact transfer, acknowledges message reception, and allows users to visualize the connection of other users.
- *Telegram*: free, open-source, cloud-based instant messaging application, that connects to a user's phone number. Uses server-client encryption and peer-to-peer encryption in their messaging protocol. Allows file transfer, image and video transfer, contact transfer, acknowledges message reception, and allows users to visualize the connection of other users. Because the application is cloud-based, users don't have to download the media directly onto their phones to visualize it, and it is available to re-download anytime. One of the main advantages of this applications is the automatic removal of all data if a user has not entered the application in a period of time (by-default that period is 6 months).

3. SYSTEM ANALYSIS

This chapter will be dedicated to performing a detailed analysis of the components of the system to be developed, hence obtaining a detailed specification of the system. This way, the resulting system will then be later submitted to the design phase.

First, the idea the system evolves around will be explained, detailing the main functionality of the system. Second, we will identify the use cases, in order to be able to identify the processes that the user will undergo. Third, we will depict the requirements that the system must follow, both functional and non-functional. Lastly, a traceability matrix will be made, in order to verify that the requirements cover all of the presented use cases.

3.1. DEFINITION OF THE SYSTEM

This section will explain what is wished to be fulfilled by the project. The system must be deployed inside a Server which will receive periodically updates from a motion detector. The Server must provide the users with a clear and simple set of executable functions in order to guarantee the main functionalities that normal intrusion-detection systems in the market implement.

Given that the system is meant to be used for the safety of one's home or office, it has been designed so that each individual user can create and use their own individual system, as such, giving only them the access to the files that may be created.

The figure below will depict the main functioning of the project.

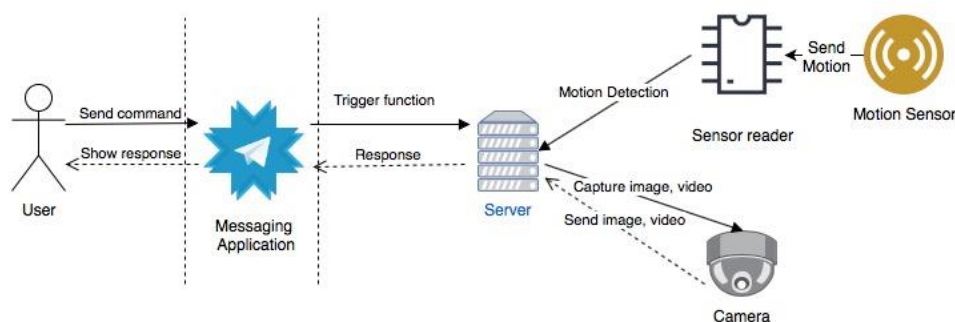


FIGURE 4: APPLICATION DESIGN

3.2. USE CASES

This section will specify all of the different use cases of the system. Use cases are a list of actions or events that normally define the interactions between a user and a system.

To start designing the intrusion system, it is necessary to formally specify the different use cases from the different possible users. The agents that are involved in these scenarios will be the end users of the system, and the application itself.

Once having determined all of the user cases, it is necessary to analyze each of them in a detailed and precise way. In order to do this, a table can be used. All of the tables will use a standard structure, presented below.

ID	UC-XX
Title	
Actor	
Preconditions	
Description	

TABLE 7: USE CASE TEMPLATE

Each of the fields in these tables allow to describe the user cases in detail:

- **ID:** This is represented by UC-XX, where UC stands for Use Case, and XX is a two-digit number, giving the use case a unique identifier. This identifier will be used in a later section to trace the use cases with the system requirements.
- **Actor:** user or set of users that can perform actions in the designed system.
- **Preconditions:** all of the conditions that must be fulfilled in order for the use case to be undergone.
- **Description:** brief description of the functionality that is wished to be achieved in the system.

ID	UC-01
Title	User adds token.
Actor	User.
Preconditions	User has opened the interface.
Description	User adds token to the code.

TABLE 8: USE CASE UC-01

ID	UC-02
Title	User adds system.
Actor	User.
Preconditions	User must have downloaded the code and can edit the code through a text editor.
Description	User introduces system.

TABLE 9: USE CASE UC-02

ID	UC-03
Title	User starts the system.
Actor	User.
Preconditions	The code has the token.
Description	User presses start, and the system sends the user a message with all the instructions available for execution.

TABLE 10: USE CASE UC-03

ID	UC-04
Title	User adds system into a group.
Actor	User.
Preconditions	The system has been initialized, and the token for it has been added into the code.
Description	User adds the system into a chat group as if it were another user.

TABLE 11: USE CASE UC-04

ID	UC-05
Title	User sends command to the bot.
Actor	User.
Preconditions	The system is running in the Server and has already started.
Description	User introduces a command from the possible commands of the system and the execution starts.

TABLE 12: USE CASE UC-05

ID	UC-06
Title	User stops execution.
Actor	User.
Preconditions	The system has been initialized.
Description	User deletes and stops the execution of the application.

TABLE 13: USE CASE UC-06

ID	UC-07
Title	User removes system from group.
Actor	User.
Preconditions	System has been added to a group and is running in the group.
Description	User removes system from the group and the execution of its functions stop.

TABLE 14: USE CASE UC-07

ID	UC-08
Title	User wishes help on how the system works.
Actor	User.
Preconditions	Command help previously programmed.
Description	User sends command help, system returns a list of actions that the user may perform.

TABLE 15: USE CASE UC-08

To ease the comprehension of the use cases presented above, standard UML diagrams will be used to depict them.

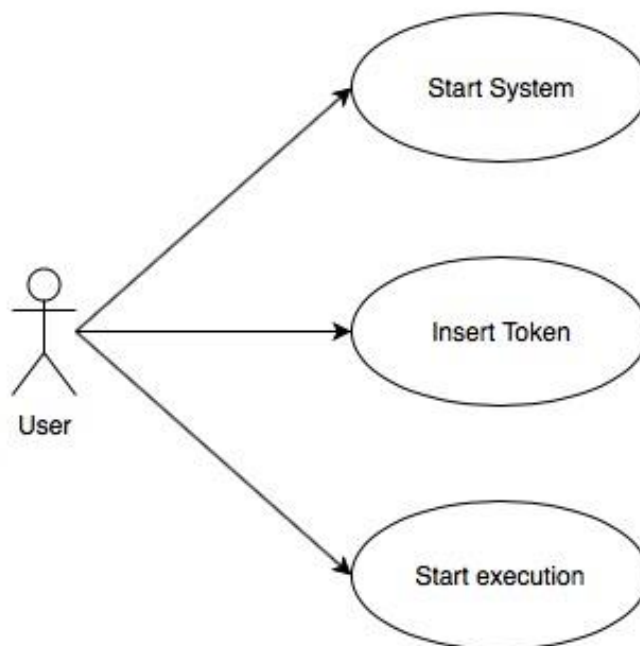


FIGURE 5: USE CASES 1

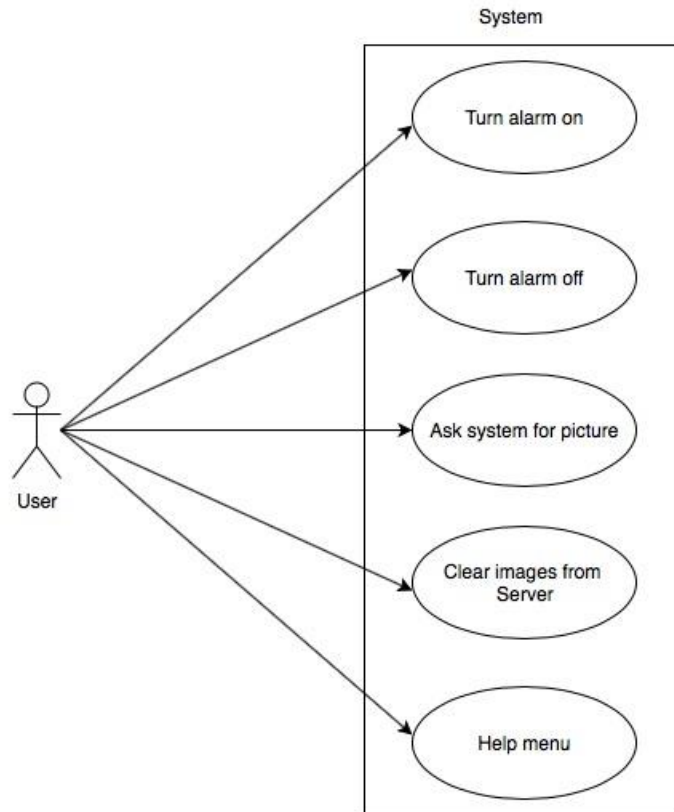


FIGURE 6: USE CASES 2

3.3. SYSTEM REQUIREMENTS

This section will be dedicated to determining all of the system requirements that are necessary for our system to work. System requirements are the configurations that an application must fulfill in order for it to run efficiently. They can be functional requirements, or non-functional requirements.

In order to analyze all of the requirements that the system must fulfill, a table will be used. All of the tables will use a standard structure that is presented below.

ID	FR-XX NFR-XX
Title	
Description	
Priority	
Use-Case(s)	

TABLE 16: REQUIREMENT TEMPLATE

Each of the fields in these tables allow to describe the requirements in detail:

- *ID*: This is represented by FR-XX or NFR-XX, where FR stands for Functional Requirement, NFR stands for Non-Functional Requirement and XX is a two-digit number, giving the system requirement a unique identifier. This identifier will be used in a later section to trace the use cases with the system requirements.
- *Description*: brief description of the functionality that is wished to be achieved in the system.
- *Priority*: priority to fulfill the requirement. There exist three levels of priority:
 - High: necessary for the application to work properly.
 - Medium: extra functionality for the application.
 - Low: not necessary for the proper application workflow.
- *Use-Case(s)*: Use Case or Cases that the requirement fulfills, this will only be indicated in the functional requirements, given that non-functional requirements are those needed for the system to work.

3.3.1. FUNCTIONAL REQUIREMENTS

Functional requirements are the set of requirements that specify the intended functions of a system and its components. It allows to determine the outputs that the application must provide depending on the input it is receiving.

<i>ID</i>	FR-01
<i>Title</i>	Starting the bot.
<i>Description</i>	When the system execution is started by a user, it must send a description of what it does, and it must show the help instructions on what each command does.
<i>Priority</i>	High.
<i>Use-Case(s)</i>	1, 2, 3, 8.

TABLE 17: FUNCTIONAL REQUIREMENT FR-01

ID	FR-02
Title	Adding system into group.
Description	When the system is added to a group, it must send a description of what it does, and it must show the help instructions on what each command does.
Priority	High.
Use-Case(s)	1, 2, 3, 4, 8.

TABLE 18: FUNCTIONAL REQUIREMENT FR-02

ID	FR-03
Title	Visible keyboard with commands.
Description	A keyboard menu with the different commands available must be shown in the chat at all times.
Priority	Medium.
Use-Case(s)	3, 4.

TABLE 19: FUNCTIONAL REQUIREMENT FR-03

ID	FR-04
Title	Help menu.
Description	If a user asks for the help menu, the system must reply with the instructions for its use, detailing the list of commands that are available and what they do.
Priority	High.
Use-Case(s)	3, 4, 6.

TABLE 20: FUNCTIONAL REQUIREMENT FR-04

ID	FR-05
Title	Alarm activation.
Description	The system must have the option to activate the intruder detection system. It must send a message stating that the alarm has been turned on.
Priority	High.
Use-Case(s)	5.

TABLE 21: FUNCTIONAL REQUIREMENT FR-05

ID	FR-06
Title	Alarm activation in group.
Description	Any user must be able to activate the alarm. It must send a message to the group stating that the alarm has been turned on and by who.
Priority	High.
Use-Case(s)	5.

TABLE 22: FUNCTIONAL REQUIREMENT FR-06

ID	FR-07
Title	Alarm deactivation.
Description	The system must have the option to deactivate the intruder detection system. It must send a message stating that the alarm has been turned off.
Priority	High.
Use-Case(s)	5.

TABLE 23: FUNCTIONAL REQUIREMENT FR-07

ID	FR-08
Title	Alarm deactivation in group.
Description	Any user must be able to deactivate the alarm. The system must send a message to the group stating that the alarm has been turned off and by who.
Priority	High.
Use-Case(s)	5.

TABLE 24: FUNCTIONAL REQUIREMENT FR-08

ID	FR-09
Title	Motion notification.
Description	If the intruder detection is on, upon movement detection, the system must send a message stating that it has detected movement.
Priority	High.
Use-Case(s)	5.

TABLE 25: FUNCTIONAL REQUIREMENT FR-09

ID	FR-10
Title	Motion notification with picture.
Description	If the intruder detection is on, upon movement detection, the system must send a picture.
Priority	High.
Use-Case(s)	5.

TABLE 26: FUNCTIONAL REQUIREMENT FR-10

ID	FR-11
Title	Motion notification with video.
Description	If the intruder detection is on, upon movement detection, the system must send a five second video.
Priority	Medium.
Use-Case(s)	5.

TABLE 27: FUNCTIONAL REQUIREMENT FR-11

ID	FR-12
Title	Picture request by user.
Description	If the intruder detection is on, the user may ask for a picture at any time.
Priority	Medium.
Use-Case(s)	5.

TABLE 28: FUNCTIONAL REQUIREMENT FR-12

ID	FR-13
Title	Picture request by user in group.
Description	If the intruder detection is on, a user may ask for a picture at any time.
Priority	Medium.
Use-Case(s)	5.

TABLE 29: FUNCTIONAL REQUIREMENT FR-13

ID	FR-14
Title	Cleaning images from Server.
Description	A user can send a request to remove all the pictures that are stored in the Server until that moment.
Priority	Medium.
Use-Case(s)	5.

TABLE 30: FUNCTIONAL REQUIREMENT FR-14

ID	FR-15
Title	Inexistent command.
Description	If a user sends a command that does not exist, the system must inform the user of such.
Priority	Medium.
Use-Case(s)	5.

TABLE 31: FUNCTIONAL REQUIREMENT FR-15

ID	FR-16
Title	Log.
Description	If any user performs any command on the bot, it must be logged with the action, timestamp, and user.
Priority	Medium.
Use-Case(s)	5.

TABLE 32: FUNCTIONAL REQUIREMENT FR-16

ID	FR-17
Title	Removal of system in group.
Description	If the system is removed from a group, its execution of any command must be stopped immediately.
Priority	High.
Use-Case(s)	7.

TABLE 33: FUNCTIONAL REQUIREMENT FR-17

ID	FR-18
Title	Removal of bot.
Description	If the system is deleted or stopped by the user, its execution of any command must be stopped immediately.
Priority	High.
Use-Case(s)	8.

TABLE 34: FUNCTIONAL REQUIREMENT FR-18

3.3.2. NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are the set of requirements that specify the criteria necessary in order for the application to work correctly in a system.

ID	NFR-01
Title	Operating System.
Description	The operating system used to develop the system must be UNIX based.
Priority	High.

TABLE 35: NON-FUNCTIONAL REQUIREMENT NFR-01

ID	NFR-02
Title	Programming Language.
Description	The programming language for the development of the system must be Python 3.0.
Priority	High.

TABLE 36: NON-FUNCTIONAL REQUIREMENT NFR-02

ID	NFR-03
Title	Application control.
Description	The application must have the option to be controlled through a Smartphone.
Priority	High.

TABLE 37: NON-FUNCTIONAL REQUIREMENT NFR-03

ID	NFR-04
Title	Microcontroller.
Description	To guarantee reading the PIR Sensor in real time, a dedicated microcontroller must be used.
Priority	High.

TABLE 38: NON-FUNCTIONAL REQUIREMENT NFR-04

ID	NFR-05
Title	Microcontroller-server communication.
Description	To guarantee communication between the microcontroller and the server, the communication cables will be duplicated.
Priority	High.

TABLE 39: NON-FUNCTIONAL REQUIREMENT NFR-05

ID	NFR-06
Title	Encrypted communication.
Description	To guarantee that personal information is encrypted, all public communications must use HTTPS protocol.
Priority	High.

TABLE 40: NON-FUNCTIONAL REQUIREMENT NFR-06

ID	NFR-07
Title	System configuration.
Description	The System must have configured the Debian Linux Distribution.
Priority	High.

TABLE 41: NON-FUNCTIONAL REQUIREMENT NFR-07

ID	NFR-08
Title	Device communication.
Description	The two embedded devices must communicate using the serial port.
Priority	High.

TABLE 42: NON-FUNCTIONAL REQUIREMENT NFR-08

ID	NFR-09
Title	Camera configuration.
Description	The system must have a software dedicated to the capture of images
Priority	High.

TABLE 43: NON-FUNCTIONAL REQUIREMENT NFR-09

ID	NFR-10
Title	Video storage.
Description	The system must have a software dedicated to the capture of videos.
Priority	High.

TABLE 44: NON-FUNCTIONAL REQUIREMENT NFR-10

ID	NFR-11
Title	Images storage.
Description	The camera folder for storing the pictures must be created in the Server.
Priority	High.

TABLE 45: NON-FUNCTIONAL REQUIREMENT NFR-11

ID	NFR-12
Title	Internet connection.
Description	In order for the system to work, it is necessary for the Server to have access to the internet.
Priority	High.

TABLE 46: NON-FUNCTIONAL REQUIREMENT NFR-12

ID	NFR-13
Title	PIR Read.
Description	In order to obtain real time responses, the PIR must be read every 100ms.
Priority	High.

TABLE 47: NON-FUNCTIONAL REQUIREMENT NFR-13

ID	NFR-14
Title	System efficiency.
Description	In order for the compete with other systems in the market, the user must be notified within a 3 second range of an intrusion.
Priority	High.

TABLE 48: NON-FUNCTIONAL REQUIREMENT NFR-14

3.4. TRACEABILITY MATRIX

To provide a better understanding on how the relationship of the user cases to the functional requirements works, a traceability matrix is provided. In this matrix, we can state which functional requirements are meant to satisfy which use case. Please note, that only the functional requirements are considered for the traceability matrix, given that non-functional requirements are those needed for the system to work.

		Use cases							
		UC-01	UC-02	UC-03	UC-04	UC-05	UC-06	UC-07	UC-08
Functional Requirements	FR-01	X	X	X					X
	FR-02		X	X	X				X
	FR-03			X	X				
	FR-04			X	X		X		
	FR-05					X			
	FR-06					X			
	FR-07					X			
	FR-08					X			
	FR-09					X			
	FR-10					X			
	FR-11					X			
	FR-12					X			
	FR-13					X			
	FR-14					X			
	FR-15					X			
	FR-16					X			
	FR-17							X	
	FR-18								X

TABLE 49: TRACEABILITY MATRIX

4. SYSTEM DESIGN

Throughout this chapter, the system design to be developed will be explained in detail, this will cover all of the different parts that make up the system. The first section will be dedicated to explaining a detailed overview of the system. After the system architecture will be explained. Following, a flowchart will be represented, which will be used to depict a visual representation of the activities. After, a component diagram will be shown in order to represent the functionalities. Next, sequence diagrams will be portrayed in order to show the relationships and interactions between the different components. Lastly, a class diagram will be shown in order to represent a detailed view of the different components.

4.1. DETAILED DESIGN

For the later implementation of the system, it is necessary to consider a detailed design. For this, the following aspects have been considered:

4.1.1. COMUNICATION PARADIGM

The model of the system to be applied will be the Model View Controller.

- Model: level in charge of controlling the data the application can contain, handles the requests performed by a user.
- View: level in charge of capturing the request from the user and sending the controller the results obtained after said request.
- Controller: level in charge of sending the user results obtained. It will consist of a Client-Server architecture:
 - Server: will receive continuous petitions from the Passive Client and the View.
 - Clients: the passive Client will send continuous information to the server when the Passive Infra-Red (PIR) Sensor has detected motion. The active client will be the View, performing the requests from the user.

4.1.2. CONNECTION TYPE

The system consists of two services, one petition service and another motion detection service. The petition service will be processed by TCP, given that it the order of the packets to be received is important. The motion detection service will be processed by UART (direct wired connection), given that the performance is critical.

4.1.3. CONCURRENCY

The system is required to be composed of a concurrent server, given that it requires to handle petitions from the passive Client and the Model part of the system.

4.1.4. STATE

The server is required to be session-based, given that each different bot contains a unique token, which needs to be stored at each of the service users.

4.1.5. NAMING

Both services (passive client and the controller) will be located on the same microcomputer, sharing the IP. For this, the petition service will use port 8080. The motion detection service will use the serial port 9600.

4.1.6. SECURITY

To avoid security breaches, it is necessary to perform secure connections, and the previous elements will need to be ciphered.

4.1.7. MESSAGE SEQUENCE

The message sequence between the different elements of the systems will be depicted in the figures below.

This first figure represents the message sequence between the Passive Client and the Server.

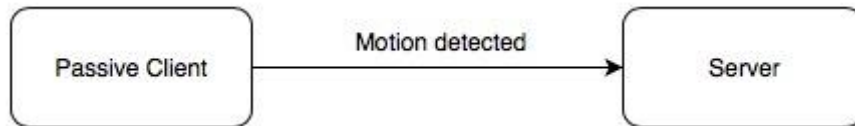


FIGURE 7: PASSIVE CLIENT - SERVER MESSAGE PASSING

This second figure represents the message sequence between the User (active Client) and the Server.

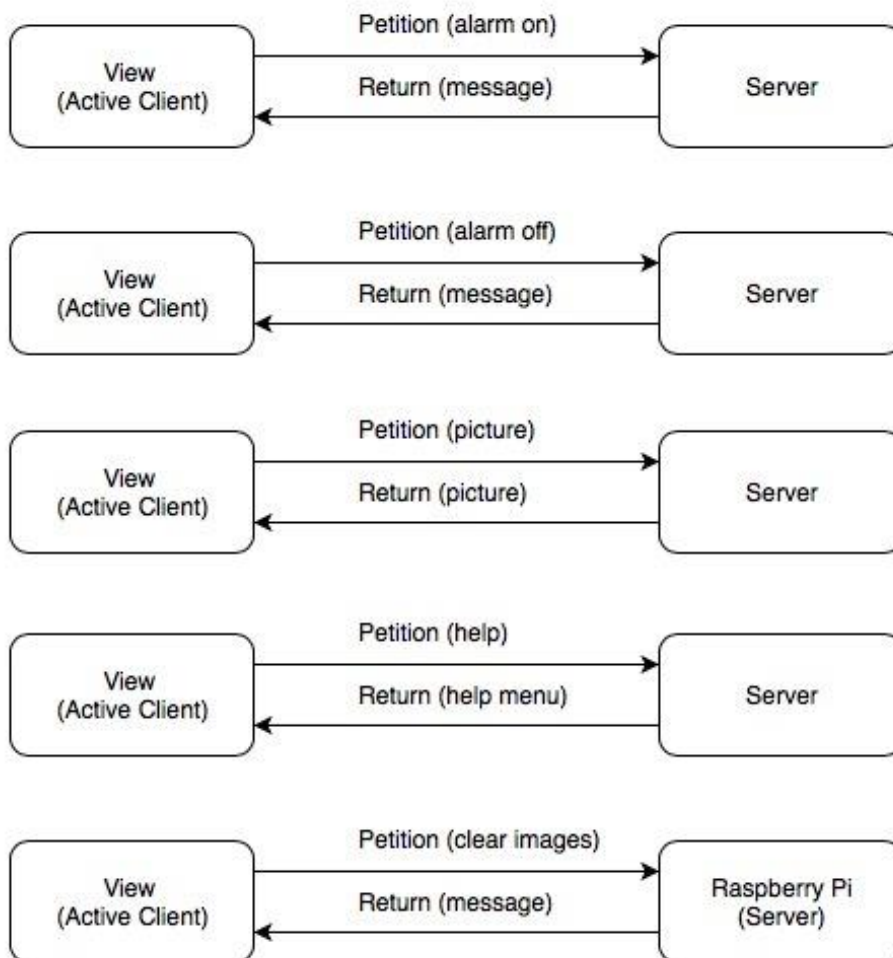


FIGURE 8: ACTIVE CLIENT - SERVER MESSAGE PASSING

4.1.8. MESSAGE FORMAT

In the first message exchange (motion detected), the passive client will send a datagram with the following data:

- Motion detected: 16 bytes (String)

First message size: 16 bytes per message

In the second message exchange (user requests), the active client will send a datagram with the following data:

- Alarm petition: 1 byte (0: alarm off, 1: alarm on)
- Clear petition: 1 byte (1: clear images)
- Picture petition: 1 byte (1: picture request)
- Help petition: 1 byte (1: help)

Second message size: 4 bytes per message

The response datagrams will be:

- Alarm: 16 bytes (String)
- Clear: 16 bytes (String)
- Picture: 600 KBytes (640x480 image)
- Video: 3000KBytes
- Help: 64 bytes (4 strings)

Maximum response message size: 3000 KBytes

4.2. SYSTEM ARCHITECTURE

The architecture followed by this system will be the Model View Controller (MVC) pattern. This pattern is commonly used in applications that have user interfaces, given that it separates the application logic into three different parts, easing functionality, scalability and modularity [16].

This pattern divides the logic pursued by applications into three different levels:

- Model: This level defines the data that the application should contain.
- View: This level defines how the application data should be displayed to the user. This will be performed through a messaging application.
- Controller: This level is situated in between the view and the model. This layer is the one in charge of controlling the different actions that a user can perform; to do so, the actions are collected and interpreted as requests by the model, then the results obtained are gathered and sent to the view, so this in turn will represent the results to the user.

The scheme that the system uses is presented in the figure below.

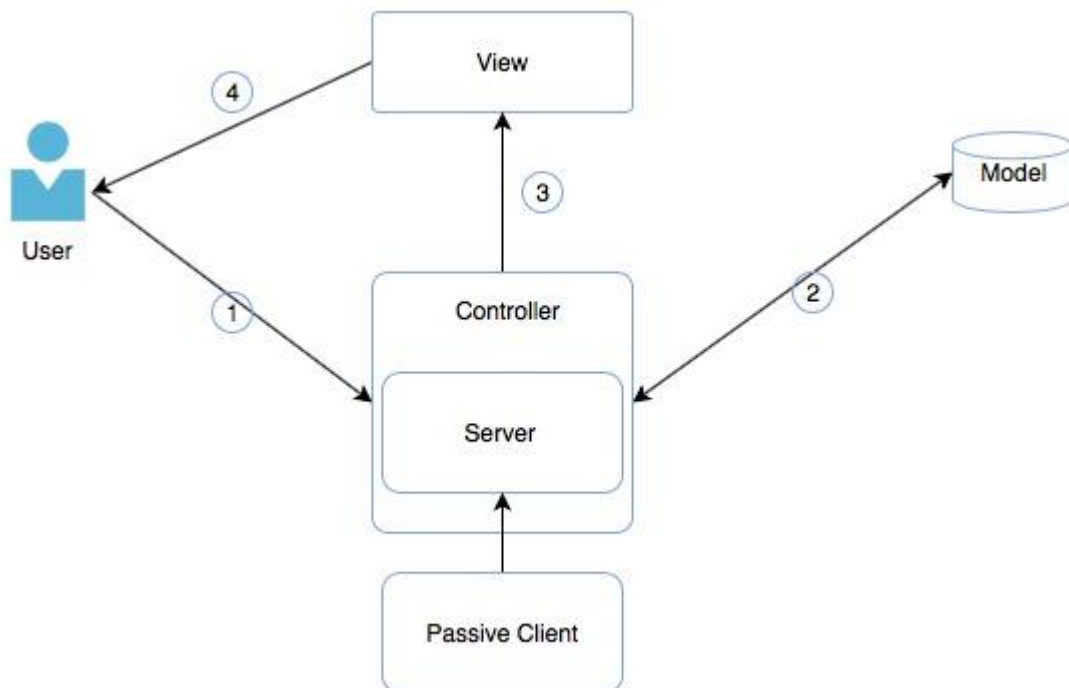


FIGURE 9: MODEL VIEW CONTROLLER (MVC) SCHEME

In this scheme, it is possible to distinguish four steps in order for the user to be able to perform an action:

1. The user will request an action to the controller. In the case of this system, these requests will be performed using a mobile application HTTPS-interface system, which will provide encryption and integrity of the messages sent.
2. The controller invokes the model in order to perform the requests of the user, these requests may be:
 - a. Turn the alarm on or off.
 - b. Request a photo.
 - c. Request help.
 - d. Clear images.

Once the model has processed the request, it will return the results obtained to the controller.

3. The controller processes the data obtained by the model and sends them to the view.
4. The view presents the user with the results obtained according to the request made.

It is also possible to distinguish the continuous information sent from the Passive Client to the Server. The server will continuously process the motion detection data obtained from the Passive client, according to the data required by the Model.

4.3. FLOWCHART

In this section a flowchart of the system to be implemented will be developed. In this flowchart we will represent the functioning and flow of the system. This diagram is not meant for a detailed and explicit explanation of the workings of the system, given that that will be explained in the sections following.

The notation used for the diagram is represented in the table below.





Symbol	Meaning
	Start: indicates the beginning of the flow in the system.
	Decision: decision making, indicates the point in the flow where different possibilities can be taken.
	Activity: this represents an activity performed inside a process.
	Flow line: indicates the direction of the flow of the processes.

TABLE 50: FLOWCHART NOTATION

Following, the flowchart representing the flow of the application is shown:

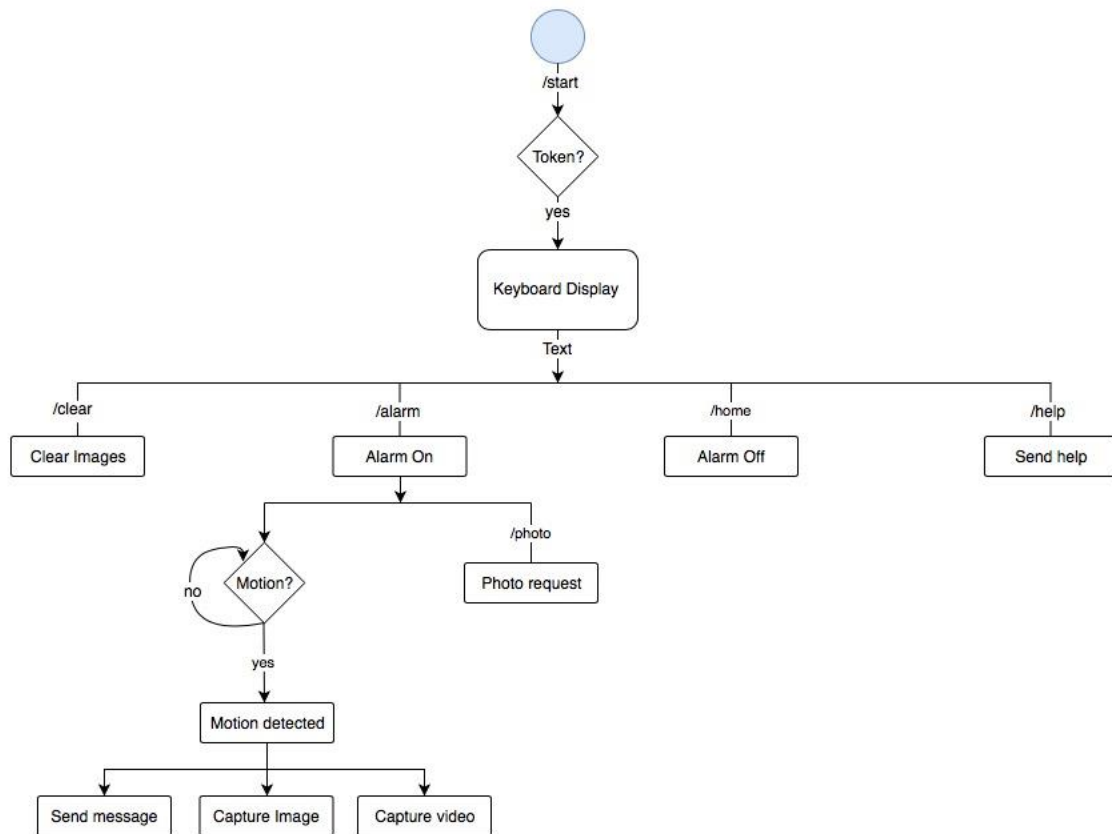


FIGURE 10: FLOWCHART

It is necessary to explain that the flow represented in this diagram does not terminate, the Keyboard is always waiting on user input to perform the actions.

4.4. COMPONENT DIAGRAM

This section will be used to perform a division of the system into components, with the goal to describe and represent graphically all of the components that will make up the system and the structural relationships between them.

The term component covers all of the software elements that make up a system, from files, executables, etc. This will allow the whole system to be represented in the diagram. UML defines a component as either logical or physical [17]. The notation that will be used to represent a component is the one shown in the figure below.



FIGURE 11: COMPONENT REPRESENTATION

Components can have dependencies between them, that is to say, if a component requires a functionality that another offers, it will be represented as a dependency relationship, as shown in the figure below.



FIGURE 12: COMPONENT DEPENDENCY REPRESENTATION

Last, components can be grouped together in subsystems, considering their logic, in order to simplify the later implementation. It is important that subsystems are able to not only contain components, but also other subsystems. The graphical representation of subsystems is depicted in the figure below.

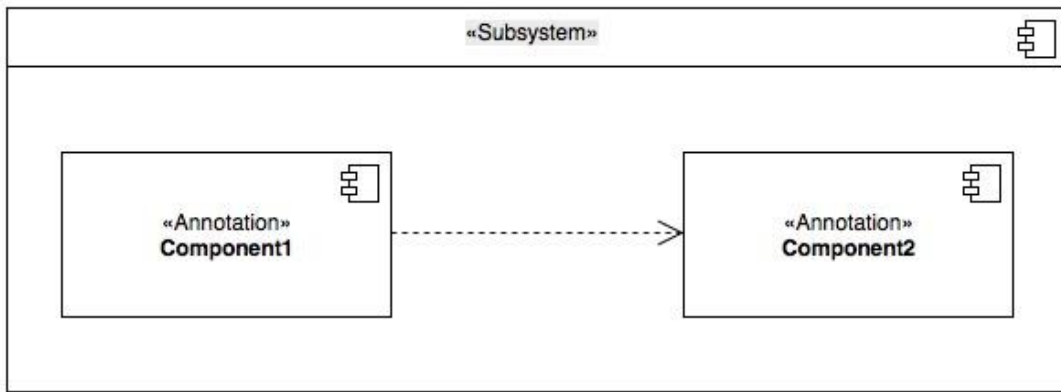


FIGURE 13: SUBSYSTEM REPRESENTATION

Following, it is necessary to differentiate between the different components of our system, for better understanding of the component diagram. In the system, it is possible to find three components:

- User interface: allows the user to interact with the system. This is provided through a mobile application interface.
- Bot: interprets the requests performed by the user.
- System: performs the requests by the user.

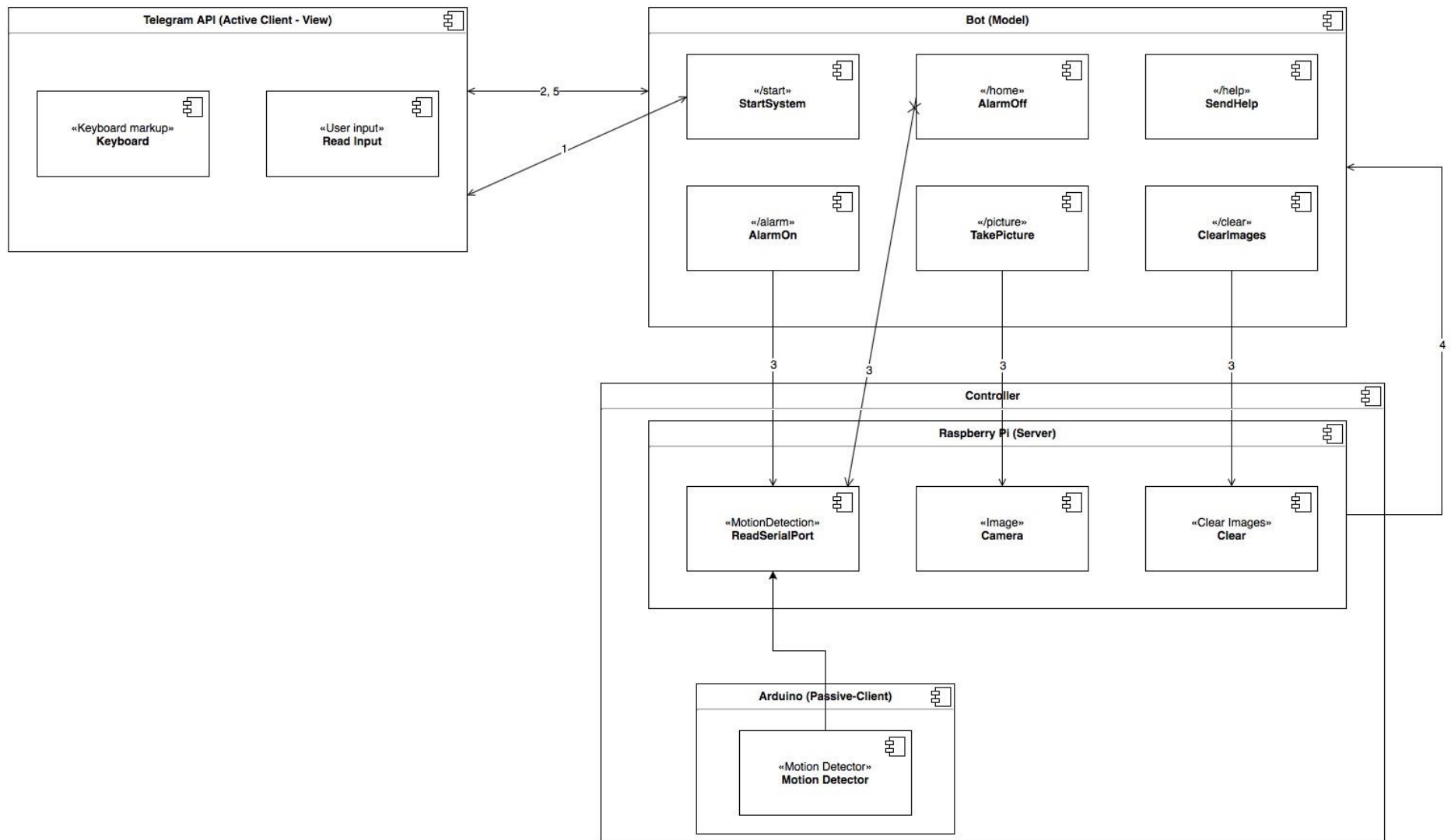


FIGURE 14: COMPONENT DIAGRAM

4.5. SEQUENCE DIAGRAM

This section will show the different sequence diagrams that can be depicted from the component diagram shown in the section above. Sequence diagrams are used to show the interactions arranged in time among the different objects that compose a system.

The parallel vertical lines represent different objects or processes that live simultaneously. The horizontal arrows are used to show the interactions between the objects.

The following sequence diagram represents the creation of a bot by a user.

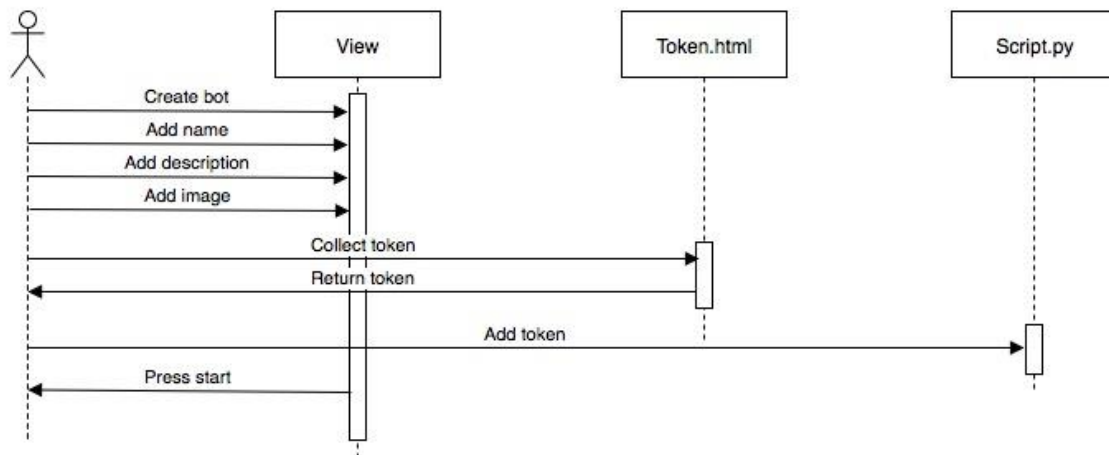


FIGURE 15: SEQUENCE DIAGRAM BOT CREATION

Next, the sequence diagram representing all the different actions that the user may perform inside the bot will be depicted. These actions can be:

- Starting the bot
- Ask bot for help.
- Turning alarm on
 - If motion is detected, user must receive a message, a picture, and a short video.
- Turning alarm off.
- Clear images in the Server.
- Make photograph if alarm is on.

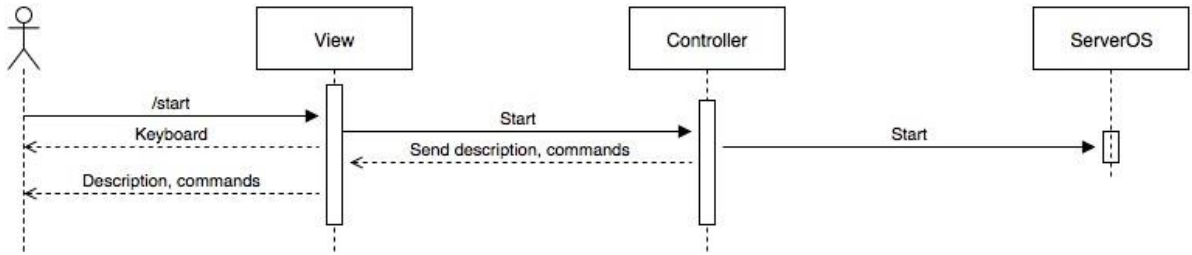


FIGURE 16: START SEQUENCE DIAGRAM

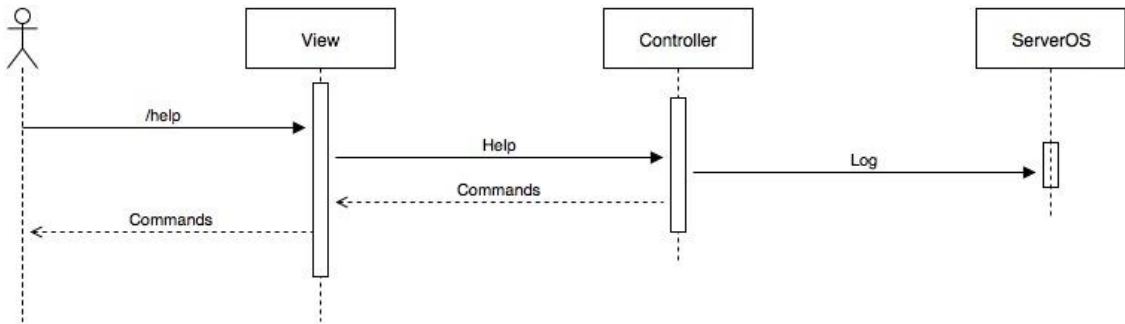


FIGURE 17: HELP SEQUENCE DIAGRAM

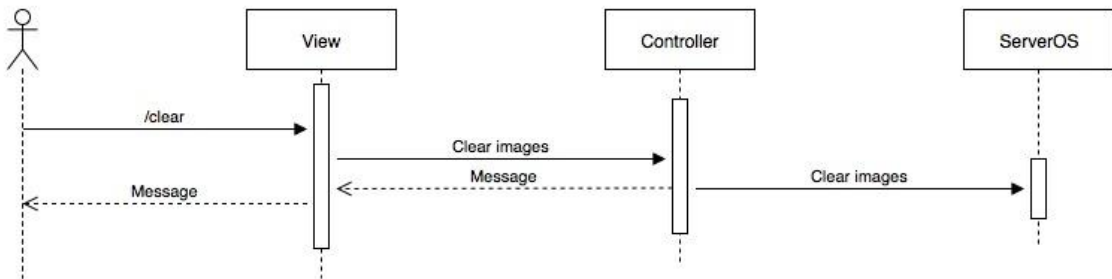


FIGURE 18: CLEAR SEQUENCE DIAGRAM

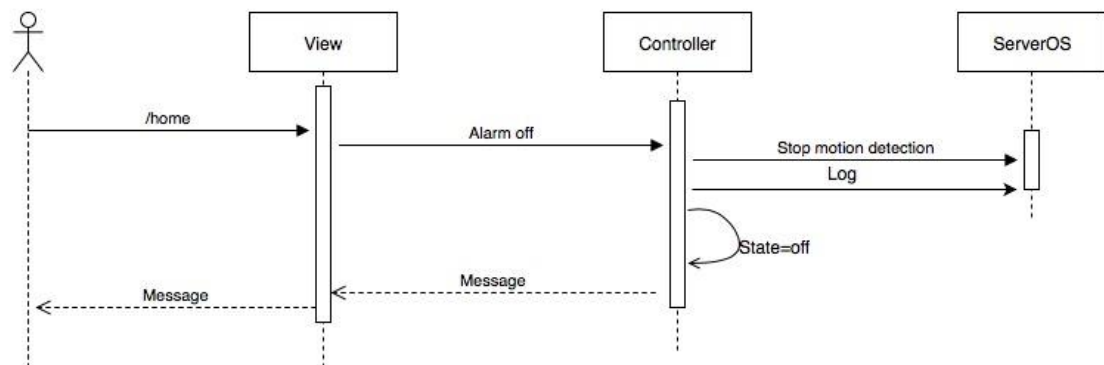


FIGURE 19: ALARM OFF SEQUENCE DIAGRAM

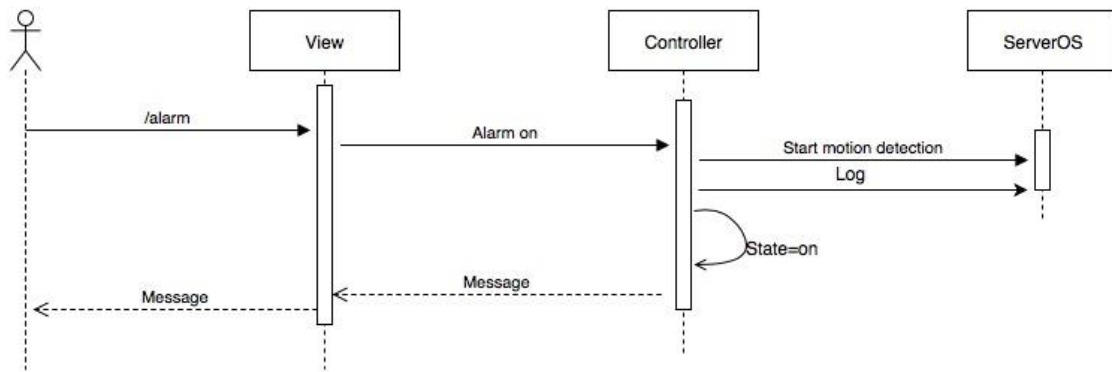


FIGURE 20: ALARM ON SEQUENCE DIAGRAM

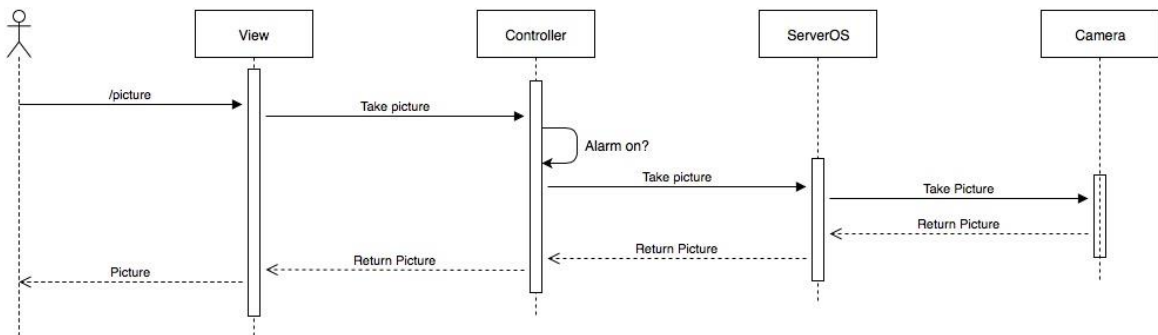


FIGURE 21: PICTURE REQUEST SEQUENCE DIAGRAM

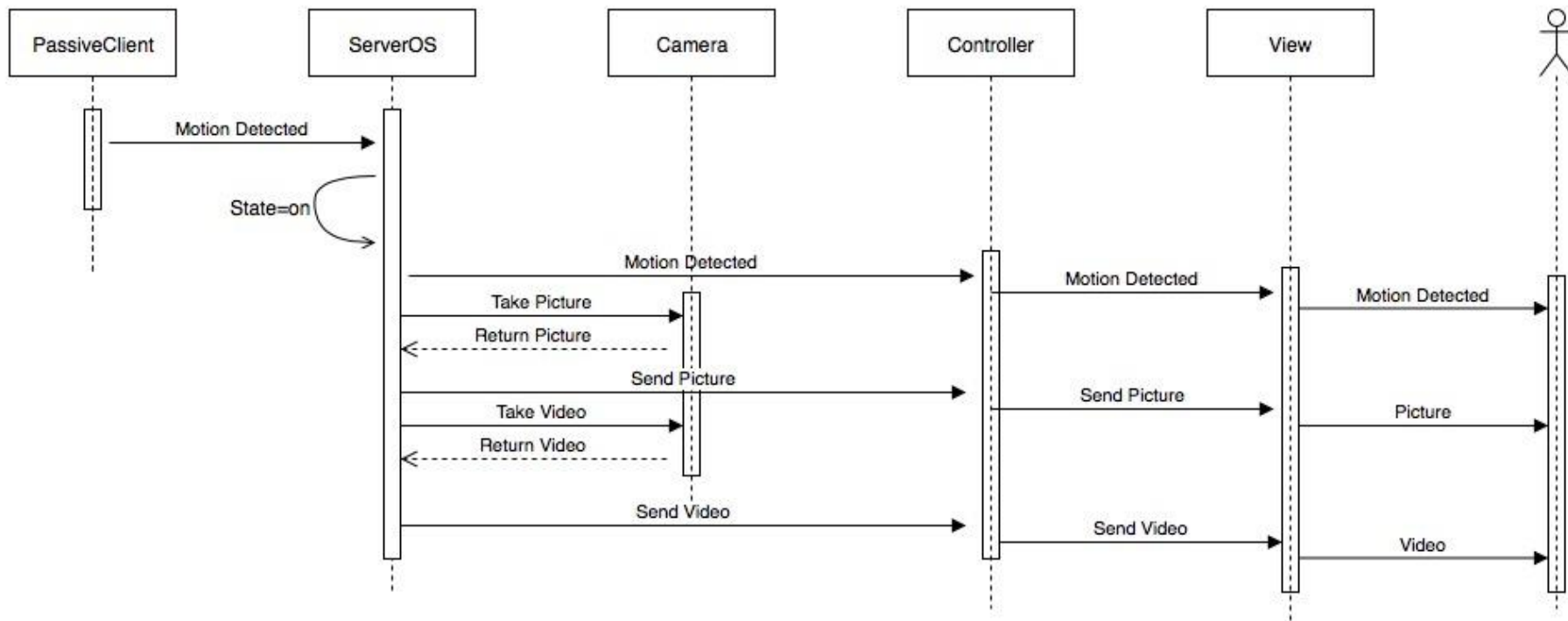


FIGURE 22: MOTION DETECTION SEQUENCE DIAGRAM

4.6. CLASS DIAGRAM

This section will depict a class diagram, which is a schematic view of the system to be implemented. This view includes the different classes, objects and functions of the system, as well as the relationships among them.

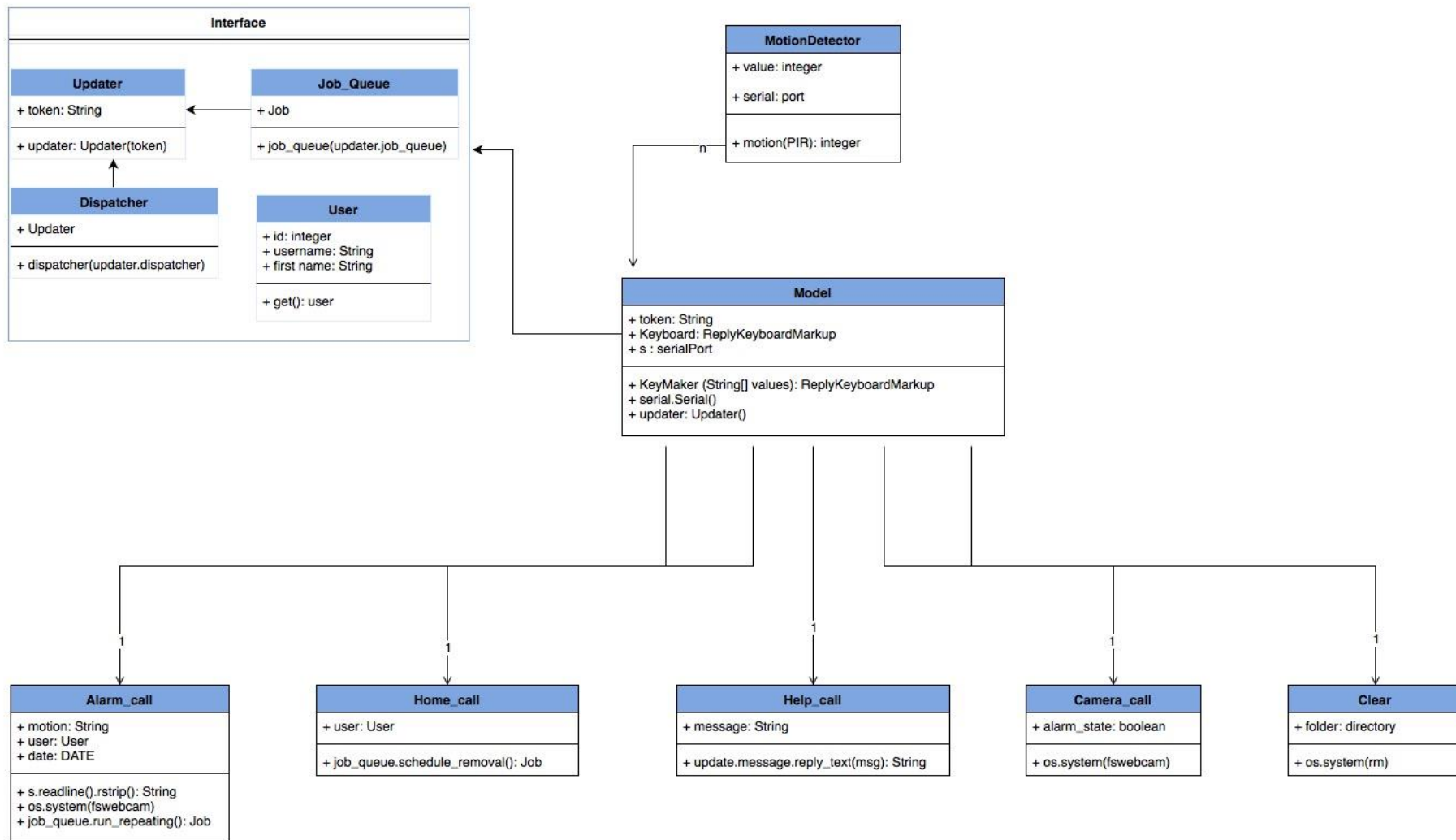


FIGURE 23: CLASS DIAGRAM

5. IMPLEMENTATION

The main objective of this section is to explain the implementation of the system, that was explained in the design section of this document. This will explain the different classes and steps followed in the system for its proper functioning. It will start depicting the necessary components for the system, will be followed by the implementation of the circuit and its specifications, it will continue with the implementation of the classes, and will end with the creation and deployment for the system to work.

5.1. SYSTEM COMPONENTS

In order to develop an advanced intrusion detection system, it is necessary to have certain electronic components, presented in the table below.

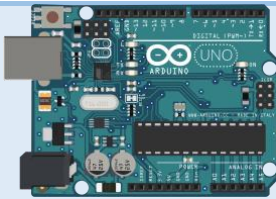

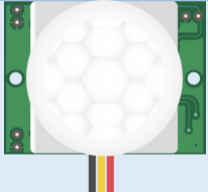

Component	Corresponds	Image
Arduino UNO	Passive Client	
Raspberry Pi 3 Model B (Includes Bluetooth and Wi-Fi modules)	Server	
Passive Infrared Sensor (PIR)	Passive Client motion detector	
USB Camera	Server camera	

TABLE 51: SYSTEM COMPONENTS

5.2. ARDUINO

In order to implement the Passive client part of the system, it was necessary to first design the circuit model. The circuit is made up of an Arduino UNO board, and a Passive Infra-Red (PIR) Sensor. PIR Sensor is a pyroelectric device which measures the infrared levels emitted by the surroundings and detects motion upon variation of said level. It consists of 3 pins, one will be ground, another signal and the last one will be power. The following circuit shows the connection of a PIR Sensor to an Arduino board.

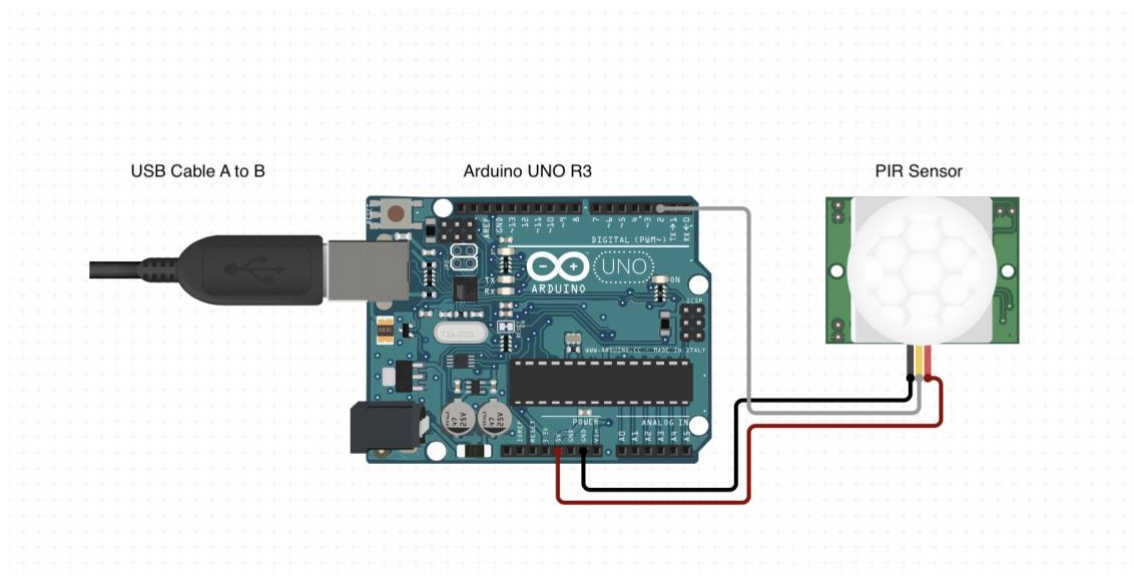


FIGURE 24: ARDUINO CIRCUIT

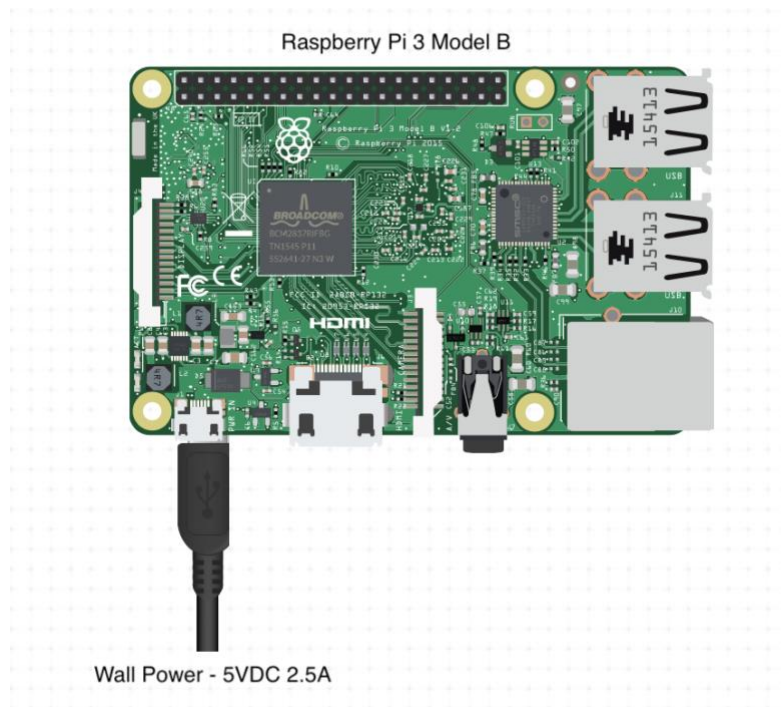
In order for the circuit to work, it was necessary to program it. To do so, we have used the Arduino programming language on the Arduino IDE.

For the code, it is necessary to indicate which will be the input pin (signal), which was chosen to be Pin2. It was then necessary to declare the PIR as input, and the LED in the Arduino board as output (allowing us to know when the sensor had detected a movement) and the serial port to be used.

At the beginning, the state of the PIR sensor, was set to low, hence indicating that no movement had been detected. The code performs an endless loop, checking every second if a movement has been detected, if so, the state of the PIR Sensor is set to high and it will print in the serial port the string "movement detected!", else it will not print anything.

5.3. RASPBERRY PI

In order to implement the Server part of the system, it was necessary to decide which microcontroller was going to be used. A Raspberry Pi 3 Model B (RPi) was chosen, given that it has already built in a Wi-Fi component, and several USB ports for the camera connection and the Arduino connection. Below, it is possible to find the figure of the Raspberry Pi.



In order for the RPi to be able to perform the role of server, it was necessary to configure it. The Debian Linux distribution was installed in the RPi given the many advantages of the operating system. The distribution is open source and it does not need a graphical desktop hence it is capable of running in 600MB.

In order to configure the microcontroller for the later implementation, it is necessary to connect it to an internet source, for such, it is required to open the RPi configuration and refer to Network options, click on a Wi-Fi network, and add the SSID and password. It is also necessary to install the Telegram API package, the camera controllers and vim, which will allow to edit the code from the device. It is also required to connect the device to the internet

5.4. CAMERA CONFIGURATIONS

In order to configure the camera, it is necessary to install two different packages, both packages include option specifications (flags) for capturing image or video as desired by the user:

- *fswebcam*: this package allows the capturing of images from any USB camera in UNIX systems. The flags used are:
 - `-r`: image resolution in bits (640x480)
 - `-no-banner`: remove banner from image
 - `/home/pi/webcam`: directory for storing images
 - `"DATE".jpeg`: image extension named as the timestamp it was taken
- *ffmpeg*: this package allows to capture video from any USB camera in UNIX systems. The flags used are:
 - `-t`: time in seconds for the duration of the video
 - `-f v4l2`: video for Linux API
 - `-framerate`: frames per second
 - `-video_size`: video resolution in bits (640x480)
 - `-i "/home/pi/video"`: directory for storing video
 - `"DATE".mkv`: video extension named as the timestamp it was taken

These packages are required given that the camera used for the intrusion detection system is not the camera module from the Raspberry Pi, but a normal USB camera.

5.5. ARDUINO RASPBERRY PI CONNECTION

The Arduino and the Raspberry Pi are connected through the serial port, following the Universal Asynchronous Receiver-Transmitter (UART) protocol. UART is a hardware device for asynchronous serial communication [18].

Although both devices can follow this protocol through the USB port, it is also possible for them to connect through the specified UART pins. The following diagram depicts the UART protocol through the pins:

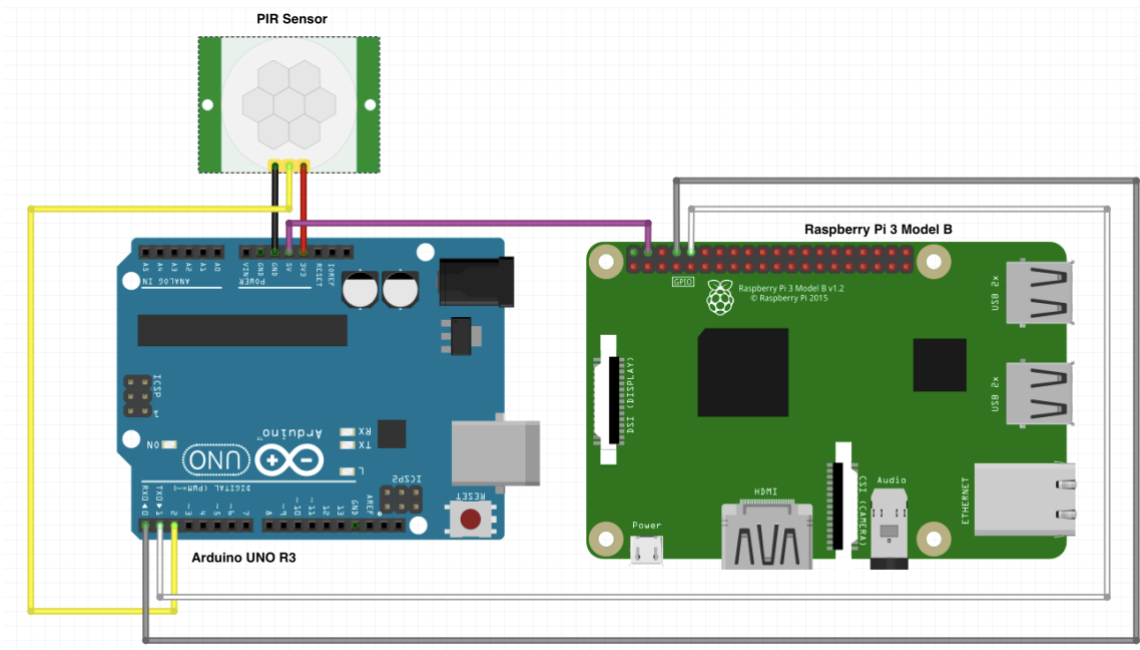


FIGURE 25: ARDUINO-RASPERRY CIRCUIT

As it can be seen in the figure above, both embedded devices must have a wire connecting the voltage input pin, given that they will both share the voltage provided by the wall power to the Raspberry Pi board.

The Pins used for the UART communication are:

- GPIO15 UART0_RXD: Raspberry General-Purpose Input-Output Receive
- GPIO14 UART0_TXD: Raspberry General-Purpose Input-Output Transmit
- D0RX: Arduino Digital Pin 0 Receive
- D1TX: Arduino Digital Pin 1 Transmit

In order for the circuit to work, it is necessary that a transmission pin from one device connects to the receiving pin of the other device, and vice-versa. As such, the connection is established as follows:

- GPIO15 UART0_RXD connected to D1TX
- GPIO14 UART0_TXD connected to D0RX

5.6. SCRIPT IMPLEMENTATION

The final step in the implementation of the project is to develop the code. For this, it was decided to use a Telegram Bot.

Telegram is a cloud-based messaging system. Its main focus relies on security and performance. This application has mobile and desktop applications which synchronize seamlessly across the different platforms. It is an open application, and as such, their Application Programming Interface (API), provides a platform for users to develop their own tools, these tools are called Bots.

A Telegram bot is a special account that does not require a phone number in order for it to work, and as such, can enrich users with extra functionalities. The Telegram API provides the interface to code running in a different machine. It also provides the Telegram MTProto encryption protocol, given that their server will handle all the encryption and communication with their API [19]. Bots communicate with Telegram servers via a HTTPS-interface, which is a simplified version of their API.

The Telegram API provides the possibility to develop bots. The code for the API exists in a Python wrapper, and as such, this code will be built using the Python 3.0 programming language.

A bot interacts with a user, by receiving input commands. These commands are normally pre-programmed to reply with certain messages or events. In order for the interaction to be possible, the API provides several classes [20] [21]:

- **Updater:** this class provides the frontend of the Bot. Its purpose is to receive the requests from the user and send said requests to the dispatcher for execution.
- **Dispatcher:** this class dispatches the requests to the Handler.
- **Handler:** instance of a class responsible for the routing of the requests to the callback functions programmed in the code.
- **Job:** this class allows the periodical execution of tasks.
 - **run_daily:** job that executes once every day
 - **run_one:** job that executes only once
 - **run_repeating:** job that executes periodically as established by code

It is necessary to indicate the functions the bot will require in order to comply with the analysis and design of the project. These functions are:

- **start:** when a user sends the command */start*, a keyboard with the options for the bot must be presented to the user, as well as a brief explanation of the different commands the bot has for possible execution.
- **help:** when a user sends the command */help*, a brief explanation of the different commands the bot has for possible execution will be sent to the user.
- **alarm_on:** when a user sends the command */alarm*, he/she will be presented with a message on the screen, indicating which user has turned on the alarm. The variable *alarm_state* will change its state to true, and system must then read from the serial port, when upon reading “motion detected!” it must start the motion detection call. This function will create several jobs that will repeatedly read the serial port, giving the possibility to inform the user continuously; these jobs will be added to the *job_queue*.
- **motion_detection:** when the alarm call has called for the motion_detection process, the system must inform the users by sending a message that it has detected a movement, it must then call the operating system and capture an image. After the image has been stored, it must call the operating system again to capture a short video. Once the image and the video have been stored, these must be sent to the user.
- **alarm_off:** when a user sends the command */home*, he/she will be presented with a message on the screen, indicating which user has turned off the alarm. The variable *alarm_state* will change its state to false, and system must no longer read from the serial port. This function will remove all jobs from the *job_queue*.
- **clear_images:** when a user sends the command */clear*, the system must call the operating system, and perform the removal of the images and videos from their corresponding folder. Upon the completion, it must return a message to the user indicating that the action has been successfully completed.
- **take_picture:** when a user sends the command */image*, the system must check the state of the *alarm_state* variable, if said state is true, it must call the operating system and capture an image, sending it to the user. If the state of

alarm_state is false, it must send the user a message indicating that the alarm has not been turned on.

- **main:** this function must ensure that all the calls to the bot pass through the handler and dispatcher, in order to provide the proper functioning. This function must also encapsulate the token of the bot in order to provide the required service.

5.7. BOT CREATION

The next step to follow is to create the bot. For this, Telegram provides a very easy step-by-step interface.

1. Look for the Bot “BotFather”, this bot allows the creation and setting modification of Bots.
2. To create a bot, use the command */new_bot*, it is necessary to introduce a name and a username.
3. Once the previous step is performed, a token will be generated. A token is a string required to authorize the bot communication with the API.
4. It is possible to add a description to the Bot by introducing the command */setdescription*, this will show users a description of your Bot.
5. It is possible to add about information to your Bot by introducing the command */setabouttext*.
6. It is possible to add an image by introducing the command */setuserpic*.
7. It is possible to set the commands that your bot supports by introducing the command */setcommands*, this will store the commands along with a brief description of what they do.

5.8. RUNNING THE SCRIPT

In order for the system to run as desired, it is necessary for the script to be running from the Raspberry Pi. To achieve this, we need to upload the script into our machine, adding the token obtained from our bot.

To upload the script, both our Raspberry Pi and the device from where we need to upload the code must be connected to the same Wi-Fi network. From there, we can transfer the script to our Server by introducing the following command through the terminal:

```
scp script.py pi@192.XXX.XX.X:/home/pi
```

This command ensures a secure copy (scp) of the script, followed by the name of our Raspberry Pi (pi) at the IP address where the machine is located. The last statement of the command is the directory where one wishes to store the code.

In order to run the code, we need to connect to our Raspberry Pi, for this, we must input the following command through our terminal:

```
ssh pi@192.168.43.106
```

Now, to run the code, we type the following command:

```
python script.py
```

Once we have the script running in our Raspberry, the Bot will be fully functional.

6. EVALUATION

This section's purpose is to evaluate and determine the correct workings of the system developed. In order to perform the evaluation a series of tests have been defined to ensure all of the system requirements are met.

Once having defined the test, they have been portrayed in tables, that follow the structure presented below:

ID	T-XX
Title	
Description	
Result	
Pass	

TABLE 52: TEST TEMPLATE

Each of the fields in these tables allow to describe the tests performed in detail:

- **ID**: This is represented by T-XX where XX is a two-digit number, giving the test a unique identifier. This identifier will be used in a later section to trace the tests to the system requirements.
- **Description**: brief description of the purpose of the test that is wished to be achieved.
- **Result**: used to indicate the result of the test (if it was successful) and any other information necessary.
- **Pass**: used to indicate if the result of the test is successful or not

Software testing requires different levels of testing, given that these tests can be performed during the different stages of the software development [22]. There are three levels of software testing:

- Unit testing: ID UT-XX
- Integration testing: ID IT-XX
- System testing: ID ST-XX

6.1. UNIT TESTING

This section will explain the different unit tests performed in the system. Unit tests are a level of software testing where the individual units of software are tested. In the case of the system, the tests will ensure that the motion detection, the image capture, clearing the image folder, and changing the state of the alarm are working.

ID	UT-01
Title	Motion Detection Arduino
Description	With only the Arduino in the circuit, ensure that the PIR sensor is capturing movement, and printing the correct statement ("motion detected!") through the serial port.
Result	Statement printed in the serial port.
Pass	Successful.

TABLE 53: UNIT TEST UT-01

ID	UT-02
Title	Image capture.
Description	Ensure that the Raspberry Pi is capturing images and naming them according to the desired specification and storing them in the correct folder.
Result	Correct name and folder.
Pass	Successful.

TABLE 54: UNIT TEST UT-02

ID	UT-03
Title	Clear images
Description	Ensure that the Raspberry Pi clears all the data from the folder for the motion detection images but does not remove the folder.
Result	Image deletion correct.
Pass	Successful.

TABLE 55: UNIT TEST UT-03

ID	UT-04
Title	Change alarm state.
Description	Ensure that when turning the alarm on or off the system sends a notification indicating that the alarm has changed its state.
Result	Notification received.
Pass	Successful.

TABLE 56: UNIT TEST UT-04

6.2. INTEGRATION TESTING

This section will provide different integration tests. Integration tests are used as a process to ensure that individual units, when combined, still work properly. The purpose of this level of testing, is to expose faults in the interaction.

ID	IT-01
Title	Arduino – Raspberry Pi connection.
Description	When connecting the Arduino to the Raspberry Pi, the RPi must only read from the serial port, when the alarm has been switched on.
Result	Arduino keeps detecting motion when the alarm is off, but the system does not send notifications to the user.
Pass	Successful.

TABLE 57: INTEGRATION TEST IT-01

ID	IT-02
Title	Camera connection – Picture request.
Description	When connecting the camera, the system is capable of capturing images on demand, by user request if the alarm state is currently on.
Result	Camera captures images and stores them using the correct naming and folder.
Pass	Successful.

TABLE 58: INTEGRATION TEST IT-02

ID	IT-03
Title	Camera connection – Motion detected.
Description	Upon motion detection, the system calls the camera to capture an image of the intruder.
Result	Camera captures images and stores them using the correct naming and folder.
Pass	Successful.

TABLE 59: INTEGRATION TEST IT-03

6.3. SYSTEM TESTING

System testing is the level of software testing where a complete, integrated system is tested. The purpose is to ensure the compliance with the system requirements.

ID	ST-01
Title	System start.
Description	Upon starting the bot, it must send description and help menu. Presenting the keyboard.
Result	Keyboard shown, message received.
Pass	Successful.

TABLE 60: SYSTEM TEST ST-01

ID	ST-02
Title	Alarm activation.
Description	“Alarm activated” is received after <code>/alarm</code> is sent. In a group, it also sends the username of who has activated the alarm.
Result	Message and user received.
Pass	Successful.

TABLE 61: SYSTEM TEST ST-02

ID	ST-03
Title	Motion detection.
Description	If alarm is activated, and a motion is detected by the system, a message must be sent stating a motion has been detected.
Result	Message received.
Pass	Successful.

TABLE 62: SYSTEM TEST ST-03

ID	ST-04
Title	Motion detection image.
Description	If alarm is activated, and a motion is detected by the system, an image must be sent triggered by the motion detection.
Result	Image received.
Pass	Successful.

TABLE 63: SYSTEM TEST ST-04

ID	ST-05
Title	Motion detection video.
Description	If alarm is activated, and a motion is detected by the system, a short video must be sent triggered by the motion detection.
Result	Video received.
Pass	Successful.

TABLE 64: SYSTEM TEST ST-05

ID	ST-06
Title	Alarm deactivation.
Description	“Alarm deactivated” is received after <i>/home</i> is sent. In a group, it also sends the username of who has deactivated the alarm
Result	Message and user received.
Pass	Successful.

TABLE 65: SYSTEM TEST ST-06

ID	ST-07
Title	Motion detection alarm deactivated.
Description	If a motion is detected, with the alarm deactivated, no message or image or video must be sent.
Result	Nothing received.
Pass	Successful.

TABLE 66: SYSTEM TEST ST-07

ID	ST-08
Title	Image request.
Description	If an image has been requested and the alarm is activated, the system must take and send picture.
Result	Alarm activated, picture received. Alarm deactivated, picture not received.
Pass	Successful.

TABLE 67: SYSTEM TEST ST-08

ID	ST-09
Title	Image deletion.
Description	If there are any pictures in the folder destined for the images, all of the contents of the folder must be erased.
Result	Folder empty.
Pass	Successful.

TABLE 68: SYSTEM TEST ST-09

ID	ST-10
Title	Unavailable command.
Description	If a user types in a command that is not available, the system must send back a message stating that said command is unavailable.
Result	Message received.
Pass	Successful.

TABLE 69: SYSTEM TEST ST-10

ID	ST-11
Title	System stop.
Description	Upon stopping the bot or removing it from a group the system must stop.
Result	System stop.
Pass	Successful.

TABLE 70: SYSTEM TEST ST-11

6.4. TEST TRACEABILITY MATRIX

In order to be able to map the fulfillment of the requirements according to the tests developed, we use a traceability matrix.

		System Tests															
		ST-01	ST-02	ST-03	ST-04	ST-05	ST-06	ST-07	ST-08	ST-09	ST-10	ST-11					
Functional Requirements	FR-01	X															
	FR-02	X															
	FR-03	X															
	FR-04	X															
	FR-05		X														
	FR-06		X														
	FR-07						X										
	FR-08						X										
	FR-09			X				X									
	FR-10				X			X									
	FR-11					X		X									
	FR-12								X								
	FR-13								X								
	FR-14																
	FR-15									X							
	FR-16										X						
	FR-17															X	
	FR-18															X	

TABLE 71: TEST-REQUIREMENTS TRACEABILITY MATRIX

7. MANAGEMENT

This section will portray all the information related with the planning of this project, including the budget, which will be broken down into the different expenses.

In order to explain the organization followed by this project, the first section will explain the different tasks that had to be performed, and the time that was dedicated to each of them.

The second section will be dedicated to depicting the budget of the project, the costs to be expected for each of its parts, and the total amount that would be required to fulfill the project.

7.1. PLANNING

The planning of this project was governed by the delivery dates that have been established by the university for the academic year. In order to complete the project in the corresponding time, it was divided into several milestones, assigning each of these a corresponding time frame for completion.

In the FIGURE 26 a Gantt Chart is depicted, presenting the different milestones that the project has been divided into, with their start dates, duration and end dates. This chart allows for a visual representation of how the different phases were performed.

The first task was to prioritize and schedule how the different milestones were to be completed. Once they were represented in the Gantt chart, the completion of the milestones was the next step, along with the documentation of the project, which was done during its entire duration. This task took five days to complete, including the delivery of the different components, and the planning of the project. This was done from the 15th of May until the 20th of May.

The first milestone was the study of State of the Art, this was the study of the different existing projects and the technology of the field of study. This milestone took one month, from the 21st of May, until the 21st of June.

The second milestone was the Analysis of the System, which includes the technologies that were to be used in the project, including the system Architecture; as well as the

different use cases that the application ought to be available to do, and the functional and non-functional requirements of the System. This milestone took 20 days to complete, from the 22nd of June until the 12th of July.

The third milestone was the Design of the System, how the different classes were going to be connected, including the depiction of a Class Diagram. This took another 20 days, from the 15th of July until the 4th of August.

The fourth milestone was the Implementation and Testing of the application. This took about one month, the first part was the coding of the system, and the later was testing that the system met all the specified requirements. This step took 25 days, from the 5th of August, until the 31st.

Once this was done, the fifth milestone was the evaluation of the project. This took 10 days, from the 1st of September until the 10th.

The final milestone was determining the impact of the application in the Social-Economic environment, and the legal aspects that had to be considered, as well as the budget for this project, which is depicted in the next sub-section of this chapter. This took another 10 days, from the 11th of September until the 21st.

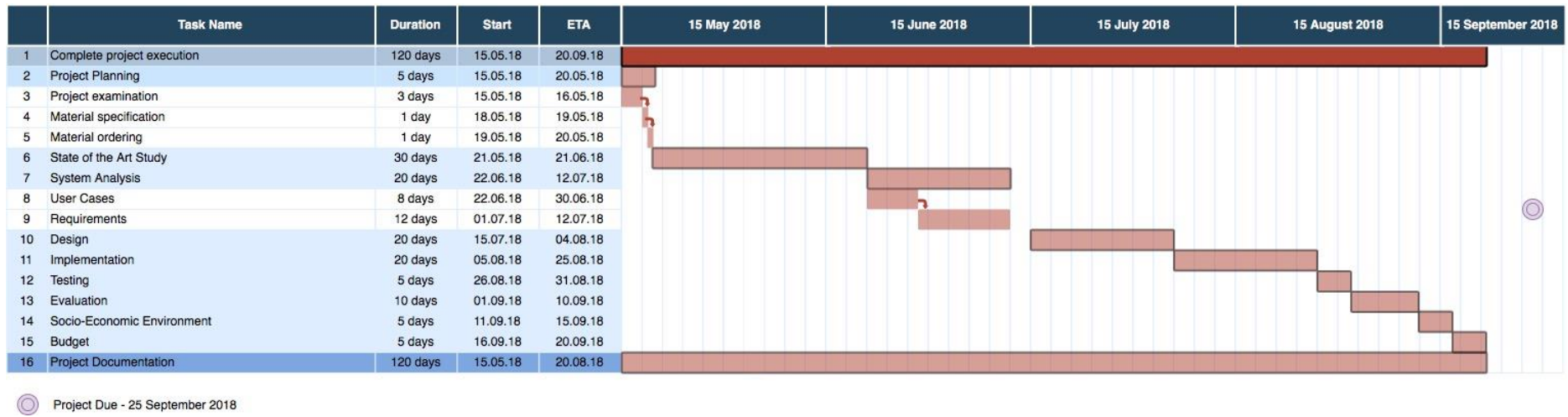


FIGURE 26: GANTT CHART

7.2. BUDGET

This section is dedicated to explaining in detail the costs related to undergoing the project. The cost will be divided into two main categories: direct costs and indirect costs.

7.2.1. DIRECT COSTS

These are the costs relating to the personnel involved in the development of the project, along with the cost of the materials used.

7.2.1.1. PERSONNEL COSTS

These costs refer to all the costs associated to the people that would be involved in the development of the project. Even though the project has been developed by two people, it is necessary to distinguish between the different roles, given that the salary between them differs in amount.

- **Project Manager:** his function is to supervise the project. The tutor of the thesis is the one performing this role.
- **Analyst:** his function is to determine the system requirements and use cases given by the client.
- **Designer:** his function is to determine the design of the system and the class diagrams based on the information provided by the designer.
- **Developer:** the function is to code all the specified information given by the designer.
- **Tester:** once the code is finish, he is the person to ensure that the code works as stated in the requirements and the use cases.

To estimate the salaries that each of the roles involved would receive, the guide “GUIA HAYS 2017 INFORME SECTORES Y SALARIOS”, presented in the official page [23], was used.

Role in the Project	Estimated Hours	Cost per hour (€/hour)	Total cost (€)
Project Manager	90	33.85 €	3,046.50 €
Analyst	150	16.34 €	2,451.00 €
Designer	50	19.95 €	997.50 €
Developer	100	12.50 €	1,250.00 €
Tester	50	13.46 €	673.00 €
Total			8,418.00 €¹

TABLE 72: HUMAN RESOURCES BUDGET

7.2.1.2. MATERIAL COSTS

These costs refer to all the materials used to undergo this project. On them, the amortization of the materials needs to be calculated, to do this, the following formula is used:

$$\text{Imputable cost} = \frac{\text{Months of use}}{\text{Lifespan of the material}} * \text{Unitary cost of the material}$$

The project has been developed over a 4-month period.

Concept	Unitary Cost (€)	Lifespan (months)	Imputable Cost (€)
MacBook Pro (15' retina)	2,799 €	60	186.6 €
Raspberry Pi 3B+	49.99 €	180	1.11 €
Arduino Uno	20.95 €	240	0.35 €
PIR Sensor	1.29 €	24	0.21 €
Total			188.27 €²

TABLE 73: MATERIAL COST BUDGET

¹ Prices Obtained from HAYS 2017

² Prices Obtained September 2018

7.2.2. INDIRECT COSTS

Indirect costs are costs not related to production and cannot be accounted as object or personnel. For this project they have been decided to be fixed as a 20% of the direct costs, in order to cover electricity, Internet connection and any other consumables.

Total Cost	
(€)	
Personnel	8,418.00 €
Material	188.27 €
Indirect Costs (20%)	1,721.25 €
Total	10,327.52 €

TABLE 74: INDIRECT COSTS BUDGET

7.3.2. RISK COSTS

To obtain the risk, one more cost needs to be added, which consists of a 10% of the budget including the indirect costs. Tallying the cost of the project to an amount of 11,360.27€.

7.3.3. TOTAL COSTS

To obtain the final budget, one more cost needs to be added, the benefit, which consists of a 120% of the budget including the indirect costs and risk. Tallying the total cost of the project to an amount of 24,992.60€.

8. LEGAL AND SOCIO-ECONOMIC ENVIRONMENT

Throughout this section, the different, legal, social and economic environments related to the development and non-commercialization of the project will be explained.

8.1. OPEN-SOURCING THE PROJECT

Since the idea for this project originated, the author's intentions were always to develop a good base for an open-source project. This was an idea based on the fact that this, would not only provide people with an economically cheaper option to protect their homes, but also the open-source community could help it grow and develop the project into something bigger.

However, there are several options that have to be considered in order to ensure that this project would not benefit the few, but the many.

8.2. LEGAL ASPECTS

Over this section, the different legal aspects governing the system are going to be explained and discussed, in order for the system to comply with the current legislation.

There are two main laws that affect this project, one is the General Data Protection Regulation law (GDPR) [24], and the Ley 5/2014 de Seguridad Privada [25] [26].

8.2.1. LEY DE SEGURIDAD PRIVADA

The purpose of Ley de Seguridad Privada is to regulate the performance and provision by private, physical or legal individuals, of private security activities and services, for the protection of persons and goods. It also regulates private investigations that are carried out by or on. All these activities are considered as complementary and subordinate to public safety.

This law specifies all the characteristics and regulations that alarm systems must follow inside the Orden INT/316/2011 [27], published in the BOE on the 18th of February of

2011. These regulations cover both, the physical characteristics of the alarm systems, and establish a degree of the security of the systems.

- Degree 1, low risk, alarms based on an acoustic signal, without connection to an alarm receiving center.
- Degree 2, medium-low risk, destined to homes or small offices, that wish to connect to an alarm receiving center.
- Degree 3, medium-high risk, destined to establishments obliged to have an alarm system by law with connection to an alarm receiving center.
- Degree 4, high risk, reserved for establishments with critical infrastructures, obliged to have an alarm system by law, with connection to an alarm receiving center.

Taking into consideration these degrees of risk in the security systems, established by the Spanish law, we can interpret that the Intrusion System that has been designed in this project does not qualify in any of the levels. Our system neither has an acoustic signal coming from it when a motion has been detected, nor we wish to connect to an alarm receiving center, given that we wish to leave that to the user on how to act.

Article 7 of Ley de Seguridad Privada, also speaks about the self-protection proceedings, and excludes them from being subject to the law. These proceedings are understood as the set of precautions or procedures that may be adopted or executed directly by the interested parties directly for the protection of their personal environment, and whose practice or application does not entail any consideration or suppose any type of private security service provided to third parties [25].

Article 42 of Ley de Seguridad Privada speaks directly about the video-vigilance systems, defines them as the exercise of surveillance through camera systems or video cameras, fixed or mobile, capable of capturing and recording images and sounds, including any technical means or system that allows the same treatments as these [26].

With all this into account, we can understand that our system would hence not be under the direct legislation of the Ley de Seguridad Privada, given that none of the articles defined inside the law have a direct connection by neither the materials nor the functionality that is provided by this project.

8.2.2. GENERAL DATA PROTECTION REGULATION

The General Data Protection Regulation is a European regulation, approved on the 25th of May of 2016, and implanted on the 25th of May of 2018. This regulation in the European Union (EU) law was established with regard to the processing and free movement of personal data of all the individuals within the EU. In Spain, this law replaced the Ley Orgánica de Protección de Datos.

GDPR protects and establishes rules for all personal data of the European citizens. It declared personal data as anything that can reveal a citizen's identity, directly or indirectly [24]. This data can be sub-divided into three different categories:

- Low risk: anonymous data, whose consent to the processing has been explicitly given.
- Medium risk: personal data regarding identification, such as name, date of birth, E-Mail address, etc.
- High risk: any data regarding a citizen's ethnicity, religion, economic income, health, sexual orientation, etc. In order to process this kind of data one must perform a Data Privacy Impact Assessment and apply safeguards to mitigate the risk.

This obliges all companies in Europe to inform their users as to what data the company is taking, and every process that that data collected may be put through. Companies are also compelled to inform their users of transparent information and communication.

This regulation analyzes all the risks related to the rights and freedom of the users in regard to their personal data; and determines that the massive processing of data, even if that data is irrelevant to the users, increases the risk of being personally identified.

Users have the right to rectify the data given, as well as the right to erasure ("right to be forgotten"). Users also have the right to be informed on whether an infraction with their data has been committed.

When it comes to intruder detection systems, the data that is strictly under the GDPR is the data collected on the clients of your system.

As the system designed by this project was done so with the idea that every person who wished to implant it in their home or office, was to have their own data storing, without access to any others, there is no personal data collected from the users.

Given that GDPR does not allow data to be transmitted to third parties without the users consent, if an infraction was committed in our home or office, the pictures taken could not be uploaded anywhere else except our machine, and the original files must be kept and handed in to the corresponding law officers as soon as possible.

8.3. INTELLECTUAL PROPERTY

To make the project open-source [28], an in-depth research of the available distribution licenses for this type of project is required.

8.3.1. GPL – Compatible Free Software License

The GNU General Public License is a free, copyleft license for software and any other kinds of works. This is used to guarantee the freedom to share and modify the versions of the software, making sure it remains free for all users [29]. If this license is applied, the following conditions must be met:

- In case of distribution and/or modification, it must be under the terms of the license, as published in the specified version or any later version.
- The Software provided “as is” without warranty of any kind.
- Authors or holders cannot be liable for any liabilities in its use.

The GNU General Public License does not allow incorporating the program into proprietary programs unless using the GNU Lesser General Public License.

8.3.2. MIT License

This License grants permission, free of charge, to any person that obtains a copy of the Software or the documentation files. This allows to deal the Software without restrictions, including the right of use, copy, modify, merge, publish, distribute, sublicense and/or sell [30]. If this license is applied, the following must be met:

- The same copyright license must be used in all the copies or portions.
- The Software provided “as is” without warranty of any kind.
- Authors or holders cannot be liable for any liabilities in its use.

8.3.3. Apache

This License grants permission and freedom to use, distribute and modify the Software, providing it is done under the same License agreement [31]. If this license is applied, the following conditions must be met:

- Unless stated otherwise by the original author, contributions shall be made under the same terms and conditions of the license.
- Does not grant permission to trade names, except as required.
- Unless required by law or agreed to in writing, the Software is provided “as is” without warranty of any kind.
- Authors or holders cannot be liable for any liabilities in its use.

8.3.4. CDDL-1.0 - Common Development and Distribution License

This License grants a world-wide, royalty-free, non-exclusive license. This allows the use, reproduction, modification, sublicensing and distribution of the original software without modifications [32]. If this license is applied, the following conditions must be met:

- Availability of Source code.
- Any modification applied is governed by the terms of the license and require a notice that identify the contributor.
- No terms that alter or restrict the terms of the license may be added.
- The Software provided “as is” without warranty of any kind.
- Authors or holders cannot be liable for any liabilities in its use.

8.3.5. LICENSE OF THE PROJECT

When making this project open-source, the main for it is to be improved and reviewed by its potential users. This can allow, not only to improve the current appearance of the code, but also to add new functionalities.

Considering the previous licenses presented, the license chosen for the development of this project has been the General Public License version 3, or GPL-v3.0 [33], that states the following:

This program is designed as a home or office intrusion detection system that sends you notifications directly to your telegram account.

Copyright (C) 2018 Irina Camacho

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>. [33]

8.4. SOCIO-ECONOMIC IMPACT

Conventional intrusion detection systems usually imply a user looking for a company that sells these kinds of products and employ their services.

Service fees from those companies can be expensive, and not economically in reach for everyone. These companies charge for both, the installation of the different materials the user has asked for, and monthly service fees.

Below, a table comparing current prices in the Spanish market is depicted, including prices by security companies, or other systems that are in the market. This table includes the minimum price for installation but excludes the energy that it will consume in a year.

System	System Price (€)	Monthly fee (€)	Total (€/year)
Securitas Direct	289 €	32 €	673 €
Segur 24	272 €	20 €	512 €
Alartec	230 €	29 €	578 €
Visegur	205 €	28 €	541 €
Blaupunkt SA 2700	368 €	0 €	259 €
G5 Touch	195 €	0 €	195 €
Netgear Arlo Pro	356 €	0 €	356 € ³

TABLE 75: MARKET PRICES COMPARISON TABLE

Now, we will detail the cost of the materials used in this project, and how much it costs to maintain it active per year, for comparison.

Device	Price (€)
Raspberry Pi	49.99 €
Arduino UNO	20.95 €
PIR Sensor	1.29 €
Camera	16.25 €
Total	70.48 € ⁴

TABLE 76: SYSTEM PRICE TABLE

In order to maintain active our system, we must consider how many Watts it consumes, and how much that would consume in an entire year. Given that the entire system is plugged into the Raspberry Pi, the energy required for it to function, is that of the Raspberry Pi.

Device	Energy Required (W)	Anual Amount (kWh)	Price (€/kWh)	Anual Price (€)
Intrusion System	5W	43.8kWh	0.165€	7.23 € ⁵

TABLE 77: ELECTRICITY COSTS TABLE

³ Prices obtained September 2018

⁴ Prices obtained September 2018

⁵ Prices obtained from Iberdrola, September 2018

In this, it would be necessary to also include the costs for having internet connection at home but considering that 83,4% of the Spanish population has internet access at home⁶ [34], this cost has not been considered.

From the above tables, it can be easily seen that the price for the materials required for the system, is half the price of the other systems in the market. Of course, in the case that the end user changes some of the materials in the system, the price for it may vary.

Taking into consideration the current economic state of the country, this project will be able to benefit everyone, given that the impact on one's economic benefits is not as high as other systems that are currently in the market.

⁶ Instituto Nacional de Estadística. September 2018

9. CONCLUSIONS

This section will explain the personal conclusions drawn from the development of this project, as well as different personal ideas for future implementation and improvement of the project.

9.1. PERSONAL CONCLUSIONS

The field of intrusion detection systems is a vast and highly evolutionary one. Since the existence of microcontrollers, people have developed and studied ways to protect one's home or office using competitive alternatives to those in the market.

The first approach to this project was to present a simple system that would allow users to have a low-cost surrogate to the systems in the market. Not only this but allow the user to control the system through their smartphones, tablets or computers.

If we take into consideration the objectives established in section 1 of this document, which are:

- Present the user with the different functionalities that the system will have with both an interactive keyboard, and the option to insert it directly through a command in Telegram.
- Inform the users when a movement is detected in their home/office.
- Allow the user to turn on and off the alarm system from wherever they may be.
- Allow the user to take pictures at any time as long as the alarm system is turned on.
- Ease the user the option to remove the files generated by the camera by sending a simple command.

Below, a table comparing the characteristics of the systems in the market, alongside the system presented in this project, will be presented.

System	Response Time	Maintenance	App control	Image capture	Video capture	Motion sensors
Proposed System	User	User	✓	✓	✓	✓
Securitas Direct	30 min	✓	✓	✓	✓	✓
Segur24	30 min	✓	X	X	✓	✓
Alartec	50 min	✓	✓	X	✓	X
Visegur	60 min	✓	✓	X	✓	X
G5 Touch	User	User	✓	✓	✓	✓
Blaupunkt SA2700	User	User	✓	✓	✓	✓
Netgear Arlo Pro	User	User	✓	✓	✓	X
Raspberry Pi Project 1	User	User	X	✓	✓	✓
Raspberry Pi Project 2	User	User	X	✓	✓	✓

TABLE 78: SYSTEMS CHARACTERISTICS COMPARISON

By looking at the table below, it is possible to observe that the proposed system for this project, provides most of the most-efficient characteristics from the private-security companies, while also maintaining a low cost.

We can say, after finishing the development of the project, that those objectives have been attained with the desired functionality. Not only this, but it has allowed to apply knowledges acquired during the study of the bachelor's degree.

Last, we can establish, several future lines of work, in order to allow the system to be more secure and introduce new functionalities, according to the user's needs.

9.2. FUTURE WORK

In this section, we will number and explain the different future lines of work presented for this Bachelor's Thesis.

9.2.1. IMAGE SECURITY

In the case that someone has perpetrated a house or office, it is necessary to keep the original files to present them to law officers or a judge in order for the culprit to be sanctioned. For this, the integrity of the images provided must prove to not have been interfered with. This could be possible by encrypting the files in the system.

For the possible encryption of these files, a hash encryption could be used. Providing the user with the original image, and the hashed imaged. These images could be sent to the user in a .zip file, also encrypted with another hash function. Hence, the user would not only receive the image automatically, but also a secure way of providing the evidence to law enforcers if need may be.

9.2.2. AUTOMATIC FILE DELETION IN RPi

In order to remove the current /clear function of the system, it could be possible to develop an automatic deletion system. This could also allow the system to ask the user to modify the period of deletion. For this, the previous step must have been developed to ensure safe-keeping of the possible images required.

9.2.3. WIRELESS SYSTEM

One of the disadvantages of the system developed is that it requires to be connected to for it to work. This could be solved by adding components that work wireless, and or could connect to our Raspberry Pi through Bluetooth. This would allow users to position the sensor as closest to the entry ways as possible, while putting the camera in the best position to capture an image of a possible perpetrator.

9.2.4. CAMERA NIGHT-VISION AND ROTATION

At the moment, without light, the camera captures images that are almost completely black. This could be solved by adding a night-vision camera to the system. It would also be an upgrade, if the system could provide the users with a motor that would allow them to move their camera. This could be possible by adding a servo motor, allowing users to rotate their camera both horizontally and vertically.

9.2.5. FACE RECOGNITION SOFTWARE

One of the main evolutions in the field of Artificial Intelligence for Intrusion Detection Systems, are the algorithms designed for face recognition. This technology can help identify individuals with images and/or video frames. By implementing this software, we would allow users to have their alarm system armed, while people of confidence can check on their homes. This system could also allow the users to create a new person on the database or add images to said database for the people that are already identified in the system.

Bibliography

- [1] [Online]. Available: <http://www.interior.gob.es/documents/10180/7146983/informe+balance+2017+cuarto+trimestre+v.2.pdf/99f3e28b-080b-4d68-b19b-4a83eafc2d7f>.
- [2] [Online]. Available: <https://www.consumerbarometer.com/en/trending/?countryCode=ES&category=TRN-NOFILTER-ALL>.
- [3] I. B. Tabanera, "Sistema Vigilancia Low-Cost," [Online]. Available: <https://eprints.ucm.es/31300/1/Memoria%20TFG%20SecBerry-Sistema%20de%20videovigilancia%20lowcost%20sin%20Autorización.pdf>.
- [4] E. Golge, "Raspberry Pi Home surveillance Python," [Online]. Available: <https://hackernoon.com/raspberrypi-home-surveillance-with-only-150-lines-of-python-code-2701bd0373c9>.
- [5] T. Wilmshurst, Designing Embedded Systems with PIC Microcontrollers: Principles and Applications.
- [6] T. Cantrell, "Microchip on the March," [Online]. Available: https://web.archive.org/web/20070927214629/https://www.circuitcellar.com/library/designforum/silicon_update/3/index.asp.
- [7] "Arduino," [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>.
- [8] "Arduino Uno Tech Specs," [Online]. Available: <https://store.arduino.cc/arduino-uno-rev3>.
- [9] "Raspberry Pi 3 B+ Specifications," [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>.
- [10] "ESP8266 Hardware," [Online]. Available: https://www.espressif.com/sites/default/files/documentation/0c-esp-wroom-02_datasheet_en.pdf.
- [11] "ESP8266 Specifications," [Online]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
- [12] "Basics of UART communication," [Online]. Available: <http://www.circuitbasics.com/basics-uart-communication/>.
- [13] "Basics of SPI Communication Protocol," [Online]. Available: <http://www.circuitbasics.com/basics-of-the-spi-communication-protocol/>.
- [14] "Basics of I2C Communication," [Online]. Available: <http://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-I2C-Single-Master-Single-Slave.png>.
- [15] D. A. M. John Francis Bell, "Instant Messaging System". United States of America Patent 60/34,197, 10 July 2003.
- [16] [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture.
- [17] [Online]. Available: <https://www.uml-diagrams.org/component-diagrams.html>.
- [18] X.-j. C. Yi-yuan Fang, "Design and Simulation of UART Serial Communication Module Based on VHDL".

- [19] "Telegram API," [Online]. Available: <https://core.telegram.org>.
- [20] "Telegram API Documentation," [Online]. Available: <https://python-telegram-bot.readthedocs.io/en/stable/index.html>.
- [21] "Python Telegram API," [Online]. Available: <https://github.com/python-telegram-bot/python-telegram-bot/wiki>.
- [22] [Online]. Available: <http://softwaretestingfundamentals.com/software-testing-levels/>.
- [23] "HAYS," [Online].
- [24] [Online]. Available: https://ec.europa.eu/info/law/law-topic/data-protection/reform/what-does-general-data-protection-regulation-gdpr-govern_en.
- [25] [Online]. Available: <https://www.boe.es/buscar/act.php?id=BOE-A-2014-3649>.
- [26] [Online]. Available: <https://www.boe.es/buscar/act.php?id=BOE-A-2014-3649>.
- [27] [Online]. Available: https://www.boe.es/diario_boe/txt.php?id=BOE-A-2011-3170.
- [28] [Online]. Available: <https://opensource.org/licenses>.
- [29] [Online]. Available: <https://opensource.org/licenses/gpl-license>.
- [30] [Online]. Available: <https://opensource.org/licenses/MIT>.
- [31] [Online]. Available: <https://opensource.org/licenses/Apache-2.0>.
- [32] [Online]. Available: <https://opensource.org/licenses/CDDL-1.0>.
- [33] [Online]. Available: <https://opensource.org/licenses/GPL-3.0>.
- [34] [Online]. Available: https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176741&menu=ultiDatos&idp=1254735976608.
- [35] "Basics of SPI Communication," [Online]. Available: <http://www.circuitbasics.com/basics-of-the-spi-communication-protocol/>.

ANNEX A: USER MANUAL

Throughout this section, we will explain how the system is installed, configured and operated. We will divide this manual in two main parts. The first one will be devoted to explaining how to install the required software in the different devices. The second one will be devoted to show how the system is operated.

Software installation

In this section it will be explained what software is needed to use the system and how to install it and configure it in the different devices.

Raspberry Pi

For the system to work we will have to flash a Raspbian7 image into an SD card. This task can be done by using a computer with MS Windows, MacOS or a Linux distribution. To illustrate how it is done we will assume that we are using a computer with either Linux distribution or a MacOS installed.

Once we have downloaded Raspbian image, we will have to follow these steps:

1. Insert SD card in our computer.
2. Check what the device name is in the system. To do so we can open a terminal and type the command:

```
ls /dev | grep mmcblk
```

3. Flash the image typing the following command into a terminal:

```
dd -bs 4M if=<Path to raspbian image> of=<SD card device name>
```

Note that this is no the only method available to flash an image into an SD card.

⁷ <https://www.raspberrypi.org/downloads/raspbian/>

Once we have Raspbian installed we have to configure it and install all the required packages needed for the application to work. We will need to install the following packages:

1. Python, Python Pip and Git. To install them we will have to open a terminal and type:

```
sudo apt install python python-pip git
```

2. Python telegram bot libraries. To install them we will have to open a terminal and type:

```
pip install python-telegram-bot
```

3. Finally, we will have to install the bot. To install it we will have to execute the following command in a terminal:

```
git clone <URL> &&  
    cd FluffyTheGateKeeper &&  
sudo cp motiondetection_bot /usr/local/bin
```

Once everything is installed we will have to configure Raspbian to execute the application on each restart, so in case of undesired reboot, because of a power loss or any other reason, the application start running automatically without the need of any human interaction. To accomplish this task, we will have to do the following command:

```
sudo sed -i 's/exit 0/ motiondetection_bot\nexit 0/g' /etc/rc.local
```

Arduino

To upload the software into the Arduino device we are going to use Arduino IDE8. We download the full software package from the page <https://github.com/Notsleptindays/FluffyTheGateKeeper>

Once we have done that, we open Arduino IDE application and in the top bar menu click on "File" tab. A dropdown list will be shown, among the displayed items we have to click

⁸ <https://www.arduino.cc/en/Main/Software>

the one that says “Open”. A file explorer will then show up, we will have to navigate to the location where we downloaded the software and open the file “MotionSensorArduino.ino”.

Finally, we will have to connect the Arduino device through the USB cable to the computer and press the button “upload”.

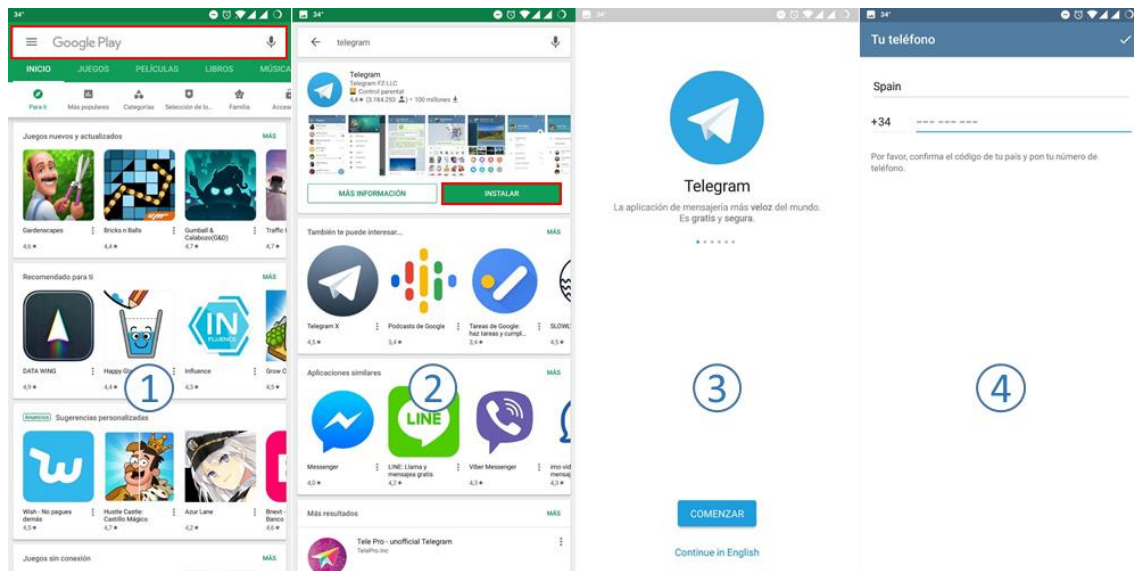


9

Mobile devices

In order to operate the software, it is necessary to download Telegram application in our mobile phone. Although this application is multiplatform and can be executed in mobile phones, tablets, computer and web application, it must be paired to a telephone number.

Mobile application is available in all official application stores. We will illustrate how to download the application in an Android device, but similar processes are used to install the application in other platforms.



10

1. Open Google Play Store application and search the application “Telegram”.
2. Install Telegram application.

⁹ Arduino IDE Upload

¹⁰ Telegram installation from Google Play Store

3. Open Telegram application.
4. Insert your phone number.

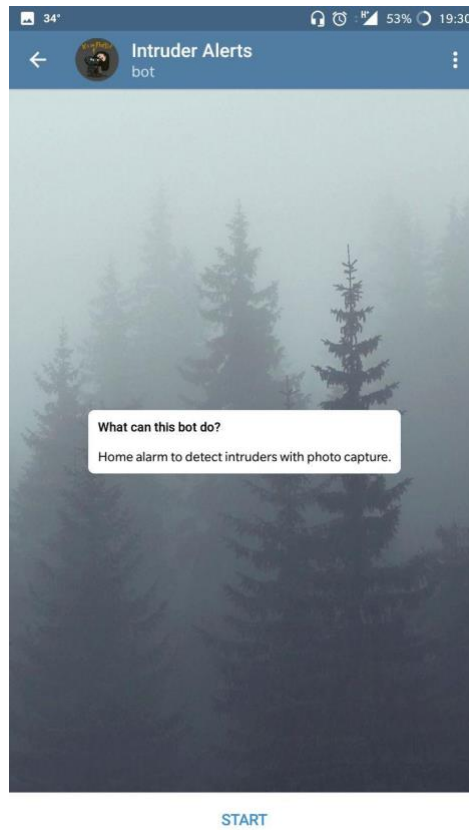
Software operation

Once everything is installed, you will have to start a conversation with the bot in order to operate the system. To do so you will have to search bot's name in telegram application. The search process has the following steps:

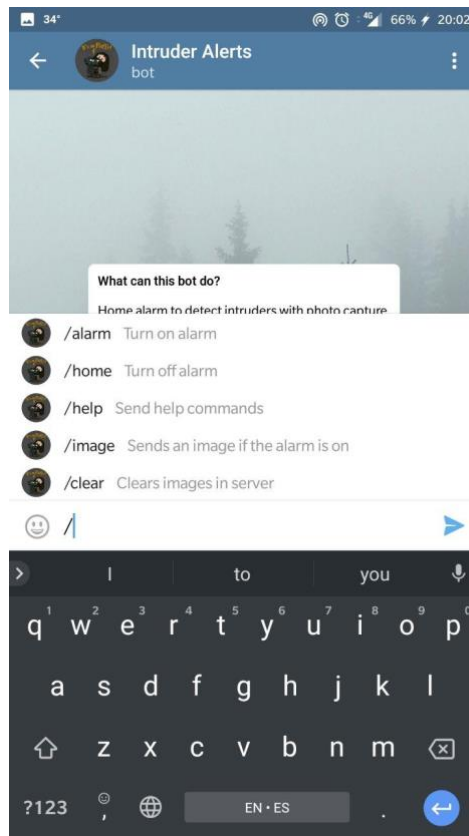
1. Search for the bot



2. Press start



3. Select command



The possible commands for the bot are:

- **/start:** a keyboard with the options for the bot will be presented, as well as a brief explanation of the different commands the bot has for possible execution.
- **/help:** a brief explanation of the different commands the bot has for possible execution.
- **/alarm:** turn the alarm on
- **/home:** turn the alarm off.
- **/clear:** clears images from System.
- **/image:** If the alarm is on, the system will send a picture.

ANNEX B: ARTÍCULO 7 LEY DE SEGURIDAD PRIVADA

Artículo 7. Actividades excluidas.

1. No están sujetas a esta ley las actuaciones de autoprotección, entendidas como el conjunto de cautelas o diligencias que se puedan adoptar o que ejecuten por sí y para sí mismos de forma directa los interesados, estrictamente dirigidas a la protección de su entorno personal o patrimonial, y cuya práctica o aplicación no conlleve contraprestación alguna ni suponga algún tipo de servicio de seguridad privada prestado a terceros.

Cuando los interesados tengan el carácter de empresas o entidades de cualquier tipo, en ningún caso utilizarán a sus empleados para el desarrollo de las funciones previstas en la presente ley, reservadas a las empresas y el personal de seguridad privada.

2. Queda fuera del ámbito de aplicación de esta ley la obtención por uno mismo de información o datos, así como la contratación de servicios de recepción, recopilación, análisis, comunicación o suministro de información libre obrante en fuentes o registros de acceso público.

ANNEX C: ARTÍCULO 427 LEY DE SEGURIDAD PRIVADA

Artículo 42. Servicios de video-vigilancia.

1. Los servicios de video-vigilancia consisten en el ejercicio de la vigilancia a través de sistemas de cámaras o videocámaras, fijas o móviles, capaces de captar y grabar imágenes y sonidos, incluido cualquier medio técnico o sistema que permita los mismos tratamientos que éstas.

Cuando la finalidad de estos servicios sea prevenir infracciones y evitar daños a las personas o bienes objeto de protección o impedir accesos no autorizados, serán prestados necesariamente por vigilantes de seguridad o, en su caso, por guardas rurales.

No tendrán la consideración de servicio de videovigilancia la utilización de cámaras o videocámaras cuyo objeto principal sea la comprobación del estado de instalaciones o bienes, el control de accesos a aparcamientos y garajes, o las actividades que se desarrollan desde los centros de control y otros puntos, zonas o áreas de las autopistas de peaje. Estas funciones podrán realizarse por personal distinto del de seguridad privada.

2. No se podrán utilizar cámaras o videocámaras con fines de seguridad privada para tomar imágenes y sonidos de vías y espacios públicos o de acceso público salvo en los supuestos y en los términos y condiciones previstos en su normativa específica, previa autorización administrativa por el órgano competente en cada caso. Su utilización en el interior de los domicilios requerirá el consentimiento del titular.

3. Las cámaras de videovigilancia que formen parte de medidas de seguridad obligatorias o de sistemas de recepción, verificación y, en su caso, respuesta y transmisión de alarmas, no requerirán autorización administrativa para su instalación, empleo o utilización.

4. Las grabaciones realizadas por los sistemas de videovigilancia no podrán destinarse a un uso distinto del de su finalidad. Cuando las mismas se encuentren relacionadas con hechos delictivos o que afecten a la seguridad ciudadana, se aportarán, de propia iniciativa o a su requerimiento, a las Fuerzas y Cuerpos de Seguridad

competentes, respetando los criterios de conservación y custodia de las mismas para su válida aportación como evidencia o prueba en investigaciones policiales o judiciales.

5. La monitorización, grabación, tratamiento y registro de imágenes y sonidos por parte de los sistemas de videovigilancia estará sometida a lo previsto en la normativa en materia de protección de datos de carácter personal, y especialmente a los principios de proporcionalidad, idoneidad e intervención mínima.

6. En lo no previsto en la presente ley y en sus normas de desarrollo, se aplicará lo dispuesto en la normativa sobre videovigilancia por parte de las Fuerzas y Cuerpos de Seguridad.

ANNEX D: GENERAL DATA PROTECTION REGULATION [24]

Regulation (EU) 2016/679, the European Union's ('EU') new General Data Protection Regulation ('GDPR'), regulates the processing by an individual, a company or an organization of personal data relating to individuals in the EU.

It doesn't apply to the processing of personal data of deceased persons or of legal entities.

The rules don't apply to data processed by an individual for purely personal reasons or for activities carried out in one's home, provided there is no connection to a professional or commercial activity. When an individual uses personal data outside the personal sphere, for socio-cultural or financial activities, for example, then the data protection law has to be respected.