Arttu Niemi

Property surveillance system

Metropolia University of Applied Sciences

Bachelor of Engineering

Information and Communication Technology

Thesis

26 April 2018

Author Title Number of Pages Date	Arttu Niemi Property surveillance system 32 pages 26 April 2018
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Communication Networks and Applications
Instructor	Heikki Rahkonen, Project Engineer

This thesis was about designing and creating a surveillance system. The system would be able to capture and provide live footage and temperature and humidity data. In addition to that there would be remote controlled power sockets. All of this could be accessed and used remotely from a web page.

The main purpose of the system was to function as a security camera and to be used when property was left for longer periods of time. The added sensors would provide useful information of the estate and the power sockets could be used to control electric appliances such as heaters and lights. These features make the system versatile and applicable for other surveillance and maintenance purposes aside from the estate.

Other goal for the project was to work with minimum budget while creating a reliable system with quality components and legal software. This was achieved by using open source software and devices created for consumers.

The completion of the project took approximately three months and all the initial features were implemented. Addition to this, some features that came up during the testing were also implemented.

Keywords	surveillance, remote, camera, sensor

Tekijä Otsikko Sivumäärä Aika	Arttu Niemi Kiinteistön turvajärjestelmä 32 sivua 26.4.2018
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Communication Networks and Applications
Ohjaaja	Projekti-insinööri Heikki Rahkonen

Insinöörityön tarkoituksena oli suunnitella ja toteuttaa monipuolinen valvontajärjestelmä asuinkiinteistöön. Järjestelmän tuli pystyä tallentamaan ja näyttämään videokuvaa sekä lämpötila- ja ilmankosteustietoja. Tämän lisäksi järjestelmään tuli sisällyttää kauko-ohjattavia pistorasioita. Kaiken tuli olla nähtävissä ja käytettävissä itse luodulla ja ylläpidetyllä internetsivulla.

Työssä tehtyä valvontajärjestelmää käytetään kiinteistön valvontaan sen jäädessä tyhjäksi. Kamerat tuottavat videokuvatiedostoja verkkosivulla katsottavaksi. Lisäksi sensorit tarjoavat hyödyllistä informaatiota kiinteistön lämpötilasta ja ilmankosteudesta. Kauko-ohjattavilla pistorasioilla voitaisiin internetsivun kautta kontrolloida erilaisia sähkölaitteita, esimerkiksi sähköpattereita ja valoja. Nämä ominaisuudet tekevät järjestelmästä monipuolisen ja mahdollistavat käytön myös muissa valvonnan ja ylläpidon tarpeissa, esimerkiksi varastotiloissa.

Yhtenä tavoitteena oli myös suunnitella ja toteuttaa järjestelmä mahdollisimman pienellä budjetilla, mutta kuitenkin niin, että saataisiin aikaan luotettava järjestelmä, jossa on laadukkaat komponentit ja lailliset ohjelmistot. Tämä saavutettiin avoimen lähdekoodin ohjelmistoilla ja kuluttajille suunnitelluilla laitteilla.

Projektin kesto oli suunnilleen kolme kuukautta, ja järjestelmä saatiin valmiiksi. Kaikki alustavat ominaisuudet saatiin toteutettua, ja niiden lisäksi joitain testauksen aikana ilmenneistä ideoista saatiin myös lisättyä, esimerkiksi ajastustoiminto pistorasioille.

Avainsanat	valvonta, etähallinta, sensori, kamera

Metropolia

Contents

Metropolia

List of Abbreviations

DNS	Domain name system. A system that changes computer's host name to IP address.
IP	Internet protocol. Communications protocol used to identify the address to forward data.
WWW	World wide web. Space with documents and other web resources that can be accessed with internet connection.
LAN	Local area network. A network of devices that connects within a limited area
WAN	Wide area network. A network of devices that connects within a large area
WLAN	Wireless local area network. A network of devices that connects wirelessly within a limited area.
USB	Universal serial bus. Standard that defines connectors, cables, communication, and protocols between devices
GUI	Graphical user interface. Graphical icon and visual based interacting method on a computer
CLI	Command-line interface. Text based interacting method on a computer
HTTP	Hypertext transfer protocol. A protocol that is used to receive and send files and webpages on the internet.
HTTPS	HTTP secure. Protocol for more secure internet usage.

HTML	Hypertext markup language. Standardized language for creating web content.
RF	Radio frequency. Electromagnetic wave that is between 3kHz and 300 GHz
SSH	Secure shell. A network protocol that can be used to operate network services securely on a unsecure network.
GPIO	General-purpose input/output. A pin on a computer board.
NAT	Network address translation. A way to preserve or hide IP addresses.
Python	A popular and widely used programming language.
JSON	JavaScript Object Notation. Easy to read and easy to compute file format.
PHP	Hypertext preprocessor. A scripting language mainly used for web development.
SQL	Structured query language. Language designed for data managing used in programming.

1 Introduction

My parents live in a rural area which over the years has been safe and peaceful location. Regardless of this, every time when they leave the house for longer periods of time they ask someone from the family or friends to be a house guard and check it regularly. Hearing about their neighbor getting a video surveillance system gave the idea for the project. A surveillance system that provides live video and temperature and humidity data over the internet could serve as a house guard.

The system would have to have at least two cameras, one for each door. The cameras would be used to provide footage of the possible intruder when detecting motion. Then there would have to be a temperature and a humidity sensor. The temperature sensor would tell the room temperature and if there is a need for heating. The humidity sensor would be most useful when the security system would be used in a storage or a boiler room when not used to secure the main building in absence of the residents. There would also have to be email notifications that would inform the user regarding the detected movement or sudden changes in temperature or humidity.

To use the backup heating consisting of electric radiators there would also have to be outlets that could be controlled remotely and used with timers. The outlets could also be used to control other electrical appliances to create an illusion of presence. This would be achieved by switching on and off the radios and a TV, for example.

All the devices would have to have a centralized controlling unit, a server. Keeping most of the configuration on the same location would help with the work and after completion with the deployment. The data and video footage would have to be available from anywhere with computers and mobile devices with internet connection and from a centralized location. This would help creating a simple and a user-friendly access and controls.

More accessibility creates more possible security threats, so the connection would have to be secured. Also, the access to the system would only be granted with correct user name and password. Implementing the security measures would be carried out while keeping the system user friendly easily accessible for the authorized users.

The project would be started with the initially set goals and requirements. One additional self-set requirement for the security system would be to make it expandable and scalable. This is to create more versatile and longer lasting system that could have new features added more easily in the feature if a need arises. This would be achieved by choosing easily scalable options and writing the documentation.

2 Surveillance

Surveillance itself can mean several different things and can be defined as keeping track of changes, actions, and behavior. The surveillance can be conducted on people, machines, and immaterial objects, to mention a few. The surveillance can be performed with the help of electronic appliances, processes, and human instincts. [1.] An example cases of surveillance could be security cameras in the bank to help discouraging crime or to help catch the criminals in case of need. Other example of a different case of surveillance could be a weather station. The station could be collecting data with temperature and humidity sensors, hydrometer and anemometer. This data could then be used for statistics and forecasts. Temperature and humidity data could also be collected indoors and used to follow growth conditions on a greenhouse or preserve goods on a warehouse for example.

Surveillance conducted by the government and law enforcers has raised an ethical dilemma about individual's privacy. The discussion has been on about which level of surveillance and in which cases it is acceptable. A high amount of surveillance can be seen as a human rights violation. As the people are being watched and listened just in case they might be committing crimes or there is a high change of that they might be. The other side of the situation is when enough surveillance is not conducted in cases when there is a high chance of crime. An example case can be a situation where a person is known to have ties to terrorist organization. [2.]

3 Commercial surveillance systems

On the market there is many different options for conducting surveillance. Different kind of sensors and cameras can be found on the stores as well as weather stations. Some companies offer services and device bundles for the surveillance. An example company is Sector Alarm which's starting packet can be seen on the figure 1. This packet contains motion ja temperature sensors. Then there is a remote controllable outlet and a smoke detector. All of this is controlled with a server and the data is viewed from a phone application that can also be used to control the outlet. As an extra the customer can also buy alarm service which contains regular check-ups made by guards. [3.]

Figure 1. Sector alarm’s starting package with sensors and the server. [4.]

When investigating different options offered by security companies a one similarity could be seen. Most of the companies did not offer cameras for their basic or starting packages. This was most likely due to the alarm services they offer also. Since if the guards get to the property fast enough after the sensors report an intruder, the video footage is not necessarily needed.

Figure 2. Netgear Arlo Pro 2 surveillance system. [5.]

On the bundles sold by electronic stores the situation is a bit different from the ones sold by security companies. Most of the basic surveillance systems sold by stores contain only the cameras, the server and the application to access the footage. An example package from Netgear can be seen on the figure 2. This difference is most likely since the stores do not offer the guard services. In these cases, the video footage will help more on identifying the intruder.

The main reason for starting this project instead of acquiring a commercial system was the price. While conducting the research and working on the project. Out of the more known security companies in Finland the Sector Alarm offered one of cheapest solutions containing the required equipment and functions. The starting price for the system is 149€ and it contains a camera, the server and a remote-controlled outlet. In addition, and as seen as extra, the package also contains a smoke detector, a door sensor and a control pad with a key tag. However, the price would raise because the package only contains one camera and one outlet when the need was for at least two of each. Each additional camera would cost 199€ and a packet of three outlets would be 89€ making the total cost of 437€. After the starting price there's also a monthly fee of 39,90€ for using the devices and the guard services. [3.]

As a reference the price for the self-created security system was approximately 100€. The self-made did not include some of the services and devices that were on the commercial system. Instead it contained just the needed functions and devices that helped to bring the price down. In addition, the self-made system could also be freely relocated and would not yield any monthly fees.

4 Hardware

This chapter will focus on the introduction of the hardware used on the project. It will also include the thinking process and reasons for all the choices made along the way.

Since the project was self-funded and thus the budget strict, some of the equipment were used or repurposed and the devices bought were usually the bare minimum. However, this did not mean that the surveillance system would be created unstable and unscalable as a system. Since the goal was a functional, trustworthy, and versatile system that would left room for possible development in future also. So, this was kept in mind while acquiring the hardware.

The hardware that was acquired for the project were bought all around the world. The more crucial and more expensive hardware such as the server was brought from the Europe because of the better warranty and faster delivery. Other hardware with more simple mechanics were bought from Asian webstores. This was due to lower prices and less crucial parts could be reordered if turned out to be unfunctional.

4.1 Server

The surveillance system required a computer to work as a server and a centralized point for the other hardware that would be connected to it. To reduce the effort of moving and installing the system to the wanted locations the computer had fulfill certain criteria. It had to be small sized and relatively quiet. Other requirement was a built in WLAN adapter or support for an external adapter.

The computer also had to have at least two but preferably more USB ports for other hardware. Mainly for the camera and the WLAN adapter but also for a mouse and keyboard for the beginning of the configuration.

The components inside the computer should also be able to process videos. Also, most of the hardware connected to the computer would take the power from it so it should be capable of that.

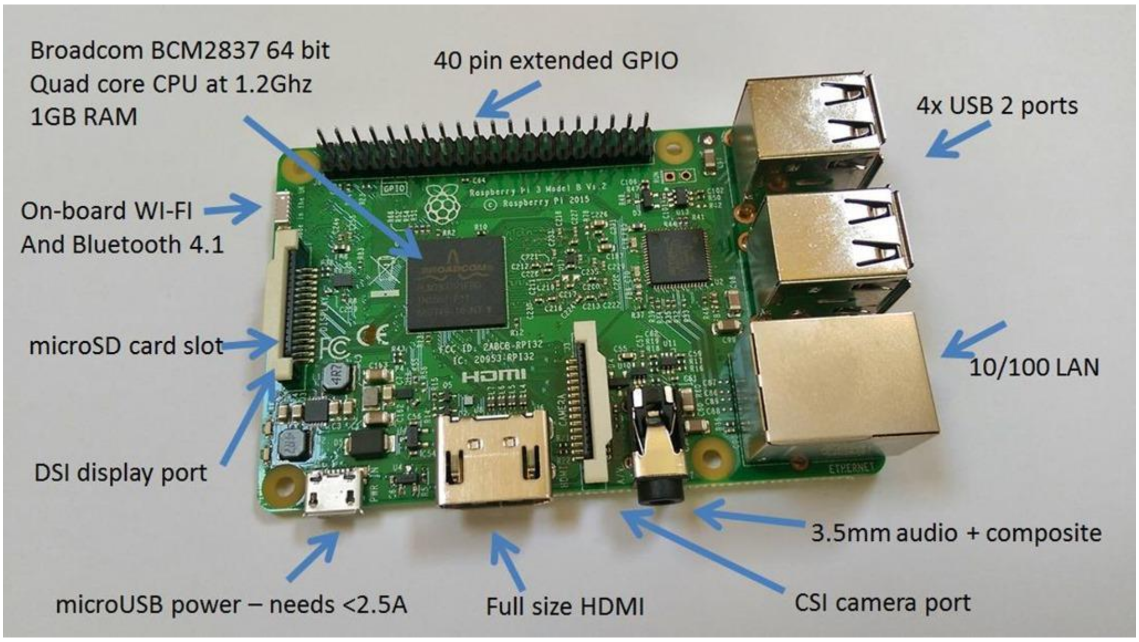

Figure 3. Raspberry Pi 3 Model B with specifications. [6.]

Raspberry Pi 3 Model B was chosen for the project because it could answer to most of the demands stated earlier. It was small sized and equipped with passive cooling to minimize the noise. Also, it had the USB ports for other devices and the GPIO pins to connect more devices and sensors. Raspberry Pi was also moderately priced which helped working with the minimal budget. The device and specifications can be seen on the figure 3. [6.]

4.2 Temperature and humidity sensor

Measuring the temperature was an important feature to make sure that the cold sensitive electronics or plumbing would not be broken because of falling temperature. Especially on winter time in the case of failing heating system or broken window the temperature could decrease rapidly. The humidity data would prove useful in the event of water damage, but this feature was aimed more for the surveillance of boiler room or storage were such incident would be more likely.

Figure 4. DHT22 temperature and humidity sensor [7.]

Finding a device that would measure both temperature and humidity and be compatible with the server was not an easy task. Most of the devices that could do both were weather station that could not be connected to the server in any way or were on a higher price point. In the end the chosen sensor on the figure 4 for the project was DHT22.

The DHT22 could be connected to the server through the GPIO pins. It could take both measurements and since it required a bit more configuration it was one of the cheapest options for the project. Other even cheaper option was the DHT11 sensor which functioned similarly to the DHT22. The key difference was the accuracy of the readings that favored the DHT22. The other important technical details are as follows.

- Low cost
- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 80°C temperature readings ±0.5°C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)
- Body size 27mm x 59mm x 13.5mm (1.05" x 2.32" x 0.53")
- 3 pins, 0.1" spacing
- Weight (just the DHT22): 2.4g [8.]

4.3 Camera

Being able to provide live or saved footage was one of the most important aspects of the surveillance system. The camera had to provide clear enough picture and be easily and surely placed to point at wanted direction. Also, there was a need to monitor a door on the other side of the building so at least one of the cameras had to be able to connect wirelessly. After the consideration and while keeping the cost in mind it was decided that one of the cameras would be a USB camera and other one would be an IP camera.

Figure 5. Fuj:tech SC-618 USB camera. [9.]

The USB camera chosen for the project was Fuj:tech's SC-618 as can be seen from the figure 5. This was because it was moderately priced and could provide video with maximum resolution of 1280 x 720 pixels. While the physical qualities were 1.4 meters long cord and adjustable stand which both would help to place it securely on most surfaces. The camera also had a microphone which was not used on the project but was seen as positive feature for the future development. [9.]

For the IP camera there was more choices and more topics to consider while making the decision. The first considered and tested option for the position was Fuj:tech's Drop HD IP camera seen on the figure 6.

Figure 6. Fuj:tech HD Drop IP camera. [10.]

The camera had the minimum requirement resolution of 1280 x 720 pixels and other useful features. Such as microphone to record audio on the site which could be useful if the surveilled property had to be monitored for abnormal noise. The camera also had an adjustable stand for better positioning and angles. [10.]

Because the camera possessed all the required features and being the cheapest option on the market. Fuj:tech's HD Drop was first considered for the project and was acquired for more testing. However, it turned out not to be compatible with the chosen camera management program Motion. This lead to consideration of options of changing the Motion for something else or getting a new IP camera.

After research and consideration, it was decided that an obsolete android smart phone would be used as an IP camera. This was to minimize the risk of purchase of other unsupported camera and if the smart phone would turn out to be not supported then it would not yield any financial losses.

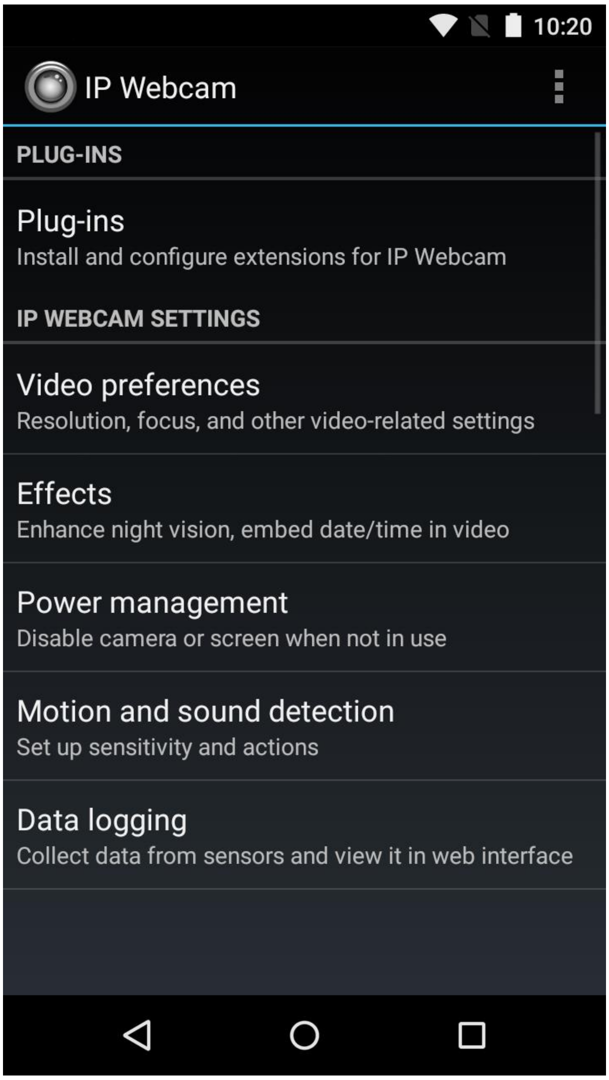

Figure 7. Main configuration page of IP Webcam

On the Google Play there was a free-to-use application called IP Webcam created by Pavel Khlebovich. The Application could be used to turn an android smart phone to an IP camera. With simple configurations it could stream video using the camera on smart phone and that stream could then be captured and configured on Motion. The application also had the option to utilize other sensors and features on the smart phone as can be seen from the figure 7. Such as microphone, speaker, and proximity sensor. Although these were not used, they could provide more ways if the project was to be further developed. One notable disadvantage of using the phone was the absence of the stand. This would require some creativity when deploying the camera on the site. But two prototypes were made by using magnets and a car phone holder for the other. [11.]

4.4 Outlets

One of the requirements for the project was a way to control outlets. This feature would be used to provide light for the cameras, use radiators as the backup heating system, or use other electrical appliances to create the illusion of presence on the house. These outlets would then be controller with buttons on the website hosted with the server. Also in cases when the need is recurring the outlets could also be timed.

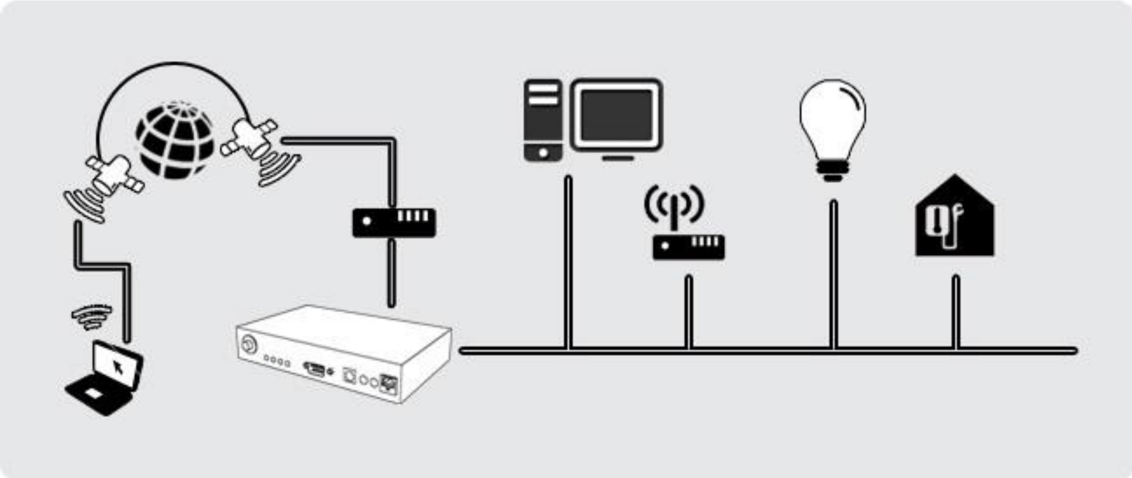

Figure 8. Topology of an IP controlled outlet. [12.]

While conducting the research for this feature two options were taken into more deeper consideration. IP controlled outlet that could be used through graphical interface with the internet connection as seen on the figure 8, in the similar fashion as the IP camera. And a RF controlled outlet that could be controlled with a limited range RF transmitter.

IP outlet would have required less work to be implemented into the project, but since the price turned out to be too high. And since the RF controlled outlet was ten times cheaper than the IP controlled variant it was initially thought for the project. Although after more research and tests on the matter, the RF controlled option, as it was, was discarded due to compatibility and reliability issues.

Figure 9. Energenie outlet (EU). [13.]

The price of IP controlled outlets kept the research for working RF option on. While looking for the Raspberry Pi’s own devices the Pi-mote control board along with the Energenie outlets seen on the figure 9 were chosen since they provided more reliable and compatible option. The Energenie outlet was a grounded and could handle maximum of 2500 watts and 10 amperes at a time which is enough for the thought electrical appliances. The outlets did not have any information about the protection marking so for the project it was treated as being 0. [13.] This meant that the outlet did not have any protection from water or dust, thus limiting the placing of the outlet. [14.]

Figure 10. Remote controlled outlet PO5505. [15.]

This cut down the versatility a bit since the initial thought was to use different outlets. The thought outlet would have been PO5505 seen on the figure 10. The PO5505 had the protection marking of 44. This meant that the outlet would have protection from particles bigger than 1mm and from splashing water. [9.] The maximum load would have also been bigger, up to 3600 watts and 16 amperes but this was not a significant lost since the Enegrenie already had plenty. [14.]

4.5 RF transceiver

The outlets needed to be controlled and since the IP controlled outlets were decided not to be used the only option was to replace the remote-controlled outlets' controller with something. The solution for the problem was RF transmitter that would work on the same frequency as the original remote controller.

The USB connected variations were nearly nonexistent and since the price was affecting factor the choice was made among the modules that could be connected to the Raspberry Pi's GPIO pins. The first though option were the nameless 433Mhz transmitter and receiver pair seen on the figure 11.

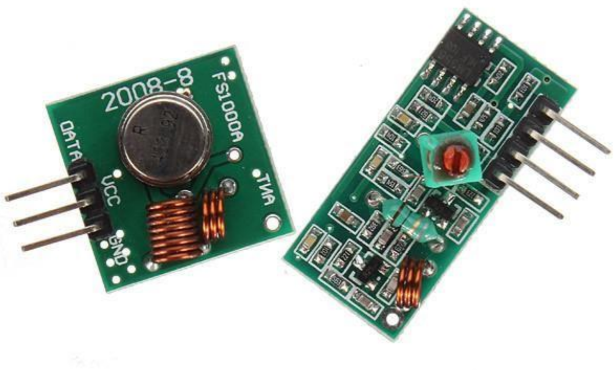

Figure 11. Wireless TX & RX 433Mhz modules [16.]

After testing and more research the idea of using these modules was scrapped. This was due to low quality of the modules that caused performance issues. The receiver that would have been used to record the remote's signal turned out to have poor noise cancellation, thus failing to get clear signal. The transmitter on the other hand provided unreliable transmission with limited range.

After searching for other option to handle the RF control, the Raspberry Pi's own Pi-mote module was chosen for the project. The Pi-mote that can be seen on the figure 12 can be connected straight to the Raspberry Pi's GPIO pins. The Pi-mote had the reliable 30m transmission range and being designed for the Pi it was fully compatible.

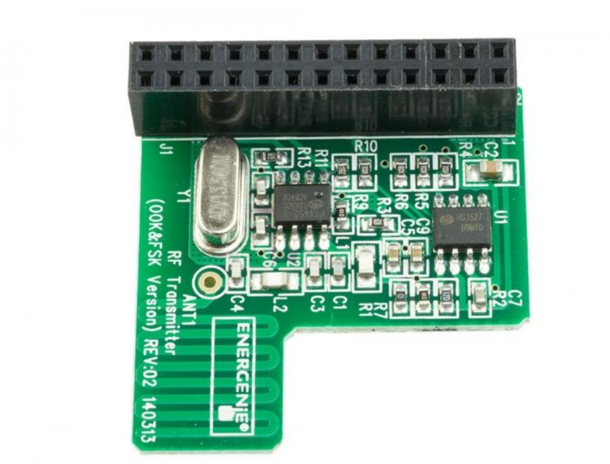

Figure 12. Pi-mote control board. [17.]

One notable flaw on the control board was the connector. The connector took 26 of Raspberry Pi's GPIO pins even though it only used 9. The DHT22 sensor required a one data, ground and power for connecting it to the Raspberry Pi. As can be seen on the figure 13 this required additional soldering so the unused pins under the Pi-mote's connector could be used for the purpose.

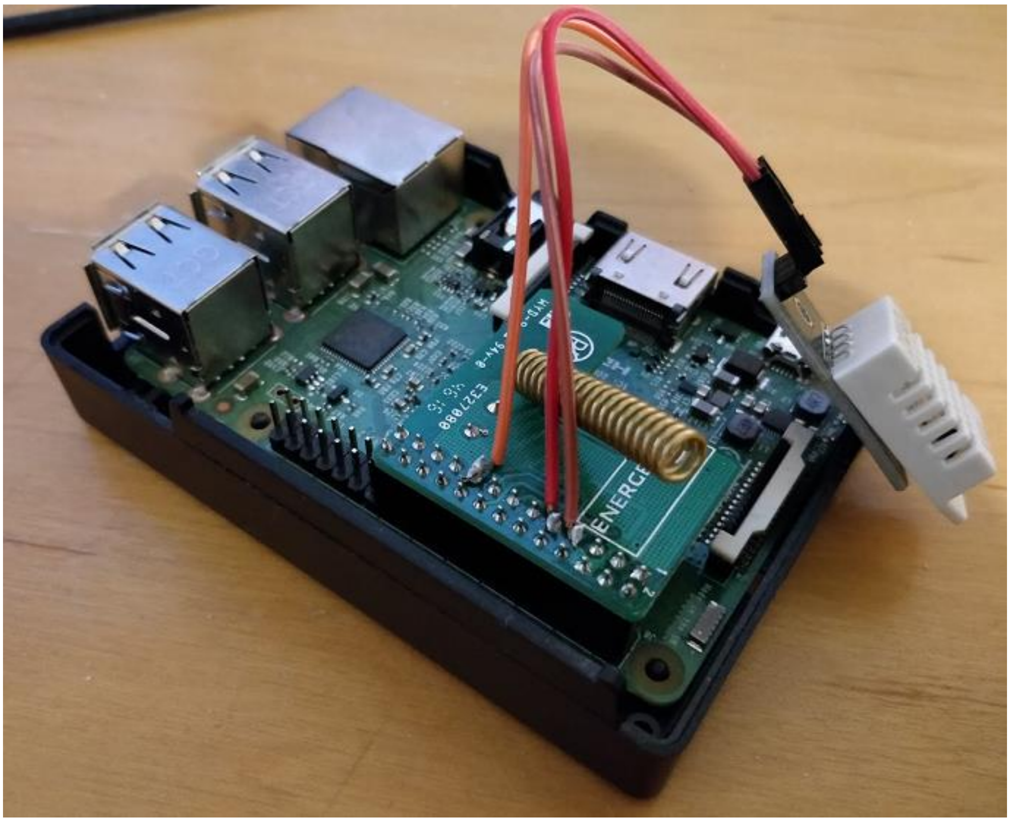

Figure 13. Modified Pi-mote

Pi-mote was also lacking an antenna which limited the transmission range. The manufacturer had recommendation of adding a 135mm additional antenna to the Pi-mote. So, to increase the transmission range and improving the signal a scrap copper coil was soldered on to the Pi-mote. [17.]

4.6 Router

The project required a router that would be used to have all the hardware connected within the LAN and to access the web page through WAN. Router would also have to have wireless connectivity and be NAT capable. Nowadays these features are not that rare so the available ZyXEL NBG6817 and LTE3301 routers were used in both testing and the end user’s site.

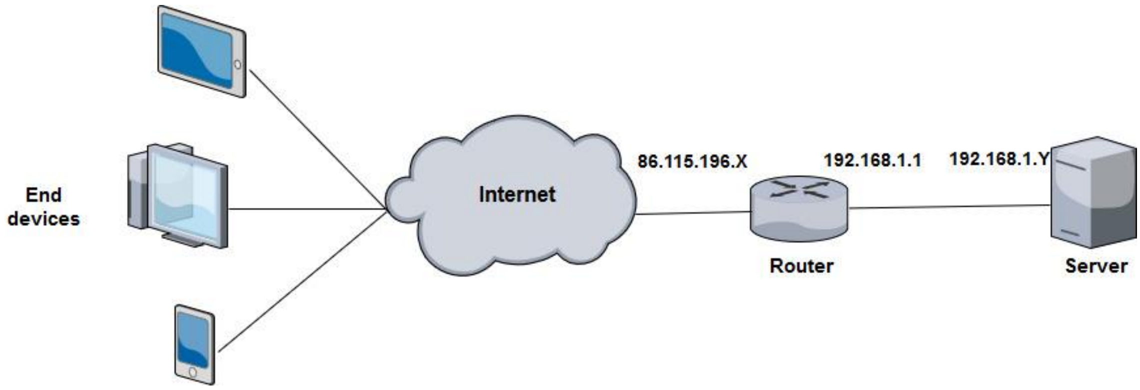

Figure 14. Topology

The figure 14 shows topology containing the main devices used on the project. The end devices which consist of computers and mobile devices are used to access the server. First the devices are used to connect to the router's WAN port address 86.115.196.X through the DNS. After this the traffic is routed to website hosted on a server or the camera stream set with Motion according to the port forwarding rules set on the router.

1	HTTPS	TCP	443	192.168.1.	443
2	User define	TCP_UDP	2022	192.168.1.	22
3	User define	TCP_UDP	8081	192.168.1.	8081
4	User define	TCP_UDP	8082	192.168.1.	8082
5	User define	TCP_UDP	8090	192.168.1.	8080

Figure 15. Port forwarding

For the router to be able to know where the user wants to connect inside the LAN a set port forwarding rules must be configured. On the figure 15 there are examples of port forwarding rules configured for the project. The most important variables can be seen on the last three columns. First there is the port that the users are using to connect to the WAN address. Then there is the LAN address of the device that is hosting the service, for example the website. Then the last column contains the actual port that is configured on the service such as the website or a camera stream hosted with Motion. This enhances the security because instead of the just the address the user also has to know the port when trying to access the services.

5 Software

The following chapter will focus on introducing the software used on the project as well as the reasons for selecting it.

The system was wanted to be created scalable and versatile while keeping the strict budget in mind. This meant that purchasing programs or paying monthly fees was not preferable solution. Also, software with limited time trials were not considered option either. Although some free versions of programs with only limited functionality were used but the limitations did not affect the system.

Ultimately this lead to that most of the used software was open sourced and thus free to use. Many of this software had very active user base providing help along with the developer's documentation.

5.1 Operating system

The operating system had to be easily modified and light for the server to run. While keeping the budget in mind it also had to be free to use. Microsoft Windows was one considered option due to its familiarity. This idea was then forfeited since it required a quite expensive license to use and might have been too heavy to be ran on the Raspberry Pi.

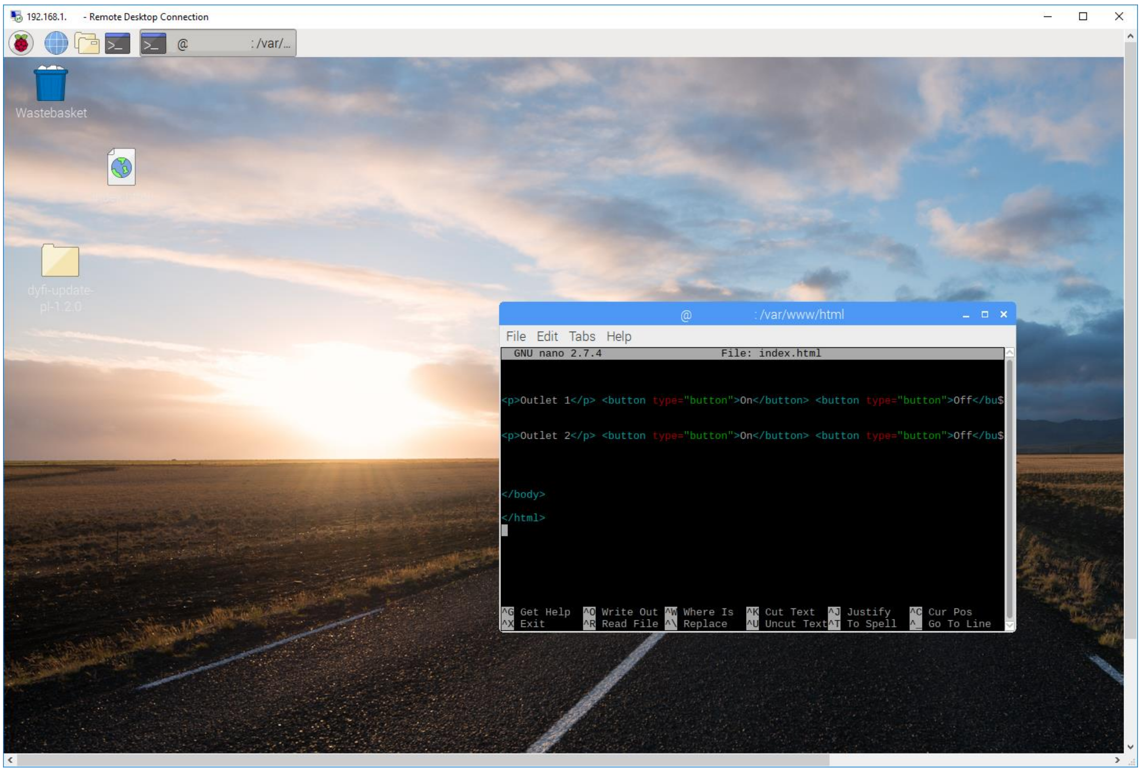

Figure 16. Raspbian's desktop

The choice was made among the wide variety of Linux based distributions. A Debian based Raspbian was chosen because it was open source which meant it was free to use. Raspbian is developed by the Raspberry Pi Foundation and the first version called Wheezy was release back in 2013. The second version called Jessie followed 2 years after in 2015 and the third version called Stretch was released recently at the end of 2017. The second version was considered briefly for the project since the very recent release of the third version. This was because of the possibility of unnoticed bugs or programs that were not yet updated to work on the new version. Although after researching it was noticed that the issues were minimal, and the third version was chosen for the project. [18.]

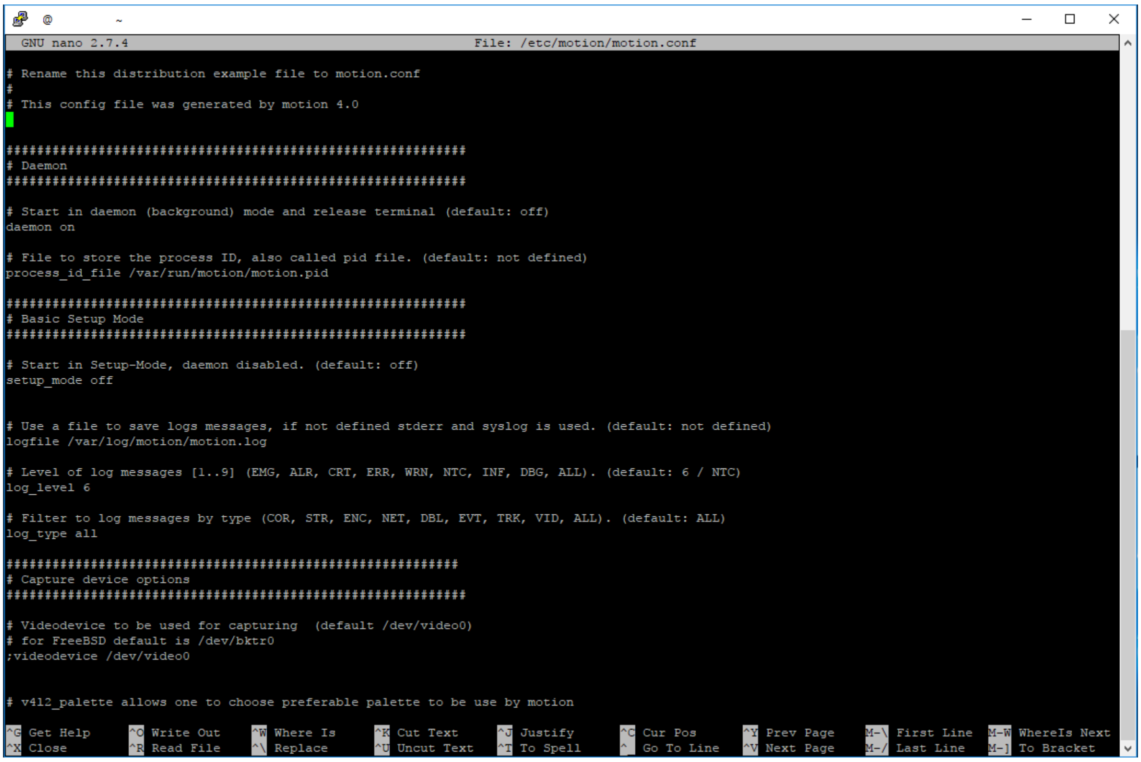

Figure 17. Remote access with PuTTY

The GUI that which Raspbian came with can be seen on the figure 16. The GUI was used at the beginning of the configurations and while familiarizing with the system. But while the project advanced the CLI became more used option due to the faster configuration and less lag while working on the server remotely. Remote access was also an efficient way to work since the hardware did not have to be moved and extra keyboard, mouse or monitor were not needed after the initial configuration. The SSH client used for the remote access was an open source program PuTTY which can be seen on the figure 17.

5.2 Motion

Since the project included multiple camera, a centralized system for configurations and maintaining was required. There were a few open source options for the Linux but because of the large user base and reasonably frequent updates the Motion was chosen for the project. Other reasons were extensive configurations, including the motion detection and scalability up to four cameras.

The Motion included one file for the general configurations and if more than one camera was deployed then also different, camera specific files for each camera. The Motion would host a small HTTP server, streaming the picture from the camera. It would run the general configuration file first and the individual files after that. This made the initial configuration easier and the possibility of adding more cameras in the future easier. Camera specific configuration file can be seen on figure 18.

```
#/etc/motion/camera1.conf
#
# This config file was generated by motion 4.0
videodevice /dev/video0

# Username and password for network camera (only if required). Default: not defined
# Syntax is user:password
 netcam_userpass *****:***********

# Id used to label the camera when inserting data into SQL or saving the
# camera image to disk.  This is better than using thread ID so that there
# always is a consistent label
camera_id 1

# Draw a user defined text on the images using same options as C function strftime(3)
# Default: Not defined = no text
# Text is placed in lower left corner
text_left Camera 1


############################################################
# Live Stream Server
############################################################

# The mini-http server listens to this port for requests (default: 0 = disabled)
stream_port 80XX
```

Figure 18. Camera specific configuration file

The general configuration file had a wide variety of options for setting up the cameras. But the important most options for the project were the quality, framerate, and the motion detection. The cameras were set to take five pictures per second whenever the motion was detected. Originally this was thought to be higher but with two cameras the server started lagging too much so the framerate was lowered. This makes the outcoming videos look more like a series of snapshots, but in case of burglary or on another similar occurrence, a clear picture of the intruder would be enough.

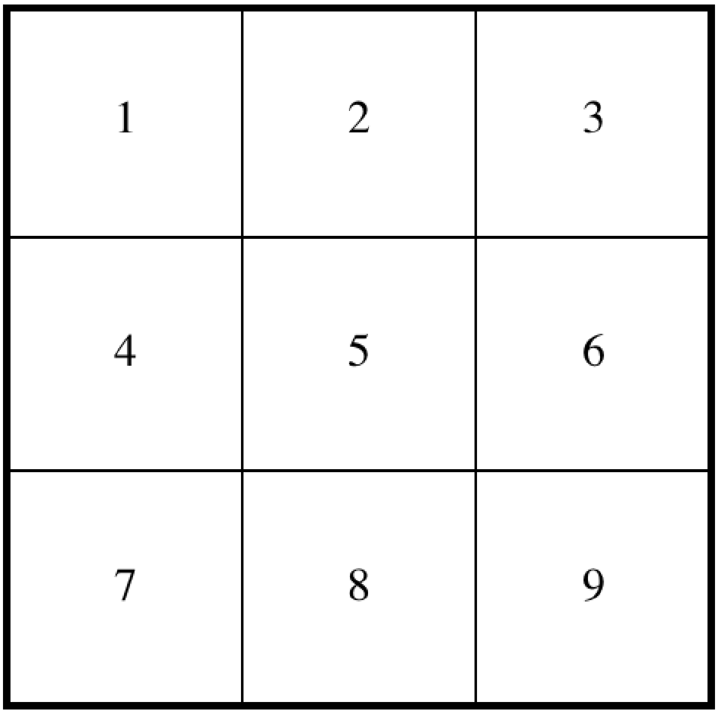

Figure 19. Motion detection areas

The motion detection option on the program worked on comparing two pictures together and checking the changed pixels on predefined areas seen on the figure 19. The program would take the configured number of pixels then reduce the noise cancellation value from it and check if the amount of changed pixels were bigger than it. If the value was bigger, then the program would start recording. After there was no more motion detected it would then wait for a one minute. If there was more movement then the recording would continue and if not, then the program would save the video file on the website.

The last requirement regarding the footage and camera part of the surveillance project was the email notification. This was done using a project called “motion-notify” created by Andrew Dean. Dean’s project initially also included the functionality of using Google Drive as a cloud storage for the footage provided by the cameras. At first this was taken to the closer consideration and testing since if the videos would have been uploaded to the cloud then the storage space on the server would have been preserved. However, in the end it was decided that the videos would be saved to the server and only the email sending part of the Dean's project was used to send an email. This was due to limit the amount of third party dependencies on the surveillance project. Since earlier in the past Google had changed its authentication method which if happened again would compromise the video storing. [20.]

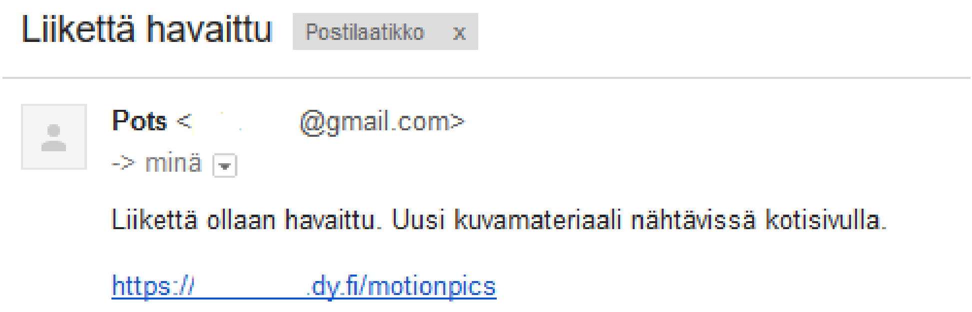

Figure 20. Email notification containing the message and link to the security footage

The email that can be seen on the figure 20. would contain subject and message informing the user that motion has been detected and a new video file is ready to be viewed. On the email there was also a link to the video section on the webpage where the video could be downloaded and watched.

5.3 DDNS

Since the system needed an internet connection to have the footage viewable through website it was decided to use the internet subscription coming to the house. Although the problem with this was that the service providers do not give out static IP addresses to consumer subscriptions. So, the chosen solution for this was dynamic DNS.

Several providers were reviewed and tested and one of them was chosen for the project. Most important features for the dynamic DNS were the cost and trustworthiness. Chosen dynamic DNS was dy.fi because it was free, and it had been online for 13 years already. Also, it was Finnish, and it provided one of the shortest domain names.

One challenge with the dy.fi was that the reserved domain name would be released after seven days if not refreshed. It would have been highly impractical to refresh it manually.

That is why the refreshing was done with the Cron, which is a job scheduler that can be programmed to execute certain tasks at the certain time.

```
#
# m h  dom mon dow   command
53 2 * * 1,2,3,4,5 wget --delete-after --no-check-certificate --no-proxy

--user=******@gmail.com --password=*********** http://www.dy.fi/nic/update?hostname=hostname.dy.fi
```

Figure 21. DDNS refreshing with Cron

The Dy.fi service approves the automatic refreshing but asks its users to do it as unfrequently as possible to not to strain their servers too much. This request was respected, and the refreshing was set to happen on every day of the week at 2.53 AM when there is less traffic. The given command to Cron can be seen on the figure 21. [21.]

5.4 Sensor scripts

The DHT22 sensor had to have logic controlling it and presenting the data on the web site. The data would have to read then stored somewhere and finally be displayed at the web site. This way the web server would only have to read the provided data and would not need any excess rights for the system, thus maintaining the security.

On the GitHub there was an available open source project called “DHT22-Temperature-Logger” created by Janne Posio. This project included a JSON config file for the individual configuration and a Python script that reads the temperature and humidity data from the DHT sensor and then places it to the database. Database used for the surveillance system project was the MariaDB which is based on the MySQL. Posio’s project also included PHP script that reads the database and fetches the readings on the web page. Although this script required refactoring as it was based on PHP 5 while the server used PHP 7. [22.]

```
#
# m h  dom mon dow   command
*/10 * * * * python /home/pi/DHT22-TemperatureLogger/DHT22logger.py
```

Figure 22. Logger script automation with Cron

The logger script was automated to be executed every 10 minutes on every day. The automation was done with the job scheduler Cron using the command seen in the figure 22. The PHP script was modified to show temperature and humidity data from the last hour and with Finnish translations.

5.5 Outlet script

In the same manner as the DHT22 sensor the Pi-mote control board required a separate script to be controlled. First script that was required was the pairing script that was used to pair the outlets with the Pi-mote. Pi-mote could be used to control up to four outlets individually and all at once, but for this project only two were used. This script was provided by the manufacturer Pi-mote and the outlets and it was written in python.

After pairing the outlets with the control board, they required a separate script for controlling them and something that could be used from the web site. It was decided that each command for each functionality would be executed with its own script, which lead to total of six scripts written in python as well. Four for turning the outlets on and off individually and all at once. Example script for tuning the individual outlet on and off can be seen on the figure 23.

```
from energenie import switch_on
switch_on(3)
```

```
from energenie import switch_off
switch_off(3)
```

Figure 23. Control scripts for turning the outlet on and off

Controlling the outlets remotely was done by adding buttons made with HTML on the website. Each of the buttons called a separate PHP script that would run the python scripts created earlier. Each python script was controlled with one PHP script and one example can be seen on the figure 24.

```
<?php
	exec("sudo python3 /var/www/html/1-on.py");
?>
```

Figure 24. PHP script for running the control script.

A dozen separate small scripts were chosen over the two bigger ones since the it was easier and less time consuming. In addition to this the possible troubleshooting would be easier in the future. Adding new scripts for new outlets would be a simple task if more outlet would be decided to add in the future.

5.6 Apache2

The surveillance system needed to have a secure and easily accessible platform to access the data and the outlet controls. The web server software Apache was chosen for the project because it was open-source and supported cross-platform, meaning that it would be accessible from multiple computing platforms.

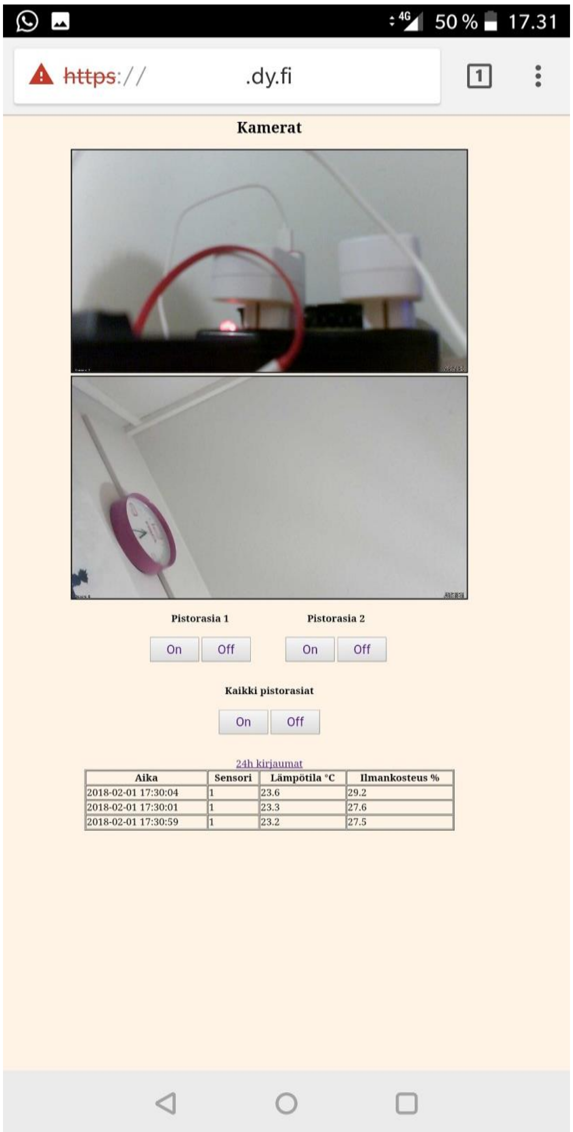

Figure 25. The mobile version of the final version of the web site

Apache was used alongside with HTML and PHP to create a compact web site that can be seen on figure 25. First on the web site there is the live video stream from both cameras that could be used to check the current situation with the property. Then there are six labeled buttons that could be used to control the outlets. Then finally the table showing the current temperature and humidity data from the last hour as well as link to another table with data from the last 24 hours.

The website was created the mobile devices in mind. The layout of the website was configured to compliment the “portrait” mode on the devices. Which meant that the elements were placed mostly on top of each other and not so much next to each other.

5.7 Security

Because the web site was accessible from the outside network the security was also an important factor for the project. One of the measures seen on the figure 26 were user credentials containing the username and password. This way it could be made sure that only authorized people could access the web site. The video streams both required their own and separate passwords which was due to wanted extra security. This way it was also possible to give somebody access to the sensor data and the outlet controls but not the camera streams.

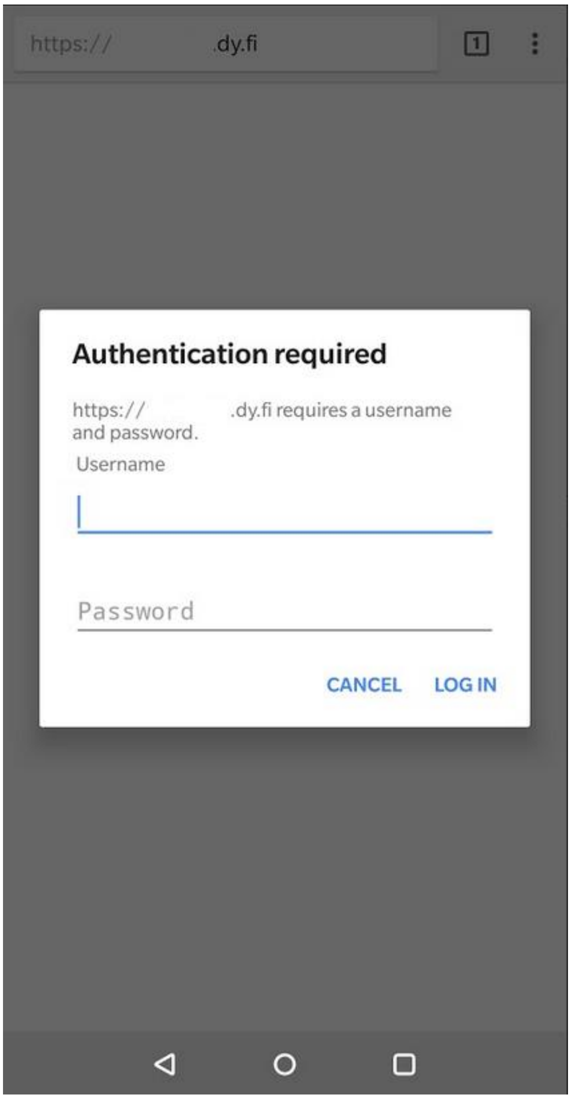

Figure 26. Log in screen accessed with smart phone

Other added security feature was the SSL security protocol. Between the browser and web page the data is normally sent in normal text. If the data is caught on the way, it can be easily exploitable. SSL adds an encryption to the data making it readable with the generated key. It was also decided that self-signed certificate would be enough because both parties know each other. In the situation when the project would be productionized a signed certificate could be bought from trusted Certificate Authorities. A self-signed certificate also produces a warning message seen on the figure 27. In cases when the party hosting the web site is known the message can be ignored.

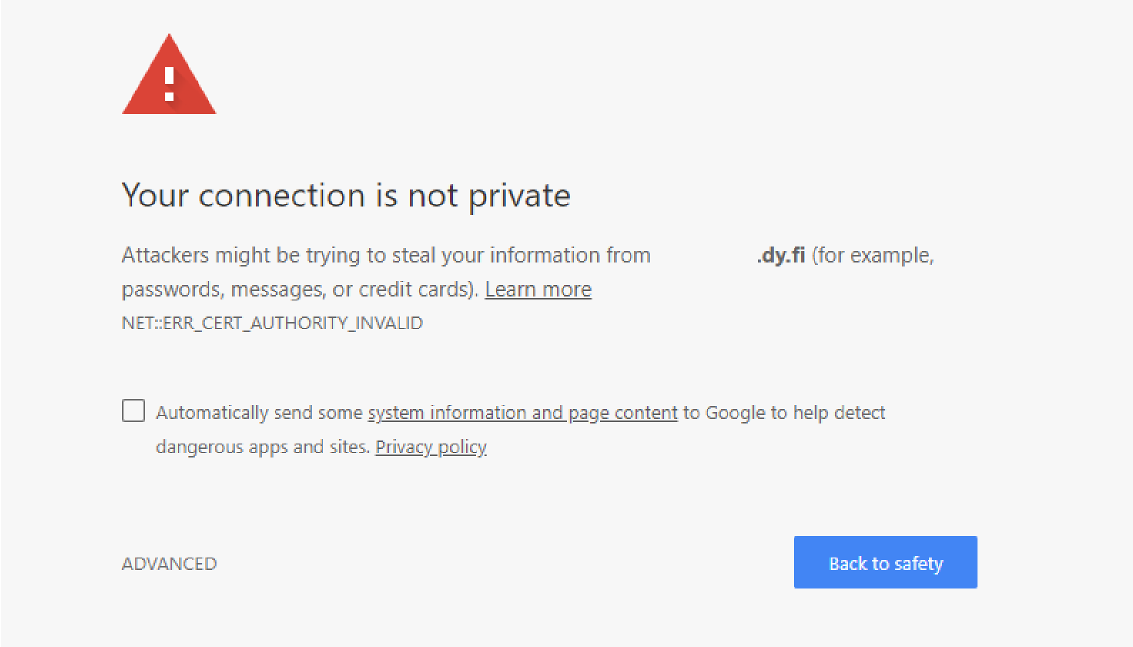

Figure 27. Message warning about a self-signed certificate

The final mentionable security measure was hiding the sensible data in the publicly available documentation. This included the login credentials, IP addresses and the domain name from the pictures and written documentation. The sensitive data was added to a more protected documentation.

6 Testing

Since the surveillance system was meant to perform quite important task helping to keep the property safe from possible intruders or damage caused by too high or low temperatures. So, it needed proper testing and put to under monitored use.

The motion detection was tested by filming house pets as they varied in size and had irregular movements. The testing showed that it was possible to set the motion detection to detect or ignore something as small as hamster. Server’s computing power was also tested by making both cameras detecting motion and saving footage at the same time. This test was also successful as the server handled the task while being fully functional.

The web site and functions on it were tested by giving access to people with varying technical skills. The test group was then asked to use the site without the instructions and give a small feedback after it. The test group was also asked to try to misuse the website and possibly to try to break it. The feedback was mainly positive, and the test group managed to use the web site effectively. This meant that the web site was intuitive and simple to use as it was initially meant to be.

The DHT22 sensor and the email warnings were also tested by bringing the temperature and humidity purposely high and low. The sensors and emails worked as intended but something was noticed. After bringing the temperature high or low it took a couple of minutes for the sensor to start showing real temperature after bringing it back to room temperature. This was deemed such a minor flaw that it did not require any actions. Since the surveillance system would have to be assembled and configured for the property's individual needs before the actual use and the temperature changes normally are not that drastic as seen on tests.

6.1 Deployment

Testing also included a deployment to a new site. The deployment to a new site had a several steps but did not create any additional problems. The IP camera's and camera streams IP addresses had to be changed on the configuration files to match the addresses given by the DHCP on the other location. Because of the different service provider, the deployment site used IP addresses of 192.168.100.X instead of 192.168.1.X used on the working site. In addition to changing the IP addresses several port forwarding would have to be added.

The cameras would have to be placed on the wanted locations and plugged to a power source. The server had to be placed close to the USB camera because of the limitations set by the cable. One of the remote-controlled outlets was placed next to the server and other on the different room but within 30 meters of the server.

These steps required a basic knowledge of networking and familiarity with CLI based Linux. The deployment was not meant to be done by the end users and thus included complicated steps to someone with limited technical skills. In case of making the surveillance system's deployment simple enough for anyone to do, or even productizing it, there would have to be some changes. The system could include a mobile router with all the configurations already made. Static addresses could also be given because the system would always use the same router. So, the addresses would not have to be changed on the configuration files.

Setting up new email account on the system for the new user or changing the motion detection settings would not be as easily solved. These would require a creation of a new application with UI simple and informative enough for the end users. Also, the initial configuration of the remote-controlled outlets would have to be included on that application. So, as the surveillance system was, the deployment would have to be done by a technical person.

7 Summary

The purpose of the project was to create a surveillance system that would help with the property surveillance. This was required in cases when the residents were absent longer periods of time, for example the holiday trips. Other possible purposes were to surveil properties that would be accessed frequently but could have rapid changes on condition, for example the boiler room.

The project started by mapping out all the needs and creating a rough plan how to achieve all of that. This helped with the big picture off the project and created a visual progress tracker. After that each part was taken under the work and crossed off after completion. Then some notes such as IP addresses and password were added for later use.

One of the initial goals of the project was work on the minimum budget. On the software side this did not lead to compromises or that much problems, but hardware side had some setbacks. All the smaller components that could were ordered from Chinese web stores because of the low prices. This meant for long delivery periods that slowed down the project from time to time. The remote-controlled outlet part especially yielded to one full working day of wasted time after the transmitter was proven to be unreliable and not working as initially thought.

The software part yielded better success since of the initially thought software were used and configured successfully. On the Linux based server there were multiple options for each part of the project, so it was just the matter of picking the best suited and supported ones for the project.

The final product turned out to fulfill every initially set goal and requirement. All of this while leaving room and readiness for future ideas and development. However, as stated on the Testing section, the deployment and configuration would require skill and experience. Although this is the case with the commercial systems as well. Since the users would not be able to add more devices or configure the system by themselves. But on both cases the footage and sensor data would be easily accessible.

The system is well-functioning and short-term testing has shown it to be reliable. However, at the moment it is too early to say if the situation would change after longer use. Most of the devices and components have 1-2 years of warranty so there is trust on the functionality on manufacturer's side. Although the conclusion could be made only after longer testing and the changes made to devices and components according to that.

In case of something breaking and then replacing it would take anything from one day to three weeks or even longer depending on the component or device. Companies offering the commercial security systems do not state the time it takes for replacements. Although, because of the nature of the business it would be safe to assume that any of the parts would be changed in less than a week. This assumption is based on available stock and staff.

In the end the commercial system would be preferred in the locations where the risk of illegal activity is higher, and the usage is continuous and, when the guard or other services are needed. While the homemade systems greatest assets are the price and versatility. It would be a considerable option for the locations with lower risk and smaller budget. Such as the site it was designed for at the beginning of the project.

References

1 Lyon David. Surveillance Studies: An Overview. Cambridge: Polity Press; 2007

2 (2015) Harding Luke. Mass surveillance is fundamental threat to human rights, says European report. The Guardian. https://www.theguardian.com/world/2015/jan/26/mass-surveillance-threat-human-rights-council-europe (Accessed 10 February 2018)

3 (2017) *Vertailimme kodin turvajärjestelmiä: hinnoissa ja palveluissa tuntuvia eroja* Available at: https://yle.fi/aihe/artikkeli/2017/01/25/vertailimme-kodin-murtohalytysjarjestelmia-hinnoissa-ja-palveluissa-tuntuvia (Accessed 27 November 2017)

4 (2017) *Aloituspaketti liiketunnistimella ja savuilmaisimella* Avalable at: https://www.sectoralarm.fi/media/57486/kotihalytin-aloituspaketti-ei-kameraa.jpg (Accessed 10 January 2018)

5 (2017) *Netgear Arlo Pro 2 -valvontajärjestelmä kahdella Full HD -tasoisella kameralla* Avalable at: https://cdn-a.verkkokauppa.com/images/87/2_437633-959x592.jpg (Accessed 10 February 2018)

6 (2016) *Raspberry Pi 3 Grand Opening* Available at: https://tech.scargill.net/raspberry-pi-3-grand-opening/ (Accessed 25 October 2017)

7 (2017) *DHT22 2302 Digital Temperature and Humidity Sensor Module – White* Available at: http://img.dxcdn.com/productimages/sku_184847_2.jpg (Accessed 5 November 2017)

8 *DHT22 temperature-humidity sensor + extras* Available at: https://www.adafruit.com/product/385 (Accessed 7 February 2018)

9 (2017) *Fuj:tech HD -web-kamera* Available at: https://www.verkkokauppa.com/fi/product/16357/gscgs/Fuj-tech-HD-web-kamera (Accessed 1 November 2017)

10 *Fuj:tech Drop HD -IP-kamera* Available at: https://cdn-b.verkkokauppa.com/images/43/2_331283-924x1222.jpeg (Accessed 8 February 2018)

11 *IP Webcam* Available at: https://play.google.com/store/apps/details?id=com.pas.webcam&hl=en (Accessed 10 Novermber 2017)

12 (2016) *IP Power Reboot, Remote Power Control* Available at: http://primalcom.com/promo/wp-content/uploads/2016/08/9258S_exp_001.jpg (Accessed 7 Januray 2018)

13 *Energenie 2 x Remote Controlled Sockets with Pi-mote (EU)* Available at: https://www.modmypi.com/image/cache/data/rpi-products/breakout-boards/energenie/ener002-2pi-eu/DSC_0812-800x609.jpg (Accessed 27 November 2017)

14 (2017) *IP-luokitus* Available at: https://www.stek.fi/Perustietoa_sahkosta/Sahkojarjestelmat/fi_FI/IP_luokitus/ (Accessed 5 November 2017)

15 (2017) *Kauko-ohjattava lisä ulkopistorasia 4564 / 6867 38-4673* Available at: http://www.motonet.fi/fi/tuote/384673/Kauko-ohjattava-lisa-ulkopistorasia-4564--6867 (Accessed 10 November 2017)

16 (2017) *RF Wireless Tx & Rx Module 433Mhz* Available at: https://mikroelectron.com/ProImg/X4/0ce0ac02-a662-46f6-9f09-45f1488d3442.JPG (Accessed 29 October 2017)

17 *Energenie 2 x Remote Controlled Sockets with Pi-mote (EU)* Available at: https://energenie4u.co.uk/catalogue/product/ENER002-2PI-EUR (Accessed 27 November 2017)

18 (2017) *RASPBIAN* Available at: https://www.raspberrypi.org/downloads/raspbian/ (Accessed 10 February)

19 *What is an SSL Certificate and How Does it Work?* Available at: https://www.digicert.com/ssl/ (Acessed 1 January 2018)

20 (2015) *Migrating to Google Sign-In* Available at: https://developers.google.com/identity/sign-in/auth-migration (Accessed 10 November 2017)

21 (2017) *Client software* Available at: http://www.dy.fi/page/clients (Accessed 10 November 2017)

22 (2016) *jjpFin / DHT22-TemperatureLogger* Available at: https://github.com/jjpFin/DHT22-TemperatureLogger (Accessed 2 December 2017)