University of Essex

Internet of Things: Connections, Data exchange and security

Michalis Panayides

Registration number: 1701922

Supervisor: Dr Hossein Anisi

Second Assessor: Mr Joseph Walton-Rivers

BSc Computing and Electronics

Acknowledgements

I would like to express my gratitude to my supervisor Dr Hossein Anisi for giving me suggestions and comments about my project. Dr Hossein Anise helped me and gave me important guidance though the time I was working on this project. Furthermore, I would like to thank the module supervisor Dr Viswanathan Mohan for providing help and support throughout the academic year.

Also I would like to mention the support I received from family, friends and colleagues.

Summary

This project is about securing the connections and data exchange in low power devices used for Internet of Things. Nowadays IoT is becoming very popular, this kind of project can be helpful when talking about the connection between the devices and the issues of security than can occur. Project uses two Raspberry Pi that are connected based on client-server mode, one Pi is the server and the other Pi is the client, Python is used for coding. The purpose of the project is to create a connection that is secure, so the data sent through this connection will be safe. Connection is made safe by implementing the MQTT protocol using username and password. Using some commands from the client data will be sent, after the connection is made, no other device can be communicating with the server at this stage, only one client at a time can send messages to the server. Using a MQTT broker there is the capability to manipulate the model of operation and add more clients to the network, if it's needed. The program is executed from the command line. In few words this project is simulating an IoT system.

Table of Contents

1.1 Introduction

This project is aiming to secure data connections and data exchange through low power devices, those are mostly used in Internet of Things applications. Since the project relates with IoT it was important to work on devices, used for these purposes. Raspberry Pi is commonly used since they can support the operations needed for the project. The idea is to connect to a server from a client and secure this connection, so it will be safe to make the data exchange. There is a communication through the two devices, server creates a connection and client connects to this connection. The two devices are exchanging messages, through some specific commands.

1.2 Importance of project

Nowadays IoT is becoming very popular, this kind of project can be helpful when talking about the connectivity between devices. Everyday something new is coming out which is related to Internet of Things. More people in recent years started to understand that Internet of Things is making our life easier. Maybe the most interest thing about IoT is that, you don't have to be a computer scientist or engineer in order to create a small IoT system. An example can be a hot water control system in a house, where some sensors are used to control the temperature of the water. The future could be the smart cities, in instance the city of Barcelona has implemented several IoT initiatives which helped solve the issue of parking. [1]

There is no doubt that Internet of Things makes our life easier, although there is one main drawback, which is the lack of security and the need to secure these connections. Having so much data sent between many devices making hundreds of connections it is obvious that is essential to shield the security of those systems and prevent an unauthorised access which will lead in lack of privacy and security. In some cases, very sensitive information is transported through the connections like the adjusting traffic light cadence. What will happen if a hacker decides to attack such a system?

For this reason, is important to research the area of security in this industry and try to find how to make the security on such systems strong. This project will help someone to understand how Internet of Things works and give an idea of what can be done in future to make this industry better.

1.3 Aims and objectives

Internet of Things is a huge chapter, which is involving hundreds of technologies, methods and protocols. The aim of the project is to make the user get used to connections that happening inside an IoT system, and give out an idea, of how this is possible. Lately the objective is to make a start into the area. Understand some principles, try some methods and get hands on Internet of Things. The project was made to give some education and information about Internet of Things to users that initially having contact with the subject. Project is aiming to find ways to increase security. Adding to this, some protocols that are applied to IoT applications are explained, as and some issues engineers and scientists faced. The development and evolution IoT offer to devices or things to make them smarter is mentioned. This project can act as a simulator to anyone that wants to explore more about Internet of Things connections and data exchange. Moreover, some suggestions are presented, which can be used to improve security in future.

1.4 How it works

The structure of the project is simple to understand. Two Raspberry Pi 3 Model B+ are acting as server and client. The IP of the two Raspberries is known and is entered manually to the client. One python program is first executed from server Pi's command line which is creating a connection, where the client Pi will connect, knowing the IP address of the server. Then from client Pi's command line another python program is executed, and the connection is established, after this the two devices communicate, by writing some specific commands, the client has the option to terminate the connection. Adding on this the two Raspberry Pi must be connected to the same network. The user in some parts will need to follow some instructions to learn how an MQTT broker is working and will learn how to configure some parts of it.

2.1 Literature Review

2.2 What is Internet of Things

Internet of Things (IoT) is one of the most popular topics in our days in the field of Computer Science, Electronic engineering and in general in the field of technology. Internet of Things is a mixture of software and hardware. In the last years computer scientists and engineers combined their knowledge and experience to make some systems do smart things. The word 'smart' is associated with IoT, since useful things are happening, that make our lives easier.

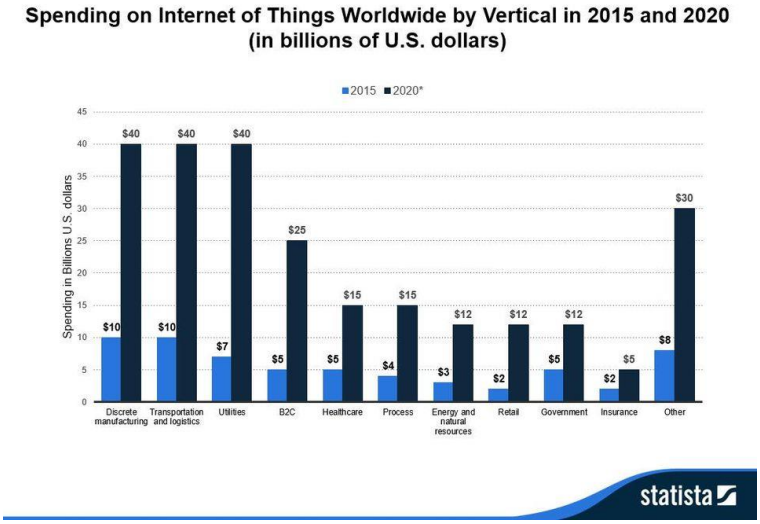
In the acronym IoT, 'Internet' stands for the connections that are happening into a system, by this we understand that we need some kind of connection, through these connections data will be transported. An example of such a data can be the temperature, this kind of data can be gained through a temperature sensor. The connection can be either through the WiFi or by a cellular connection (3G/4G). Moving to the next letter in the acronym we found out the word 'Things'. We can say that we are talking about an object, something that usually is small in size and gives us directly the data, for example a sensor [2]. Some data can be given though some coding program, depending on the use of the particular project. In other words, the IoT is working by having a piece of data, collected from a sensor or from a program which is then edited to do something useful for the user. Most IoT projects are created having as a purpose to collect the data then send it, receive it and act on it.

Adding to this another feature of IoT is the Machine-to-Machine (M2M) communications, which is the establishment of connections between several devices. Furthermore, Internet of Things offers many automatizations which is meaning that no individual is needed, less time is wasted to control this kind of system. This is a result of efficiency. [3]

A brilliant example of an IoT product can be a smart watch. People previously were using a watch to just see the time, last few years things have been change dramatically. Now with a smart watch you can, make a call, but most importantly you can do much more, like monitoring your heartbeat when exercising. A smart watch has many sensors like accelerometer, gyroscope, heart monitor etc, from there some values are monitored. Through this, data is transported though the several connections that are happening, to a server or to a smartphone. This data can be used to create some analytics, for instance about the time you spent each day doing exercising or the steps you run on a particular day. Some modern smartwatches are even able to detect when a person is having a heart attack and call an ambulance. A smartwatch can use WiFi,3G/4G or Bluetooth to send data.

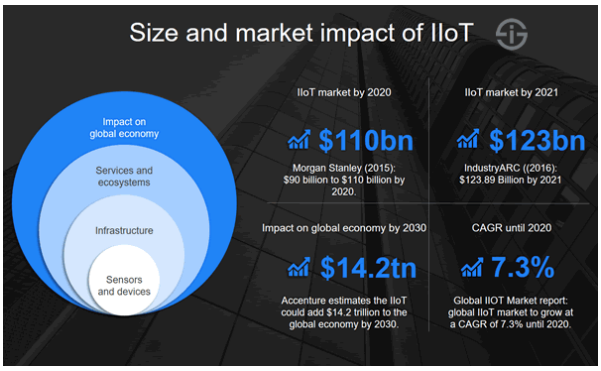
Through the above example we can also understand that Internet of Things is making devices and things becoming evolutionary. Devices that previously were used to do few things, now are doing much more and they are becoming smarter.

The industry of Internet of Things is rapidly developing, the specialists say that IoT is the future. According to recent researches year by year more and more companies are investing in Internet of Things. A article in 'Forbes' is confirming this. [4] Through the following graph it is clear to see a dramatic increase of spending in Internet of Things. The graph illustrates the spending on Internet of Things in 2015 and in 2020, we can see that companies coming from different categories are spending dozens of billions of dollars.



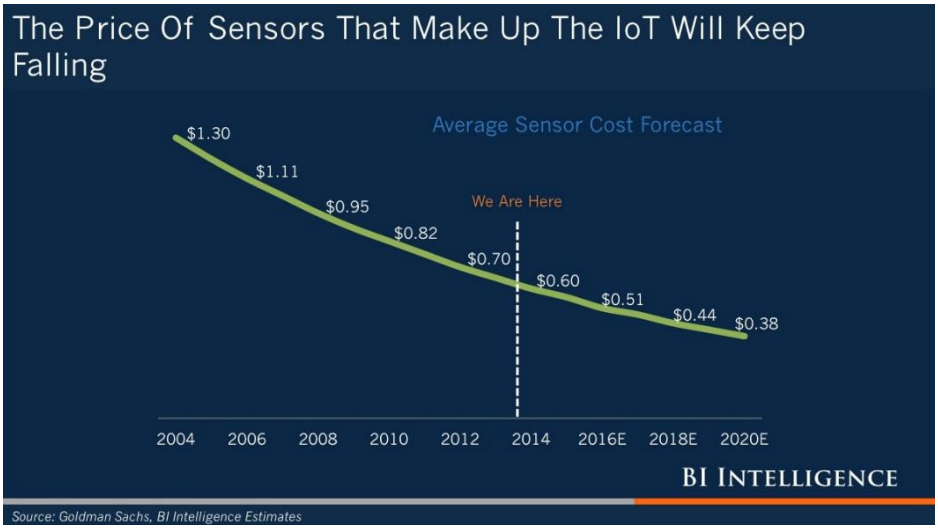
Source: Statista[5]

Looking at another interesting graph below, to understand the huge impact IoT has in the last few years we can see the IoT market value is increasing in incredibly fast pace. In 2020 the IoT market is around $110bn and by 2030 this value will reach the $14.2tn![6]



Source: i-scoop.eu,The Industrial Internet of Things (IIoT): the business guide to Industrial IoT[7]

Adding to this, sensors are essential in most of IoT project, because of their important role and the big demand, this led to the fall of their price, as is presented in 2014, the graph following an average sensor's forecast price has fallen to $0.38 in 2020 from $1.30 it was in 2004. [8]



Source:Sogetilabs[9]

In order to have growth and development, it means that a variety of advantages are offered in Internet of Things and this is true. Starting with, one good thing that IoT is offering is the ability to accessing information easily, in instance you can control a device that is in your home when you are in work, in real time. Also, before the discovery of IoT it was difficult to connect together devices, but because of the way IoT works this is not a problem anymore, techniques, implementations and protocols developed so the communications are now clearer and easier. Data packet transferring over a network is saving time and money, before same data in size would take much more time to be transferred, now data is transferring many times faster. [10] Furthermore, the automation that is offered through Internet of Things has reduced the human interaction in a system but most importantly it increased the efficiency of a system, which means that productivity is increased. Because of the machine-to-machine communication, more information is gained which can help you make better decisions. [11]

For sure Internet of Things provides many positive things, unfortunately there are and many disadvantages that can cause a few issues. When there are so many connections and communications, between many devices, it is obvious there is a huge risk of losing some data. Most of the devices that are used in IoT have limited memory and the security offered from these devices, is another problem to consider. Privacy and security is a major drawback of this technology, since sometimes more than one companies are monitoring data. Another related issue is the data storage [12], so much data have to be saved and be safe. We can say that is some circumstances the networking behind an IoT system can be a little complex. In a scenario where a device is broken or it's wrong installed, this can create a loophole in the entire system, where this will cause many confusions or even a failure. Because of such an evolution, the need for so many connections, makes it a real challenging to achieve reliable connectivity. More and more devices are using internet to send and receive data, which leads to the problem about the connectivity. It is one the key factors of IoT and is essential to find ways to expand network, so all these connections can be possible and affordable.

As mention before there is a significant progress in this industry which in many cases can help for the better, although there and the cases where this evolution can make people lose jobs.[13] Some companies have implemented some methods of IoT, are hiring less people, which lead to lesser jobs opportunities. Adding to this too much development makes our life more depended to technology.

2.3 How to improve security

One big headache that IoT professionals are facing is to find ways to minimize the lack of security and privacy. The devices used to interact with IoT are small, that is meaning that the code written on such devices, may have some memory leaks and some vulnerabilities, those two things means that is easier for a hacker to break through, the solution to this problem can be the continuously testing.[14] There are well structured testing tools on the market that can help expose many issues of security. Moreover, reviews are a very popular method that will minimize the risk of a possible attack. Risk review involves testing, where a strong risk assessment will help to confirm that there aren't any gaps. Another thing can be done is to encrypt anything that can be encrypted. There are so many connected devices in an IoT system and is essential to encrypt the communications. Many devices are expected to be connected so, authentication is the key to secure what devices is logged into the system and when. This can be done by having a log file and by using certificates to help provide this authentication. [15] Where is possible is very methodical to isolate the Internet of Things network, it is advised to keep the IoT to their own private network. This will ensure that devices have internet and are also separated from the rest networking in a building, although this can be tricky since, a separated virtual network(VLAN) will have to be made so IoT devices will have access to internet, but they will not be able to give access to the other network, this have to be configure through the firewall. Unauthorised access to a system can be limited through software protection. By implementing a security measures in software's like a password will make it more difficult for hackers to gain access. Adding to this, using passwords or digit codes into specific parts of the system will minimize the chance for an attacker to break into a full system. Furthermore, is very essential that should be a constant monitoring of potential threats, so will be indications showing that something is going wrong. [16]

2.4 Securing connections

There are many techniques and options that can be used to secure a connection. In the IoT area there are two main protocol that offer encryption, but also there are still some risks. MQTT and CoAP are these protocols. The two protocols are the ones that are mostly used, in small scale IoT projects. MQTT is one of the most used protocols, as MQ Telemetry Transport is one lightweight messaging machine to machine protocol used in Internet of Things devices and it was firstly developed by IBM. The protocol runs over TCP/IP and is used to transport messages between devices. Furthermore, this protocol is open standard, and many implementations are available on open source. Adding to this, MQTT is ideal in cases where network bandwidth is limited. The project is designed so only one device is connected to the server, MQTT supports the client/server mode but also supports many clients to be connected to the server. This can be done because of the way the protocol operates. The architecture of the MQTT gives the opportunity to control almost everything in a connection this is done by a broker, which can be said that is the heart of the protocol. Broker controls communication and data exchange; broker is responsible to receive and filter the messages and it also decides where the data should go. To understand better the way a broker operates, you can imagine a list of pre-defined receivers of a message. The job of the broker is to send a specific message to the receivers interested in this kind of message. Because of the way it works, MQ Telemetry Transport version 3.1, makes the one to many messaging easier, something that makes the protocol better is the ability to pass username and password through SSL.[17] There are three quality of service levels, "Fire and forget", "delivered at least once" and "delivered exactly once". [18]

The other protocol used in machine to machine data exchange is the CoAP which stands for Constrained Application Protocol. It can be said that CoAP is a simple protocol designed for specific devices, like microcontrollers and constrained networks. This protocol is one-to-one protocol for transferring state information between client and server. To be able to operate in those kind of devices the protocol is low

overhead. Many people find CoAP similar to HTTP the truth is that indeed CoAP is based on HTTP but it has some differences. CoAP was designed to be optimized to work under IoT tasks. [19] This protocol is a REST[20] model. Also the packets sent through CoAP are much smaller that those of HTTP TCP flows and CoAP runs over UDP not TCP. The resources are made available through a URL where clients can access them, there are four methods to do this GET, PUT, POST, and DELETE. CoAP provides security applying default choice of DTLS parameters which is equivalent to 3072-bit RSA keys.[21]. There are two quality of service in this protocol the "confirmable" and "nonconfirmable". Confirmableare the messages that are acknowledged by the receiver with an ack packet. Nonconfirmable messages are "fire and forget". [22]

MQTT and CoAP are two useful protocols developed for IoT devices and they have many differences. These two protocols gives the opportunity to the user to work on them in different programming languages like Python, Java, C, C++, Android and iOS.

2.5 Differences of MQTT and CoAP

Starting with MQTT, it is a many-to-many communication protocol for passing messages between a number of clients through a broker. The client that is subscribed to a topic can receive the related data and it can publish data, where the broker will decide how to filter the data. CoAP is a one-to-one protocol that transfers data between a client and a server. CoAP operates better and is more useful in state transfer model. MQTT does a TCP connection to the broker, which normally don't have any issue when talking about NAT. The CoAP from the other hand because it doesn't have any filtering mechanism, like a broker for MQTT, CoAP sends the message through UDP packets.

In one hand MQTT doesn't provide labels to message to help clients understand it, the messages can be used for any purpose, although the clients must know the formats-up message to allow the communication. In the other hand CoAP, provides support for content negotiation and discovery which means that devices can probe each other to find ways of exchanging data.[23]

2.6 Ethics

Ethical matters are protecting the rights of the creator. This project is under the university policy of the intellectual property rights of a student. The intellectual rights state that a students has full ownership of their work and in case there is profitability through a third party, the student doesn't have to share it with the University. However, the student must recognise the support and the facilities provided by university in order to create the project, so there is full ethical understanding through both sides.

3.1 Project specification

Project is about securing connections and data exchange for IoT devices. Security is an important feature of the project. The foremost feature of the project is the communication between the two Raspberry Pi, the two devices should be able to receive a message and send a reply. The client should be able to connect to the server, where the server is starting the communication, a port is used to initialise a connection. After this, client knowing the IP address of the server is connected to this connection, server gives out the IP address of the client. Through this time only one device can be able to communicate with the server, a second feature is the message exchange between the two devices, the client types in some certain commands in order to have communication with the server, server also says

that has sent the data. The third feature is when a connection started between client and server, another client can connect with the server although the other client gets in a standby mode and can't have a conversation with the server.

The fourth feature is the MQTT implementation, in where the user will have to follow some instruction in order to set up a broker. In this feature in the first part, there two more Python programs that have to be executed and in the second user must follow some instruction to set up a broker and sent messages.

3.2 Technology choices and tools used

MQTT

It's very important in a project to choose the correct technologies, protocols and techniques that will work satisfactorily for the project. There were some decisions that must be taken based on research and one of the decisions was to choose between MQTT and CoAP. Both protocols have pros and cons, choosing which is more suitable, is depending on the application.

Even if both protocols are doing the same job more or less, one can be more suitable in this case, the huge difference between MQTT and CoAP is the way of their operation and their architecture. MQTT is a many-to-many communication protocol that is delivering messages between multiple clients, where CoAP is one-to-one protocol for transferring information between server and client. Furthermore, MQTT has a broker, which is filtering messages, broker sends specific messages to specific clients. Regarding persistent, MQTT broker can offer a persistent session, which is useful when there may be interruptions, by persistent session we mean that all information that is relevant to a client is saved on the broker. If there is no persistent session it means that, if there is any lost of connection or interruption the topics between client and broker will be lost and the client will have to re-subscribe. [24]. About security, MQTT because it uses IP/TCP connections it can support the SSL protocol. By Secure Sockets Layer protocol, the communications will be encrypted, which will can improve a lot the security of the project. Today the SSL is the predecessor of TLS(Transport Layer Security) encryption used in our days.

These abilities offered by MQTT make the project more versatile to be adopted in other situations and in large scale applications in the future, where the devices will be increased significantly. It important to filter the data and send it where it should, otherwise some other problems may occurred like short of storage. Also because of filtering if someone breaks into one device will be able to see what this device contains and not the whole data that flows through the system. [25]

Mosquitto

Another decision that must be made is which MQTT broker is better to use. There are quite a few brokers, although because of the use of Raspberry Pi, Mosquitto is suitable. Mosquitto is a message broker that implements the MQTT protocol, it is lightweight and suitable for low power single board computers. Adding to this, it offers a publish/subscribe model and it is running fast, which makes the testing easier. [26]

Raspberry Pi

Why Raspberry Pi is perfect for IoT projects? The purpose of the project is to make the user understand better the way of how Internet of Things work, for this reason rather than using two computers was more suitable, to use two devices that can support the idea of IoT. Raspberry Pi is one of the cheapest single board minicomputers in the size of a credit card, which it means that can be installed almost anywhere. Also, the Raspberry Pi 3 Model B+ has an ethernet port, where can be useful in some cases, where Wi-Fi and Bluetooth can be used to make a connection. Adding on this because of the way Pi is designed gives you the opportunity to use many operating systems, in this project a Linux based OS is used named Raspbian. Furthermore, this device can act as a server and as client which makes it ideal for the project. There are already programming languages installed on in and it can fully support any heavier operations in future, for example a sensor can be connected to it through, which can be used to gain some live data from environment. [27]

Python

Almost any programming language can be used in IoT, although Python in the last few years helped in the IoT development. Python is fast and easy to learn, comparing it with other languages like C++ and Java, Python is simpler. There are many libraries that support IoT mechanisms which makes parts of the coding less difficult. Adding to this, Python is embeddable and portable which makes it ideal for IoT applications where many embedded devices are used. Moreover, is a free open source language that provides support. Those are the reasons why Python is used. [28]

Jira

Jira is a software development tools that is used for individual projects and teams based on Agile methodology. Jira helps a lot with project management, since it offers many useful features that can be used to make someone be on track. Kanban boards are provided, issue tracking is available, progress reports and project backlogs are also available to the user, in order to help anyone with the management of the project. Another interesting feature of Jira is the backlogs it offers and the task management, as you can break into parts some parts of the project, in order to be more organised. [29]

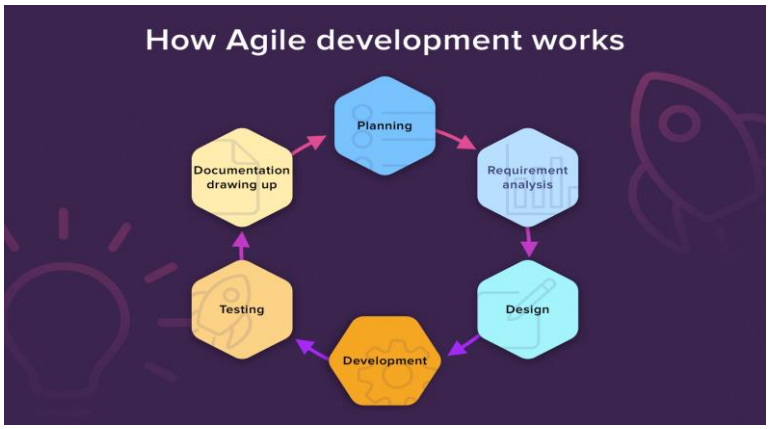
4.1 Methodology - Agile

In the industry there are hundreds of methodologies available, each one of them is different from the others, there is a wide range of options that you can choose, depending on the way someone wants to work on a project. Agile methodology is a modern and flexible approach to project management. This methodology is used for individual projects and team projects. Agile allows you to break a project into smaller parts that can be easier managed, by this you can change some features of the project if needed and be more adaptive to new additions or removals. In this methodology there are priorities, so some parts must be done sooner than others, although there is the opportunity to use more time in a particular task, so re-scheduling and early evaluation can be done in Agile. Adding on, this methodology gives you the chance to quick respond to any changes and have an adaptive planning. Agile puts the necessary pressure, so someone can be organised and be able to deal with something unexpected.

The way the methodology works is by having development life cycle for each sprint/task which most of the time, is made up of six stages.

1. Planning

2. Requirements
3. Design
4. Development
5. Testing/Evaluation
6. Documentation

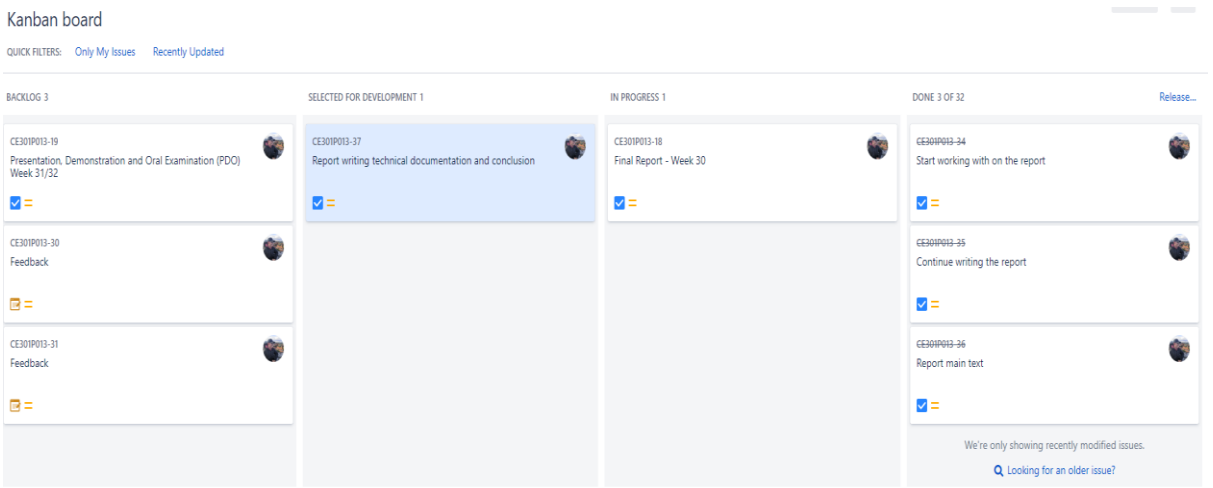


Source: Cleveroad[30]

Each task has a specific time to be completed, usually the average is 2-3 weeks, although there is the ability to work on more than one tasks on the same time. The cycle helps you to check if the requirements are met and if the task is completed correct.

4.2 Kanban board

Agile is the general methodology, Kanban is a sub-methodology of Agile and Kanban board which is available through Jira. A Kanban board is an agile project management tool created to help limit the amount of work in progress (WIP), visualize the work, and maximize the efficiency. [31] Kanban board helps you have a clear view about each task. Kanban in Jira has four columns, “Backlog”, “Selected for development”, “In progress” and “Done”. Having these four columns, you can easily see in which stage each task is. In backlog there are all tasks that are to be done in future, all tasks are in backlog before work started on them. The individual can add more tasks, as requirements can change through time. In the second column are the task selected, in third column are the task that are currently under process and in the final column are the task that are completed. Below there is the Kanban board near the end of the project.



Looking in the board above, we can see that the board is giving you all the details about was happening to the project this exact time, which is very helpful when talking about the organization of the project. Different parts of the project can be done in parallel in same time, giving the opportunity to monitoring those tasks the same time. The Kanban board allows you to stay updated and see the progress happening on the product.

Work in progress (WIP) limits the maximum number of tasks that can fit into a single column, meaning that helps you stick into the plan and don't commit too much work that in the end will affect negatively the product, since this work can be done in rush and cause problems.

Kanban has a number of benefits, we can say that the Kanban board is working in cycles as Agile methodology says, because actually the task are divided into phases, once a task or issue is created in backlog it can be sorted immediately and anytime you can see in which phase each task is. Furthermore, sometimes the priorities in a project change frequently which make Kanban board ideal in such cases. Kanban reduces the wasted time, as it is very easy to organise the tasks and do changes. Adding to this, efficiency is increasing. Using Kanban board you can have continuous delivery, which is important when you want to show that you are working hard on the project. Also, you can have feedbacks that help you improve some aspects and create a better product.

In a few words Agile and Kanban board offer lot of tools that are working in favour of the planning and the development of the project, helping us to achieve the best outcome.

5.1 Project Planning

It is essential to have a clear project plan and a timetable in order to be able to manage such a project. Starting from October and ending in early April, it is crucial to set up, carefully a detailed plan and a timetable from the start, which will be followed throughout the time working on the project. Taking into consideration the requirements and the deliverables, (deadlines) and also some circumstances that happened that was the timetable that I followed.

Event/Tasks	Date
First meeting	7 October 2019
Challenge week – Background reading	11 October 2019
Project idea	11 October 2019
Further research	14 October 2019
Change of supervisor	4 November 2019
Clear about project	6 November 2019
Research on Raspberry Pi	10 November 2019
Start working on Pi	20 November 2019
Interview with 2nd assessor	11 December 2019
Have connection	30 January 2020
Start sending messages	12 February 2020
Only one client	25 February 2020
MQTT implementation	7 March 2020
MQTT broker configurations	11 March
Research for security	22 March 2020
Start report writing	29 March 2020

The above table illustrated more or less is the plan management, although there are some things that are missing, like the meetings with the supervisor and the research done between some tasks. A research was done before moving to implementations, in instance a research has proven MQTT more suitable for the project. Moreover, in order to move to the next part of the project, the previous one had to be completed.

A key to achieve the best possible outcome for the project management, was the breakdown of the project into parts. Doing this, you can see how much time you are spending into a part and be able to understand if something is going well or not. Also having a good plan, knowing what to do, you are disciplined and consistent towards the project. Of course, there will be some issues and difficulties that may cause some delays, but you have to be flexible to deal with and give the best you can.

5.2 Time management

As it is known time management helps everybody be on track and manage the works that must complete. Organization and time are going together, so if you manage your responsibilities well and divide time among them correct, this will be beneficial. Although, sometimes even if you organise everything well, unexpected situations can occur and make you lose some time. A useful method to manage time is by estimating how much time you should spend in a part, although again its just an estimation, because you may face some issues and need more time in some parts.

5.3 Cumulative Flow Diagram (CFD)

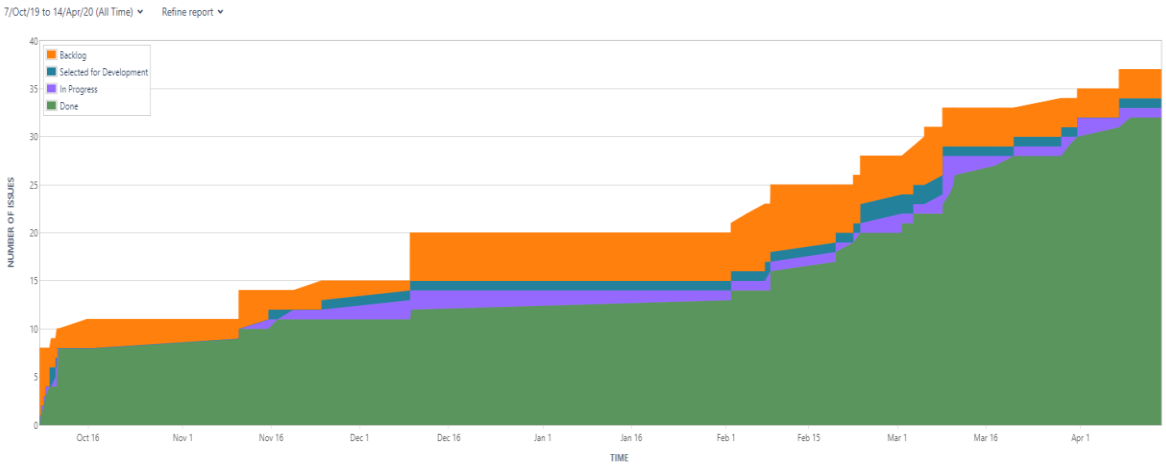
Jira offers many tools that helps you do evaluations regarding the project planning and time management. One diagram that is created by Jira when using the Kanban board is the Cumulative Flow Diagram.

What the graph crated shows:

1. the statuses of issues/task over time
2. the quantity of work in a particular time
3. the backlog which is the work that have to be done
4. the tasks that are in progress
5. the tasks that are selected for development
6. the tasks done

All the data is gained from Kanban board and is depending from the columns of it.

Below is the Cumulative Flow Diagram for the project.



Looking on the graph above, we can see at the start there is a rapid increase in the number of issues, this was expected since that period was the challenge week, so there were many things to do, and a lot of stuff in backlog. After this period there is a steady line, and there was when there was a supervisor change, there was an issue and I notified after almost a month, so I lost some time there, until to meet with the new supervisor and agree on some project changes. From November 16 until mid-September the graph illustrates a stable increasing and a parallel progressing, there was the time of Interim Interview, and actually this time was where I started working on the Raspberry Pi because of the some changes, as mentioned above, but and also because some time was needed to get the hardware needed for the project. After this period was the Christmas Vacation, after that period, from February until April there was a steady increase of work done and tasks, that was normal because this period was the time left to work on the project. At some point in early February we see a small widening between the lines, because of some trouble I faced but after this period around February 15 there is a parallel progressing, tasks are entering the workflow in parallel with those being finished. The graph is presenting the workflow throughout the project, which as we see was stable, except some small periods, where there were some situations and some issues.

Jira link for Cumulative Flow Diagram (CFD):

https://cseejira.essex.ac.uk/secure/RapidBoard.jspa?rapidView=1092&projectKey=CE301P013&view=reporting&chart=cumulativeFlowDiagram&swimlane=1846&swimlane=1847&column=4716&column=4717&column=4718&column=4719

5.4 Control flow

Control flow is another tool that is available through Jira and as CFD, uses the Kanban board data to identify the process and estimate future performance. There are three values that are monitored:

1. Visibility
2. Efficiency
3. Predictability

Below is the Control Chart for the project.

Looking at the graph, we can see that the red line illustrates the average, where the blue line shows the efficiency, more or less the control chart attributes the same as the CFD, the efficiency through the time was near the average, although after February, the efficiency line have been rolling average which indicates that was a process and increased throughput. Because of some issues there were in some tasks, in some periods we see there was a rapidly increase or decrease in efficiency, for instance see the period for early March.

Jira link for Control Flow Diagram:

https://cseejira.essex.ac.uk/secure/RapidBoard.jspa?rapidView=1092&projectKey=CE301P013&view=reporting&chart=controlChart&days=0

6.1 Technical documentation

The project is different from some others because of the concept of Internet of Things. There are two different devices (Raspberry Pi), one is the server and the other is the client. In the first part user must execute two different codes from the command line, one from each device. The programs where written, and you have to know the server's IP address in order to make the connection. After running the server's code, you are creating a connection, and the server waits for a connection to be made. After running the second code from client, a message is displayed in server's terminal, showing the IP address of the client, the same happen and to the client's terminal as server's IP address is shown. After this through some specific commands a conversation is happening. Regarding the second part of the project the user have to follow some instructions in order to learn how to set up a broker.

More information on technical requirements are found in the following link in GitLab:

https://cseegit.essex.ac.uk/ce301_2019/ce301_panayides_m/-/blob/master/Technical%20documentation/README.md

6.2 Architecture of project and Development

Starting with the making of the connection, server's code was written, using some python libraries and an identified port was used to create a 'socket' so the client could connect to it. In this stage a socket was created, if there is any issue, for example if there is no internet access, a message is printed out stating the reason why socket can't be created. When the code from client is executed and client connects to the socket, server is issuing a message, giving the IP address of client. That was the first part of the project, after this part another piece of code was created so the client, would use the same port to connect.

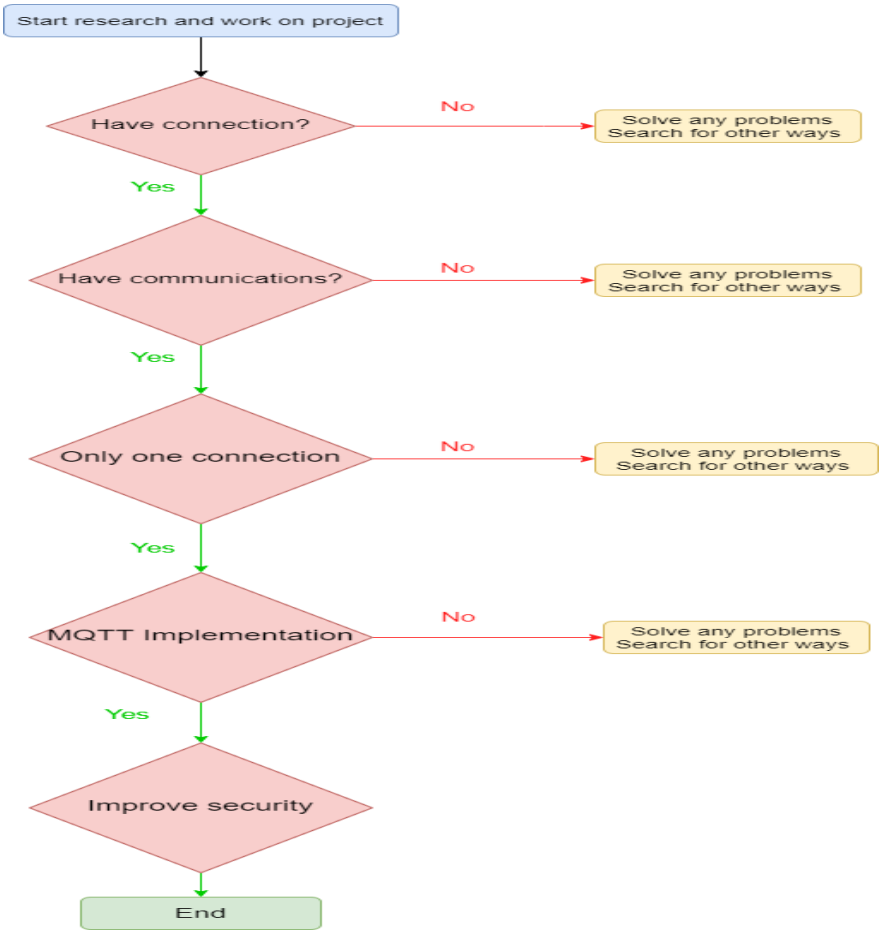
Later, both codes were developed more, so client was able to reply to server using some commands, and server would display a message that message was send. This was done by creating a switch case, where there were commands, checked by if statement so if the correct command was typed the client would send the data to server.

Then, the server should allow only one client to send messages to the server. This is done by saying to the server to only 'listen(1)' which is meaning that only one person is allowed to exchange data with the server into the connection. When another device/client tries to connect to server connection is made but the server doesn't respond to client until it finishes from its current conversation.

The final part of the project and the fourth fixture is a completely different from the other and is explained in detail later in this report

The development method was done based on the progress happened in each task of the project. The plan was to start a research before start doing coding, to find ways on how to create the code needed for each part. After one part was completed, then the other task come into place, if there was a problem and the work on a task wasn't completed, then further research was done, to solve the issue, or a change on the way of working to that part, so the task could be completed.

Below there is a flow chart that explain this better.



7.1 Implementation

As the time was passing there was a development on the project, of course there were some issues that in most of the cases they overcame. There were four main features set up in the start, all were met. There was another feature that could be implemented which was the ability to stop an attack, this feature wasn't completed, but there are some methods that can help in the improvement of security and give a guidance so it could be implemented in the future.

7.2 Basic implementation of features

As mentioned above four main features are presented into the project. Those four features are:

1) Make connection with client
2) Start conversation with client
3) Allow conversation between clients one at each time
4) MQTT implementation

The first feature is to make the connection, so server and client can be connected in order to communicate later. This feature works by having set up a socket, where the server waits the client to make a connection, the client have to knows the IP of the server and the IP is manually entered into the code. At this point, it is important to first run the server's code, so a connection session is in place, after this the client's code is executed, at this stage a connection is made. An appropriate message is displayed into both devices.

The second feature has to do with a communication between client and server. After the connection, clients can type some specific commands which are 'HELLO, 'REPEAT','BYE' and 'CLOSE'. Through these commands, the client sends messages to the server. This model works like a chat room, where the server knows that data sent from client. The way it operates, is easy to understand, each time when a command is typed, the client asks for another command, after the conversation is over, client can exist by typing the commands 'BYE, although the client is left, the connection remain open, when client type in 'CLOSE' the connection is closed, and the code from server have to be re-executed.

Third feature is about allowing only one client connected to have a conversation with server, this feature was implemented into the server's code, where the connection was first created, the server allows only one client to communicate. This was tested and it worked since when there was already a connection with the client, the code was run by another device, connection made but the other client couldn't receive respond from server. There is the ability to allow the data exchange to more clients, for example increasing clients communicating the same time to server from one to three.

Moving to the fourth feature, it's something entirely different from the previous coding done. The purpose before was to make a connection and send specific commands. Now with MQTT implementation, things are changing a lot, since now there is the application of MQTT, and the subscribe/publish model is adopted. There was necessary to do some installations to the Raspberry Pi, which are related to python and some MQTT libraries in order to create some topics to subscribe and built an MQTT server through Mosquitto in order to understand how things work in such as environment. The aim now is to see how to subscribe to an issue and how to create a file in the Raspberry to save passwords and usernames, but also how to send messages and understand how the communications are happening in real applications of Internet of Things, where small embedded devices or sensors are involved.

Taking things from the start, there are hundreds of ways to set up a broker and subscribe to a topic. The project is presenting two of those ways, the first one has to do with two codes written in Python, which are executed from the two devices. Basically, this is a straightforward way, since everything is done automatically when the codes are run, similar to what is happening in IoT products for Smart Home. The first code creates the MQTT broker and is waiting for data to arrive from a publisher. Subscriber is subscribed into a specific topic and publisher is publishing on this topic.

Even if this project was designed to accept one client, in this part more devices can be used, and subscribe to topics of interest, although only the devices within the internal network can be connected there, and those that are installed the Mosquitto packages and libraries needed.

The part described above, was a good way to let anyone understand what's happening into a system with MQTT configured and it was a feature of the project to learn how MQTT broker works and how data is processed. Although except the fact that only devices within the network can connect there wasn't any other security measure. Security is very important for many reasons, so password and username configuration where added. This was possible after editing the configuration files and blocking anonymous access, after this through the command line, password and username were created and saved to the configuration file. The configuration file is saved in the server/broker's device where a permission must be given in order to access it, which means that the file is secured. Furthermore,

when wrong password or username is entered, the connection can't be made, so the client can't subscribe to the topic, and a message is displayed saying that access denied. Moreover, if a client/ left, it continues to receive messages, and when it gets back online the data feed continues normally so, there is no need to re-subscribe to the topic. The mechanism of Mosquitto broker is also mentioning the size of the data send in bytes.

The implementation of MQTT wasn't direct to the initial codes used for the first two features, but with this way, the user understand better the way of operations and be able to learn how the devices communicate, also it can be said that the project is a simulation of a real IoT system, since the client can be assumed that is a device or a sensor. Furthermore, when there is the "publish'' of topic from the broker, any data sent is monitored. Adding to this the IP address of the broker was necessary to be known by client, in order to receive the data.

8.1 Evaluation and testing

Evaluation is one of the most important parts of any project, since it's the part that the testing is taking place and see if the requirements or features of the product are met. One markable aspect of the evaluation is the performance of the project, some measurements are done to see some important values regarding the speed and the time needed, each time a feature is executed. Take into consideration that the codes are run from the command line of the Raspberry Pi, the Raspberry Pis are not connected direct to a router, a WiFi hotspot is created from laptop, so because of this the internet speed may be affected.

8.2 Performance Analysis

The project was broke down into three parts:

1) first, second and third feature were evaluated together because, the two codes created for those parts was the same.
2) In the fourth feature which was the MQTT implementation was divided into two parts, the first one where a 'subscribe' made on one topic, here there are two python codes implementing some MQTT libraries and are run in the same way as the first part.
3) The second part of fourth feature, let the user do some programming and editing of files through the command line, the final part was evaluated differently, since the user had to enter manually some commands in order to set up a broker, subscribe to a topic of it, set up password and username and send data through the connection to the topic.

The values measured for the first part was the time in second(s) needed to run the code, have a connection and start a command conversation between the server Raspberry Pi and client Raspberry Pi.

Part 1 - Execution of two Python program through Raspberry Pi terminals-before MQTT implementation

Event	***Time(seconds)***
Run the server code	≈8
Time needed to create the connection	≈1.5-2
Run client code	≈8

Time needed for the client to connect	≈1
Time needed to write and send a message through commands	≈4
Time to send the data, and ask again for commands	≈1.5

Part 2 – Execution of two Python program through Raspberry Pi terminals after implementing MQTT, publisher/subscriber mode

For this part note that, the subscriber code is continuously running because there is loop, as this is simulating a real IoT system. The subscriber sends message to this topic one at a time when is executed, mentioning that data is sent, so after the subscriber sends a message, program must be re-executed to receive more messages.

Event	***Time(seconds)***
Run the server code	≈8
Run client code	≈8
Time needed to subscribe and receive data	≈5

Part 3 Execution of the commands in terminal, edit the configurations file, set up password and username, connect, subscribe and send data

Event	***Time(minutes)***
Time to install the necessary packages and libraries(eg. Mosquitto broker)	≈5
Time needed to execute the commands	≈10
Time needed to modify and save the configuration file of broker, to enable authentication	≈5
Time needed to set up password and username	≈0.5
Time needed to connect to broker through authentication and subscribe to a topic	≈1
Time needed to receive the data sent from publisher	immediately

The time mentioned in the tables above is an estimated average of someone that has a basic knowledge, about computer science/electronic engineering, someone that hasn't such knowledge may need the double time. The results illustrates that the operations of the project are done quick enough, except the last feature, the reason for this is because of the necessary operations that have to be done by the user, talking about the time needed for a subscriber to get the message, it can be said that is fast.

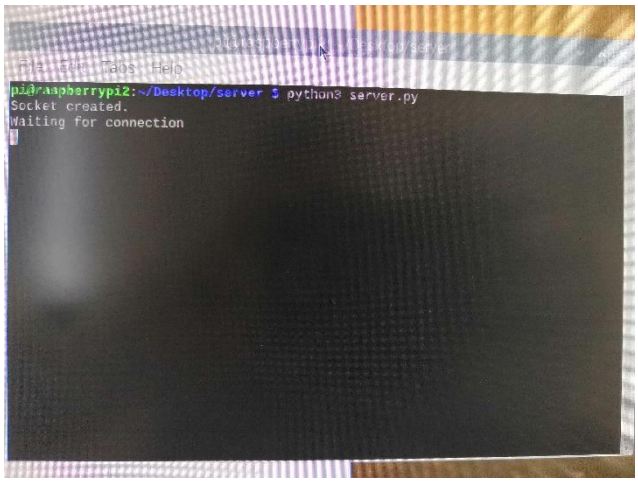
Asking the opinion from few colleagues that review the project, was easy to understand that the general idea is that the project works fine, although in some parts in the fourth feature about MQTT implementation, there may be some things that spoil the simplicity and confuse things a bit. Regarding the use of terminal window, to run the programs they said that there were familiar, and they didn't find any difficulty, although they mention that in the configuration of broker, easily a mistake could happen. People reviewing the project had background from computer science and told that it was interesting the way they learned more about IoT connections, data transfer and security. The instructions given about how to create a broker and configure it, were clear enough, although someone not having some experience would prefer a graphical user interface, this was a suggestion about the future. About the performance the feedback given was positive.

8.3 Testing

Testing is a must for evaluation as testing is used to check that codes are working correct and find any issues so they can be fixed to achieve the best possible outcome. Test were done throughout the project and were helpful to identify some problems in coding. In each part, when something was finished it was

tested to be sure that the task completed without any issues, so there would be a moving to the next task. Through the results a more accurate evaluation is shaped.

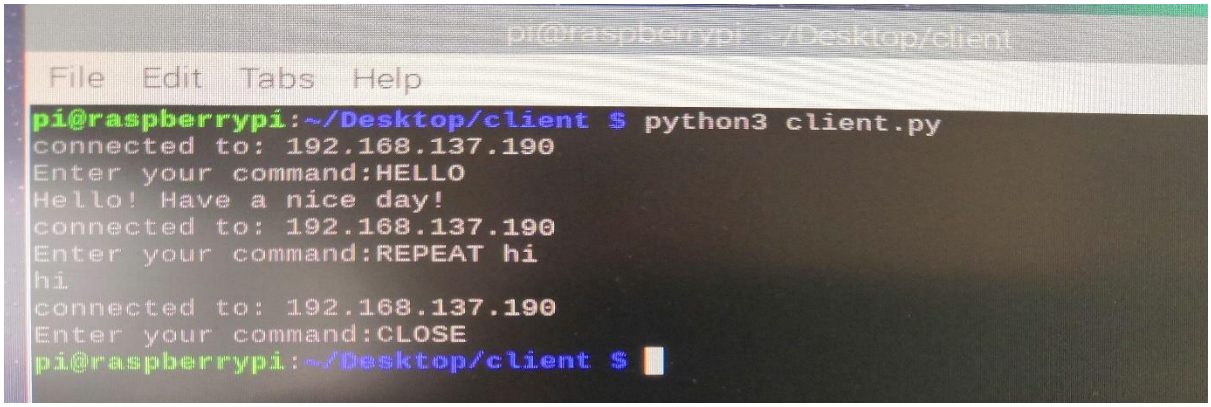
Through testing, some problems were discovered in some features, most of them were solved. In the first feature, server was creating the connection and was waiting for the client to connected, although the client couldn't be connected to the server, after searching about why this occurred, it has been found that this was because updates were missing from the client Raspberry Pi. Raspberry Pi updated and connection completed.



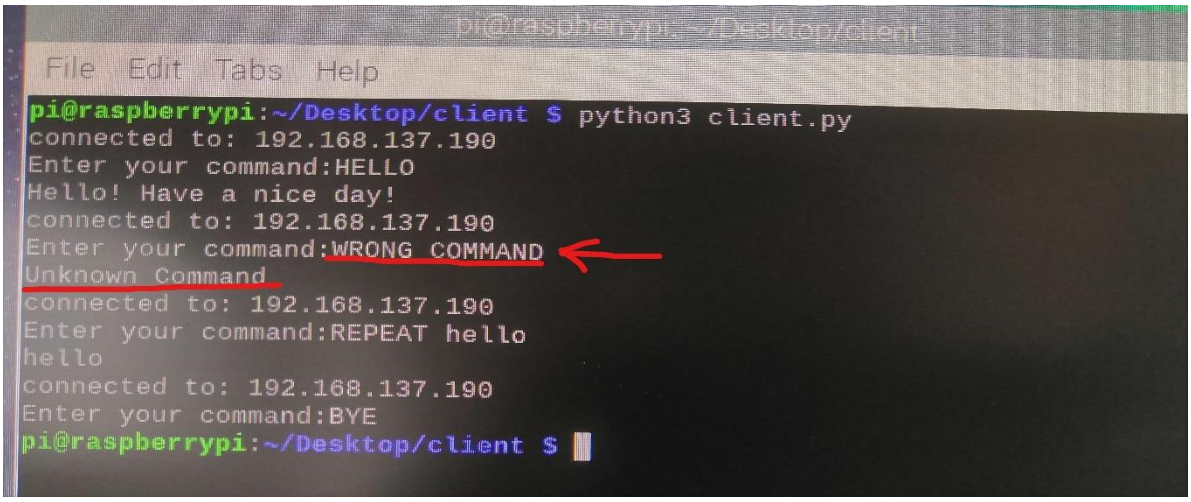
Source: Photo taken in project's testing shows the execution of code from server, as you can see server inform the user that socket is created and that is waiting for connection

Source: Photo taken in project's testing shows the IP address of the client when client connects

In the second feature a small detail was detected that could make the user believe that the program was having a bug. Because everything is done through the terminal of the Raspberry Pi is important to run the Python code in Python 3 because sometimes the system may be confused and run the code in Python 2, which will make the commands don't work and give the sense to the user that something is going wrong. To solve the issue the user should type 'python3 client.py', this detail is also mentioned and in the technical documentation. Commands tested and the functionality is working correct although the four commands must be entered in uppercase. Conversation is made, and when a wrong command is entered server asks again the user to enter the command. At the start of the connection server's IP address is displayed.

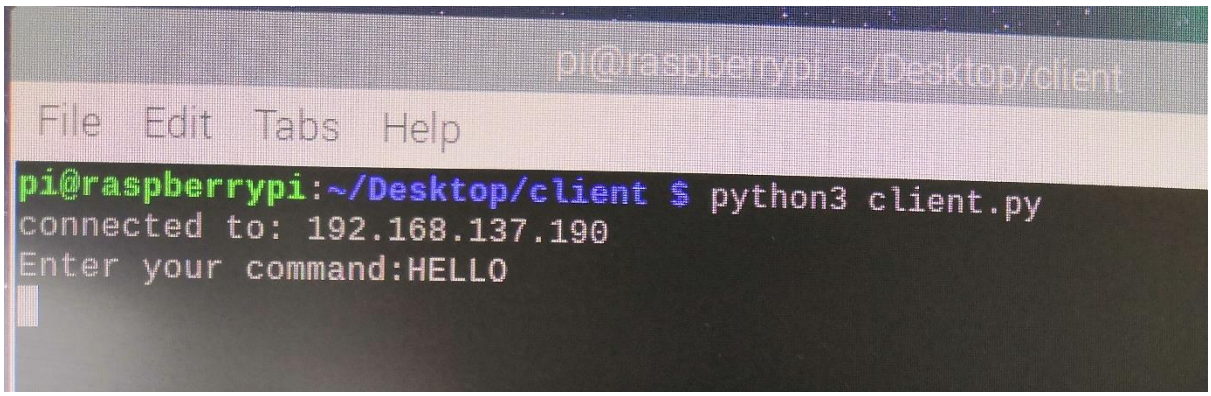


Source: Photo taken in project's testing shows a conversation and message exchange between client and server, photo taken after executing client code



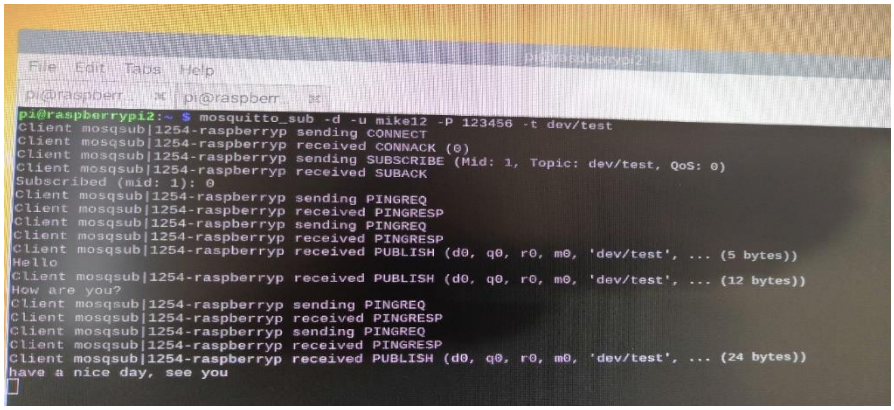
Source: Photo taken in project's testing shows that when a wrong command is entered an appropriate message is displayed and user can enter again the correct command

About the third feature, the main test done was about the number of devices that can communicate with the server. There was some complexity at the start and for this reason, a more suitable approach made. In this part the test was successful since when another device is connected to server, can send a command, but server don't receive the command and the conversation is stand by, until the other client 'talking' with server is left.

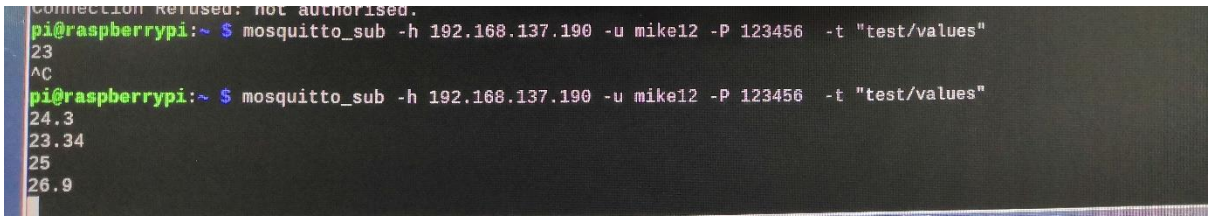


Source: Photo taken in project's testing shows that even if the connection made, a conversation can't be continue, until the other client is left from the connection. Until the other client is left the client is on standby. To do this the same code was run first from a laptop and then from the Raspberry Pi.

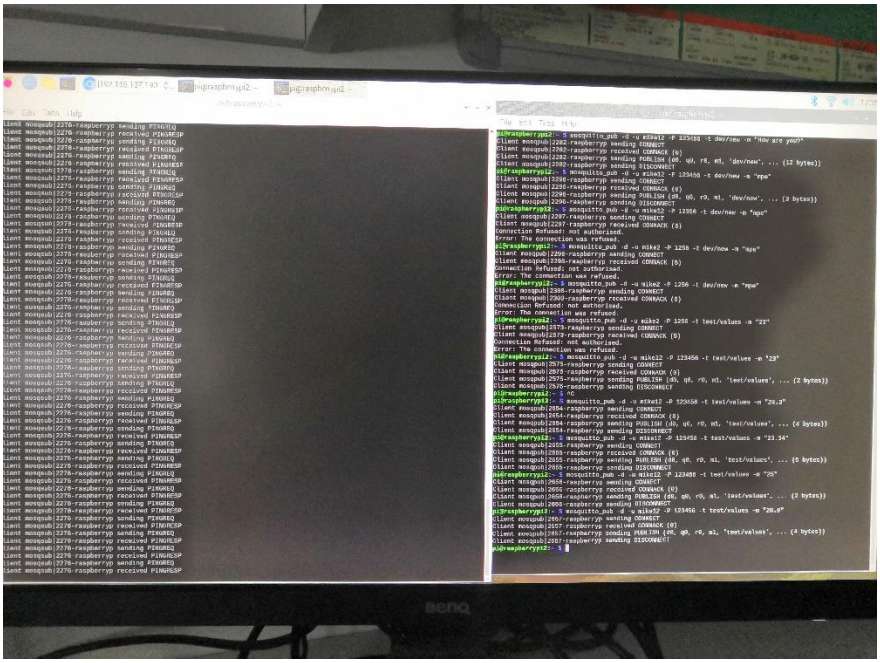
In the fourth and last feature of the project, more testing done, because it was important to see that the Mosquitto broker was correct installed and configure. Furthermore, the order in which the commands must be executed by the user are very crucial. In the implementation of MQTT, a guide is followed by the user, so it was necessary to test this part many times to be convinced that everything was working appropriate. Moreover, it was critical to see that indeed the authentication with password and username works, so no other could have access to the connection. Broker was tested about the continuous data flow, so it wouldn't matter if the subscriber gets offline, data was there and when client wanted could access them again providing the username and the password. The results show that access to the topic was denied because of wrong authentication, in other words the security measure worked. Talking about the two python programs, testing was done to see, if the data is sent for a specific topic is received to the client that subscribed to that topic. When the topic changes, the message was sent, but the client didn't receive the data, which means that the testing was successful. Another small problem detected is that because of the 'paho.mqtt' library the two codes have to been run as 'python filename.py', in contrast with feature two that the codes have to be executed in python3.



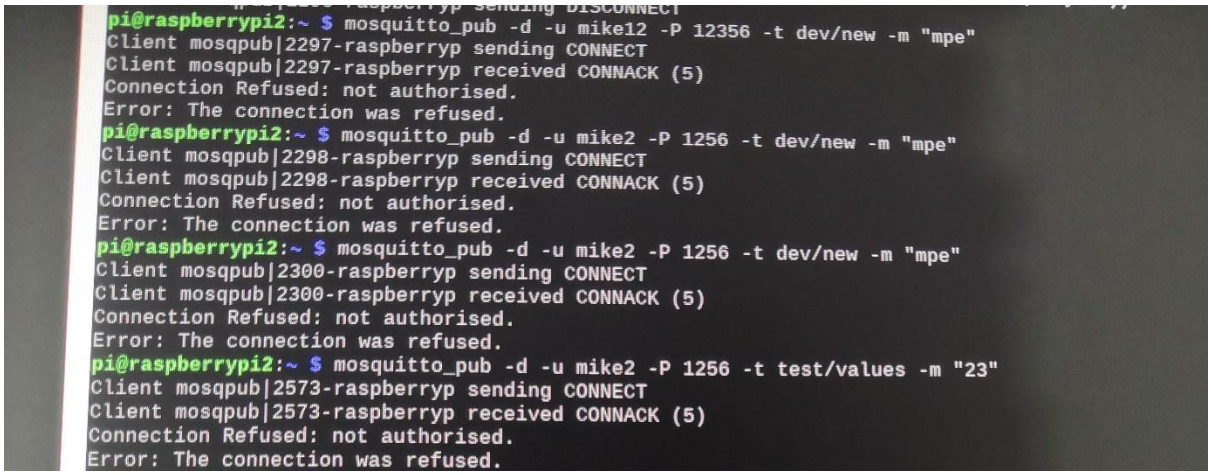
Source: Photo taken in project's testing shows the messages received



Source: Photo taken in project's testing shows the messages received

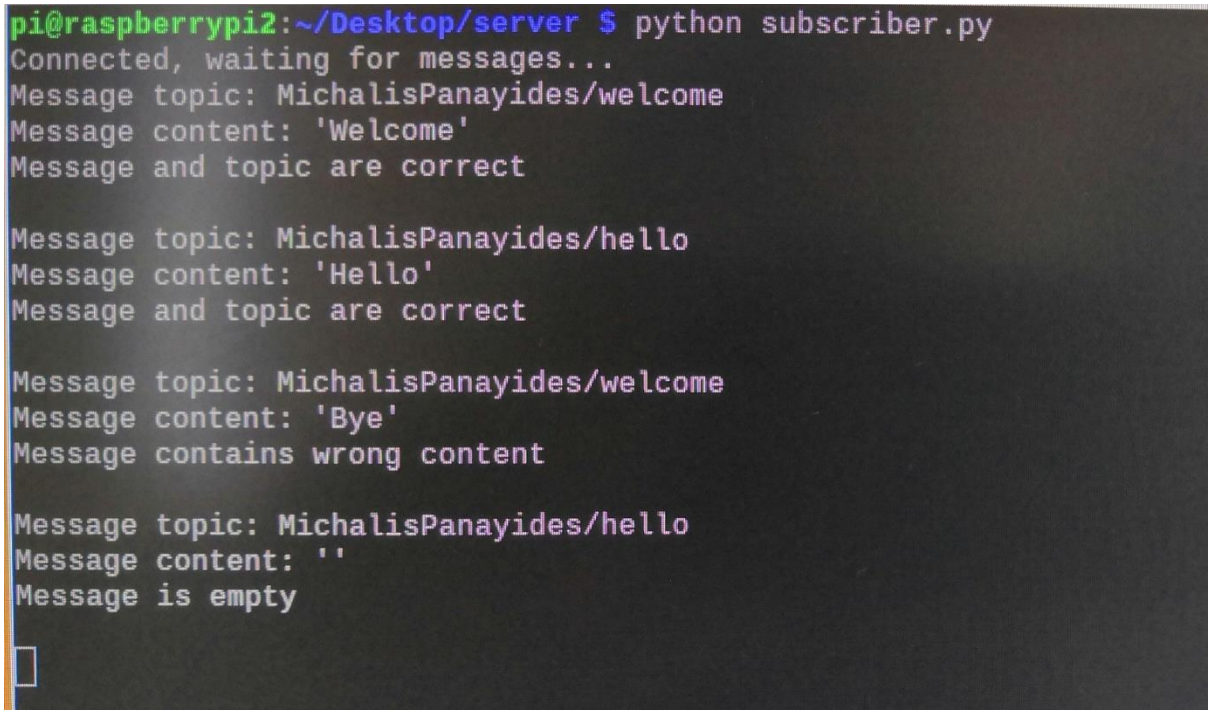


Source: Photo taken in project's testing shows the continuous data flow

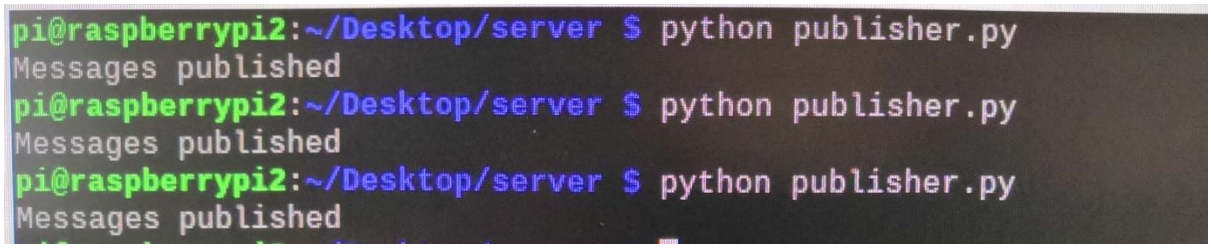


Source: Photo taken in project's testing shows that no access is authorised when username and/or password are wrong. As the photo shows connection is refused.

Regarding the two programs that must be run from terminal, the client supposes to receive, all messages that are subscribed to the topics. Client is filtering the messages and gives out information about the data received. If the data received is containing the correct message in the correct topic a message is showed so the user knows that the published sent the correct data to the correct topic, if not a message is printed that notifies the user about the mistakes he did.



Source: Photo taken in project's testing shows the messages received from subscriber and also details about the messages that are displayed.



Source: Photo taken in project's testing shows publisher sending messages

Another feature that could be implemented, had to do with SSL and security. This feature wasn't from the main ones although it would enhance more the project. This feature would ask for an SSL certificate before letting the TCP/IP connection happen, although because of the lack of the necessary resources, it wasn't possible to implement the SSL, as the project would be escape from securing connections and data. In order for this to be implemented the data had to be uploaded to a cloud where a domain would be available, in order to have a certificate. In other words, the way the project operates would be much different. At this stage the purpose is to simulate an IoT system, in future the project can be modified to include SSL configuration.

8.4 Results

After doing the testing, it was time to analyse the results to see what improvement could be done in future and were there are any issues. The tests showed that the implementations done in different parts of the project were done correctly, since the expected outcome was met. In some parts some

configurations could be done better, like the implementation of MQTT where the communication started manually. In general, the results were satisfactory and gave the impression that most parts of the project were working well.

9. Conclusion

Internet of Things is one of the most fast-growing technology in computer science, that offers a lot of things to the humanity. There are millions of products that are related to IoT and making simple things becoming smart. The project focuses on the many connections that are happening between the devices that are working in an Internet of Things system, and the way data is exchange between those devices. This technology and machine-to-machine communications turned out to be revolutionary and effective, something that led to a magnificent progress through the years. To understand the massive impact IoT brought we can go twenty years back, where the smart watches, smart bulbs and smart cities were a science fiction scenario and products we saw only in fantasy movies. Nowadays, this is the reality. Internet of Things makes the human life, easier and smarter.

Even if there are several benefits, in the same time there is and a number of drawbacks that IoT brings. The largest one, is the lack of security and privacy. Through this research the concerns about security are presented as and the ways there are to minimize the risks. The research suggests to use a dedicated network that is separated from the rest network, where it is possible, since it reduces the chance for anyone breaking into the network, not be able to gain every data passing through. The implementation of MQTT broker and the authentication of password and username acted as a shield to security. This protocol applied into the network used for the communications between the devices and is explained in feature four. Furthermore, the broker mechanisms, through the testing and results proven to be working which made the project be more secured. Talking about the connections, project illustrates a simple way to communicate running separate codes from two different devices in feature one and two. The use of the commands made the connections be more controlled, since the client have to know the commands in order to send messages to the server. Moreover, the third feature limited the number of clients that can access the server the same time to one, this was done as a security measure. An objective was to create a simulated Internet of Things system, which allows the user to understand how connection are made and how data is sent, something which was met according to the findings.

Using the performance analysis data, it can be said that the performance of the project is satisfy. The time needed to execute the different parts is within the logical limits, although in the fourth feature, where the user have to set up the broker, there is space for improvements. Probably some people may find it little difficult or not have the necessary knowledge to complete this task, but the aim is to help people understand the operations presented and make them learn more about these methods, that are applying in real IoT systems. Somewhere in the project there may be some parts that are easier than others and some that are seem a little confusing this is because, of the way a broker works. It may seems difficult at first although if instructions are followed there will not be an issue.

There are many approaches on how to make connections and data exchange through IoT devices. Some protocols and mechanisms have been selected to be implemented into the system and are explained in detail but, many information about the other methods is mentioned in the research as so and many ways how security could be improved. Results showed that the authentication applied helped to improve

security. Initially SSL was to be implemented, but it decided not to, because it would make the project fell of the initial purpose which was to simulate connection and data transfer happening in an IoT system. Furthermore, more resources were needed that wasn't available, for instance a domain name to create a certificate.

Project is giving a lot of information about Internet of Things and is giving the necessary knowledge to someone that wants to start exploring more and start experimenting, for this reason project is important as it provides the data needed to start understanding better the operation of IoT. Work done in this project can be consider in a future research from anyone. The research explains how Internet of Things works, but not only this, the development of IoT, the ways someone can build a system, the security issues, the positives and negatives, ways which help in shielding a system, the many protocols that can be used in a system, information about connections and more are presented in the research. This research can be used from someone that wants to learn more about Internet of Things or wants to have a first touch with the topic. Adding to this, as said before some parts of the project can be developed more in future, for example the MQTT broker implementation can be modified to do more stuff more automatously. The project is focusing into server/client mode or subscribe/publish, and only one client can be connected to the server/broker, because of MQTT mechanism the project can be edited and expanded into more clients. Codes are executed from terminal windows of Raspberry Pi, in future a graphical user interface can be designed to make the project more attractive and easier to use.

Taking all into account, the project is presenting all the fundamental information about Internet of Things. The implementation of features was successful, and they achieved their expected outcome. A user running the codes and following the instructions will enrich his knowledge in this field. User will learn more about the connections and data exchange happing in an IoT system. Issues of security are presented but also and ways to improve these issues. The objective was largely achieved, the user learns how to create and handle the connections occurring in system as so and the data traffic. Importance of security is understood as well as the ways to strengthen it.

10. References

[1] Andrew Meola, "A look at examples of IoT devices and their business applications in 2020" 18 December 2019 [Online]. Available: https://www.businessinsider.com/internet-of-things-devices-examples?r=US&IR=T [Accessed: March 2020].

[2] A. McEwen and H. Cassimally, *Designing the Internet of Things*. Chichester: J. Wiley and Sons, 2014, pp. 9

[3] Tommy Quek, " The advantages and disadvantages of Internet Of Things (IoT)" 14 February 2017 [Online]. Available: https://www.linkedin.com/pulse/advantages-disadvantages-internet-things-iot-tommy-quek/. [Accessed: March 2020].

[4] Louis Columbus, "10 Charts That Will Challenge Your Perspective Of IoT's Growth" 6 June 2018 [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2018/06/06/10-charts-that-will-challenge-your-perspective-of-iots-growth/#3674b74b3ecc. [Accessed: March 2020].

[5]Arne Holst, "Spending on Internet of Things worldwide by vertical in 2015 and 2020" January 2017 [Online]. Available: https://www.statista.com/statistics/666864/iot-spending-by-vertical-worldwide/ [Accessed: March 2020].

[6,7] i-scoop.eu, "Business guide to Industrial IoT (Industrial Internet of Things) "[Online] Available: https://www.i-scoop.eu/internet-of-things-guide/industrial-internet-things-iiot-saving-costs-innovation/[Accessed: March 2020].

[8,9]Sogeti Labs, "9 graphs on how businesses are using the Internet of Things" 17 March 2015 [Online] Available: https://labs.sogeti.com/9-graphs-on-how-businesses-are-using-the-internet-of-things/ [Accessed: April 2020].

[10,13]RedAlkemi, "Pros&Cons of Internet of Things" 7 May 2018 [Online] Available: https://www.redalkemi.com/blog/post/pros-cons-of-internet-of-things [Accessed: 8 April 2020].

[11,12]Varun Bhagat, "What are the Pros and Cons of Internet of Things?" 23 June 2019 [Online] Available: https://www.pixelcrayons.com/blog/what-are-pros-and-cons-of-internet-of-things/ [Accessed: April 2020].

[14] Bob Violino, "7 steps to enhance IoT security" 25 June 2019 [Online] Available: https://www.networkworld.com/article/3404198/7-things-you-can-do-to-enhance-iot-security.html [Accessed: April 2020].

[15]Macy Bayern, "How to improve enterprise IoT security: 5 tips" 14 November 2018 [Online] Available: https://www.techrepublic.com/article/how-to-improve-enterprise-iot-security-5-tips/ [Accessed: April 2020].

[16]Naveen Joshi, "Three ways to improve IoT security" 19 August 2019 [Online] Available: https://www.allerin.com/blog/three-ways-to-improve-iot-security [Accessed: April 2020].

[17] MQTT.org, "Frequently Asked Questions" [Online] Available: http://mqtt.org/faq [Accessed: April 2020]

[18] Toby Jaffey, Eclipse Foundation "MQTT and CoAP, IoT Protocols" [Online] Available: https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php. [Accessed: 10 April 2020].

[19] Francesco Azzola, "CoAP Protocol: Step by Step" 8 November 2018 [Online] Available: https://dzone.com/articles/coap-protocol-step-by-step-guide [Accessed: April 2020]

[20] "What is REST" [Online] Available: https://restfulapi.net/ [Accessed: April 2020]

[21]CoAP.technology, RFC 7252 Constrained Application Protocol [Online] Available: https://coap.technology/ [Accessed: April 2020].

[22,23] Toby Jaffey, Eclipse Foundation "MQTT and CoAP, IoT Protocols" [Online]. Available: https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php. [Accessed: April 2020].

[24] HiveMQ Team, " Persistent Session and Queuing Messages - MQTT Essentials: Part 7" 23 February 2015 [Online] Available:https://www.hivemq.com/blog/mqtt-essentials-part-7-persistent-session-queuing-messages/ [Accessed: April 2020]

[25] MQTT.org, " MQTT.org" 3 April 2019 [Online] Available: http://mqtt.org/ [Accessed: April 2020]

[26] Mosquitto.org, " Eclipse Mosquitto TM- AN open source MQTT broker" [Online] Available: https://mosquitto.org/ [Accessed: April 2020]

[27] raspberry.org, " Raspberry Pi 3 Model B+" [Online] Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/ [Accessed: April 2020]

[28] python.org, " Python " [Online] Available: https://www.python.org/ [Accessed: April 2020]

[29] Atlasian Jira Software " The #1 software development tool used by agile teams" [Online] Available: https://www.atlassian.com/software/jira [Accessed: April 2020]

[30] Marina Ch, "Agile Application Development Methodology: How We Ensure App Quality" 5 February 2017 [Online] Available: https://www.cleveroad.com/blog/what-is-agile-methodology-for-mobile-development [Accessed: April 2020]

[31] Collabnet, "What Is Kanban? An Introduction to Kanban Methodology" [Online] Available: https://resources.collab.net/agile-101/what-is-kanban [Accessed: April 2020]