BSc (Honours) Degree in Computer Science

Final Year Project

Home Security System based on Raspberry Pi

By

Kyriakos Naziris

Project unit: PJE40
Supervisor: Athanasios Paraskelidis

April 2015

Computing and Mathematics Programme Area

ABBREVIATIONS

SMS	Short Message Service
IDE	Integrated Development Environment
I/O	Input/Output
L.E.D	Light Emitting Diode.
CPU	Central Processing Unit
SD RAM	Synchronous Dynamic Random Access Memory
USB	Universal Serial Bus
HDMI	High Definition Media Interface
SD Card	Secure Digital Card
GPIO	General Purpose Input/Output
SATA	Serial Advanced Technology Attachment
GSM	Global System for Mobile
RFID	Radio Frequency Identification and Detection
LCD	Liquid Crystal Display
FTP	File Transfer Protocol
UPS	Uninterruptible Power Supply
Wi-Fi	Wireless Fidelity
PIR	Passive Infrared
IP	Internet Protocol
CSI	Camera Serial Interface
API	Application Programming Interface
KB	Kilobyte
MB	Megabyte
UNIX	Uniplexed Information and Computing System
EPR	External Power Resource
DC	Direct Current
V	Volts
A	Ampere
mAh	Ampere hour

ABSTRACT

Home security system gives an extra peace of mind and security in your house while you are sleeping or even away from home. From the moment the system is armed it will automatically keep you up-to-date for unexpected movements within your property. Home security has been around for a several decades now, where people were using their animals to get protected. But the evolution of technology has allowed the use of electrical circuits and micro-controllers which made the system even more accurate and systems for every need. Furthermore, the expeditious evolution of the internet and smartphones allowed the home security systems to be controlled remotely at any time, which previously the only way to control them was through the central control panel embedded on the system.

This project examines the process of the production of a conceptual home security system for a client, which can secure and monitor a desired room. This project will involve research of the home security systems, on the currently on the market as-well as the hardware and software will be used, which will lead into the project's requirements determination. The main aim of this project is to find alternatives ways to design and develop a low cost system which is easy to configure and easily expendable in the future. The development of the system is documented, starting with requirement analysis and ending with the testing of the system. The entire project is then evaluated and concluded, discussing possibilities of future developments.

1. INTRODUCTION

1.1 Problem Overview

Home Security has been around for quite a few years, as humanity have the need to be protected as well as their properties. However, as the technology keeps rising up the hardware part got the firm to be more affordable, more accessible and smaller. Back in the years, such security systems required remote controls or some kind of a keypad, but now with the Smartphones and Tablets you can control your system and keep an eye on your protected area from anywhere at anytime.

Mr. Gregoris Gregoriou an individual plumber, has requested the development of a home security system. Mr Gregoriou keeps his expensive tools and different types o pipes in his storage which is currently not equipped with any sort of security. The requested system should monitor the storage and report any unexpected movements. Those reports should be send via SMS to Mr. Gregoriou, but because sometimes he is not monitoring his phone, an e-Mail should be send to his wife e-Mail address which is monitored most of the times. The project scrutinize the required hardware and software to develop such a system. By going through this investigation, every step of the development process is evaluated, and well documented

1.2 Project Aims

The aims of this project are to go through the development process of a home security system and produce a conceptual system, so it can be developed further as a commercial product for the use of the client. The objective of this work is to design and implement a security alarm system that alerts the owner via a SMS and e-Mail if unexpected movement is detected. The system will also feature two different forms of activation/deactivation and a live streaming video will be also available. The advantages of this home security system using alternative open systems, are its high security level, robustness, low cost and ease of use and that there is no distance limitation for contact.

1.3 Project Objectives

1. Evaluate various home security systems currently on the market
2. Discuss with the client and determine the systems requirements.
3. Select a suitable software development life cycle methodology to develop the system.
4. Select suitable components for the hardware part of the project.
5. Select suitable software development tools for the web based control panel.
6. Put the hardware of the system together and prepare the hardware to run the software.
7. Develop a stand-alone application and a web-based application which can control the systems components.
8. Document the whole process of the project

1.4 Conclusion

Now that problem which needs to be solved, and the aim of the project is known the next step is to do some research of the similar systems currently on the market, so the developed product could be comparable to those. The following section covers the research procedures that have been followed.

2. RESEARCH AND LITERATURE REVIEW

2.1 Research Overview

This section examines the research that has been made on the home security substance, going through the very early until the modern technologies and methods, used to monitor and secure homes. Crucial investigation has been made, on hardware and software currently on the market, to elect what will be used to develop this project

2.2 Hardware

Hardware for this kind of project is essential. The heart of the system will be a single board which will surrounding all necessary data. On the main board a sensor is attached, where they record data about the environment it is in, such brightness, and movement. The data once they been recorder, they are going to the main board which for further processing.

2.2.1 Arduino

Arduino (Arduino Software, 2015) is one of the most popular controller boards today. There are different models of Arduino available on the market but, for this project Arduino UNO which is based on the ATMega328 was put under consideration.

Arduino has 14 digital I/O pins and it has no embedded Ethernet port but, it can be provided by purchasing an ethernet board shield which is extra of cost. Because of the luck of the ethernet port, the communication to the board is limited to Serial, and only through its own IDE. Arduino in reality is a micro controller, it does not need interpreter, operating system or any kind of firmware for that reason it 80% lower power consumption compared to the boards that are described underneath. Its ATMega328 chip “understands” on the C++ programming language,. There are some articles on the internet that make it work with python but is not what is designed for. Sensors, Displays and L.E.Ds work better using a micro controller but is not a good idea for projects that includes cameras or other interfaces because, micro controllers are more on controlling than processing. Finally, Arduino Uno costs around 20 to 25 pounds.

2.2.2 Raspberry Pi B Model

The Raspberry Pi (Raspberry Pi Foundation, 2015) is a Single Board Computer (SBC) which runs on a single-core Broadcom CPU at 700 Mhz and 512MB SD Ram. On the board it has Video, Audio, USB Host, HDMI, SD Card and Ethernet port. Its Ethernet port, shares the same bus with the USB Host so, its speed is limited to 10/100 MB. What it makes it so special are the embedded 26 GPIO headers which look like Arduino pins but are slightly different. The difference is that Raspberry Pi is a microprocessor, so it has weak I/O and needs transistors to drive most hardware but, microprocessors are good at processing which make the a little bit brainier than a micro controller. Raspberry Pi is runs a form of linux designed especially for this board. It is also have a massive list of components design just for it self, as well as the GPIO utility for manipulating those components from the command line. Additionally, it has a community of thousand of people building different projects or providing support to other individuals to build their own projects. This board does not use an IDE like Arduino, it uses the terminal console of the linux system to write the programs, and instead of using the serial communication, it can be controlled just as a normal computer or if its going to be used remotely via ssh.

The development languages it supports are not limited to C++ but, instead the most popular and most widely used is Python. The board cost around 28 pounds, including the SD Memory Card that is required as a storage, in order to set it up.

2.2.3 Banana Pi

Banana Pi (Lemaker, 2014) is an imitation of Raspberry Pi with some better specifications and added features, It has 1GB DDR3 Memory, it runs on a Allwinner dual-core A20 CPU at 1GHz. On the board are embedded all the ports of the Raspberry Pi but in addition, it has an IR, Microphone, SATA and some other ports that are not really needed for this kind of project. Also, instead of a 10/100 MB ethernet port, Banana Pi supports a 10/100/1000 MB (Gigabit) port which really increase the bitrate. Furthermore, it includes 26 GPIO headers, it uses its own open source operating system and it needs a SD Memory Card as a storage. The downside of this board compared to the Raspberry Pi, is that the quality of the board it self lucks, the paring system designed for it, is not stable and is still lucking, most of the components software are not really designed for it, are just ported from the Raspberry Pi. In addition, its components compatibility list and its community support is very short but it keeps increasing. This board has the highest power consumption and price which is around 35 pounds without the SD Memory card, but this s might happen because of the feature it has.

2.2.4 Hardware Evaluation

Based on the research that has been made on already attempted projects and current home security system hardware on the market,

- Arduino - Cheap, huge list of components compatibility and great community support. However, it lucks on internet connection, it is just a micro controller and the interface of the project will not be able to run on it. Which leads to be unconsidered for this kind of project.
- Raspberry Pi - Probably the best and most reliable software and hardware it can be used for this project. Its cheap, it has a massive list of compatible components and thousand people in the community to provide support.
- Banana Pi- It has the best CPU and Memory and the longest list of features. However, it lucks on the quality and reliability. It is also expensive compared to the other 2 boards.

2.3 Similar Systems

This kind of project has already been implemented in the past either as a commercial product or as a DIY project, each of them with various numbers of features. On the following section it will be given a brief overview of a DIY project and of a commercial product currently on the market.

2.3.1 PiLarm: Portable Raspberry Pi Room Alarm

PiLarm (Jeff Highsmith, 2013) is a really good inspiration on this project. PiLarm is a simple home security alarm involving a matrix keypad, a speaker, a camera, a motion sensor and LEDs. Highsmith's system on motion detection, is sending a message on tweeter and an email, in order to inform the user about the unexpected intrusion. In

addition it allows some basic voice instructions through the embedded speakers, to inform the user about the system status or sound the alarm to warn the user.

The total cost from the scratch of the Highsmith's project isn't exceeding 120 pounds. The project uses simple python functions, and the structure of the implementation is even simpler to understand. In addition, it gives a good kick-start, on the implementation and design of the current project.

2.3.2 Arduino Anti-Theft Alarm Shield

Arduino Anti-Theft Alarm Shield is a Complete anti-theft alarm compatible with any kind of sensors, with RFID recognition, GSM and web server, based on Arduino Mega (IngGaro, 2014). The system can be managed and configured from the embedded web application, and it is own control panel, offering LCD display, capacitive buttons and RFID recognition. Furthermore, the system sends e-Mail and SMS notifications when it is alarmed, and a backup battery is available in case of a power fault. Also it offers certain automation features such as close rolling shutter when it is activated. IngGaro's system has 4 different sensors which can be activated independently, open window, temperature/ humidity, perimetral and volumetric.

2.3.3 PrivateEyePi - Home Alarm System Project

PrivateEyePI (PrivateEyePi, 2013) is another home alarm system based on the Raspberry Pi. This system can have different zones which can be operated independently. Each zone may have door and window open sensors and motion sensors. The zones can be managed using the web based control panel. The control panel contains activity logs and the current temperature of the house. The system is easily expandable and has no limit to zones and sensors.

2.3.4 Commercial Home Security Systems

Except of Highsmith's PiLarm, another source of inspiration is the massive number of commercial security systems. Commercial products could help a lot to understand and start thinking about what features a real-world Home Security System needs and how the user should interact with the system on a daily basis.

The ideas that have been collected from the commercial products wasn't from just a couple of systems but, a mixture of various products from various companies. Also, a visit has been made in a major company named Multitech L.T.D in Cyprus, which made a huge impact on the features requirements. They have been asked several question about what their clients really asking from a home security system and what is their current best selling home security system. They have introduced the D-Link DCS-6004L HD(D-Link UK, 2012) which is an IP surveillance camera that offers live streaming video and audio, motion detection and remote access. It also offers e-Mail notifications and FTP remote storage. The questionary can be found on the appendix C.

2.3.5 Conclusion

PiLarm - A good a simple home security approach but no remote access control is available neither SMS notifications. The way it is designed though it is inexpensive and it could really give a kick start to the project.

Arduino Anti-Theft Alarm Shield - As the creator states is a complete anti-theft alarm, as is offers great features but most of them are useless for a storage monitor system, and the quality of the design is way to complicated and expensive. It could give feature ideas though, to the system.

PrivateEyePi - Another simple home security system like PiLarm, but instead of embedded control system, it offers a web based control panel which is poor in functionality. A really good feature it offers though, is the unlimited number of zones and sensors it can support.

D-Link DCS-6004L HD - This commercial product is more a surveillance camera than a home security system. It offers some great features like live streaming and email notifications, but some essential features like siren sound and expandability are not provided.

None of the four systems fulfil clients requirements, but a mixture of the great features of each one can. On the following section the system's requirements will be determined taking in mind the client's needs, so the design of the project can start.

3. REQUIREMENT ANALYSIS

3.1 Research Conclusion and Chosen Methodology

Taking in mind the undertaken research and the discussions with the client it has been decided that Raspberry Pi will be used as the main board. As Arduino can not reach the systems requirements and the Banana Pi lucks on compatibility, reliability and support, this is a one way option. Fancy features and huge amounts of memory sometimes is not what is only needed, quality, reliability and support worths more than that

Due to time constraints for this project, Highsmith's PiLarm project idea is used at the early stages and then started building on it, until the required result is reached. On this project though, some of the components and features that Highsmith used will be excluded as the client does not required them.

3.2 Determining Requirements

The requirements of the projects have been discussed with the client and both functional and non functional requirements were determined. Considering the client's needs a system idea have been proposed to the client, which he was quite happy with the proposal system idea but instead of RFID tags to alarm and disarm the system, he preferred to somehow type a four digit passcode. Because he is afraid of damaging or loosing the tags. Also during the discussions, the needed time and my commitment to the university's workload were also discussed and accepted by the client. The requirements are followed into the next sections.

3.3 Functional Requirements

3.3.1 Specification

BASIC SYSTEM REQUIRMENTS

ID	Priority	Description
FR1	1	The system should be able to be armed and disarmed through the connected 3x4 keypad.
FR2	1	In case of unexpected movement the system should take a picture store it locally and on an external FTP
FR3	1	In case of unexpected movement the system should send e-Mail and SMS notifications.
FR4	3	The system should have voice indications - A voice will inform the user after the passcode is entered if the system is armed or disarmed.
FR5	3	The system should have L.E.D indications - The L.E.D will indicate whether the system is powered and
FR6	1	The system should display indications on a screen - The indications will be instructions to the user or current system status.
FR7	4	The system should have a UPS battery as an external power source.

Figure 3.1 - Main hardware system functional requirements

3.3.2 Additional Discussion and Justification

WEB BASED CONTROL PANEL

ID	Priority	Description
FR8	1	The system should be able to be armed and disarmed through the control panel.
FR9	1	A live streaming page should be available on control panel.
FR10	2	The control panel should be secured - The user should be able to see the control panel only if its logged in with its user account.
FR11	2	The user should be able modify or reset details like username and password.
FR12	2	On the control panel it should be a force take picture button where it will capture a picture and store it, no matter if the system sense unexpected movement.
FR13	4	A shutdown button should be available where will allow the user to shut down the system.
FR14	1	A page with a grid of all the images the system captured should be available - The images will be gathered from the external FTP server,
FR15	4	The system should keep a logs records - Those records will be systems incidents and what time they happened.

Figure 3.2 - Web based control panel functional requirements

The requirements of the project have been categorised into 4 different priorities:

Priority 1 - Core requirements that are required to be implemented in the first stage of the development, because other requirements are depended on them.

Priority 2 - At this priority lever will be implemented the core requirements of the web based control panel

Priority 3 - This priority level cover some important requirements but not really essential, that makes the system comparable to the most commercial products out on the market.

Priority 4 - At this priority level will be implemented some extra requirements that are not really needed for the system to function, but they really improve the usability of the system.

3.4 Non-Functional Requirements

3.4.1 Platform

- The web based control panel must be supported on most modern browsers and mobile devices.
- Must be as compatible as possible with future browsers and mobile devices releases.
- The system should be easily expandable for desired added feature in the future.

3.4.2 Usability

- The system must be as user-friendly as possible.
- The English language should be the system's main language.

3.4.4 Performance

- The system should be stable and able to work 24/7 uninterrupted.
- The web based control panel should perform in a timely manner on most modern browsers and mobile devices.

3.4.5 Portability

- The system must be able to work using Ethernet or a Wi-Fi network.
- The enclosure of the system should be as small it can be and easy to move.

3.5 Methodology

Methodologies are the framework used to structure, plan and control the process of developing a system (Chowdhury & Huda, 2011). Choosing and following a methodology on a project, is one of the most important tasks, because it provides guidelines to follow for completing a project, including techniques, models and tools. In addition, choosing the correct methodology for your project can speed up the project process, can help to deliver the product on time and reduce the risk of failure.

Incremental development development methodologies methods are really suited for small individual project that have to be developed in very short period of time. It is similar to the waterfall model but in addition it allows backtracking which is not possible to waterfall (Van Cauwenberghe, n.d.). Backtracking allows changes to the initial design and implementation, if needed. In fact this model works in sprints that last 1 or 2 weeks. Using backlogs from the user's requirements it creates different sprints and each sprint is a group of similar features. In each sprint the developer works on each feature separately and tests it concurrently. Often, testing involves the client/user to be encouraged to test the system and make sure is what is expecting (Panoptic Dev, 2014). When testing is done, a new sprint starts, which in reality each sprint adds a new functionality on the application. In addition, during the development process, the application is in a continuous state of testing until it is ready for deployment.

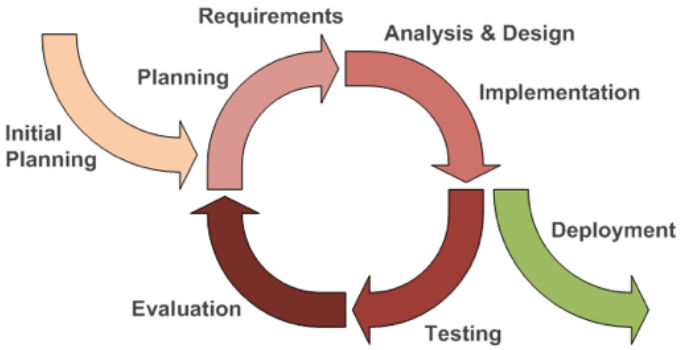


Figure 3.3 - Iterative and Incremental Development Model (Sheehan, 2013)

The chosen methodology for this project will be Feature Driven Development (FDD) which is an Iterative and Incremental Development process. Since it allows feedback to be given from the client on each iteration, it will make sure that the client will be pleased from the final outcome. Also, it allows to give priority to selected core feature that need to be developed first and then add features with less priority later if sufficient time is available.

3.6 Planning

The project plan will be followed is located in appendix E.

3.7 Conclusion

In this section the functional and non functional requirements were determined and the methodology will be followed is chosen. The next section will give shape to the project. It will go through the design of the hardware, and the control panel which they will fulfil the requirements have been determined.

4. DESIGN

4.1 Design Overview

This section covers the design decisions that have been made after discussions with the client. The followed documented decisions were followed as planed, but the client in the process of the implementation asked for some extra feature were they are added later. Those extra features have been added proving that the chosen FDD methodology was the right option, cause it allowed design changes to be made on to the project for each iteration.

4.2 Hardware Design

The hardware design is a very important task, as most of the requirements are depended on the hardware. If the wiring quality of the components is not very good, then the user it might get false indications and notifications or even cause damage to the components it self. Therefore, the components should be wired in a thoughtful manner in order to avoid any isuues.

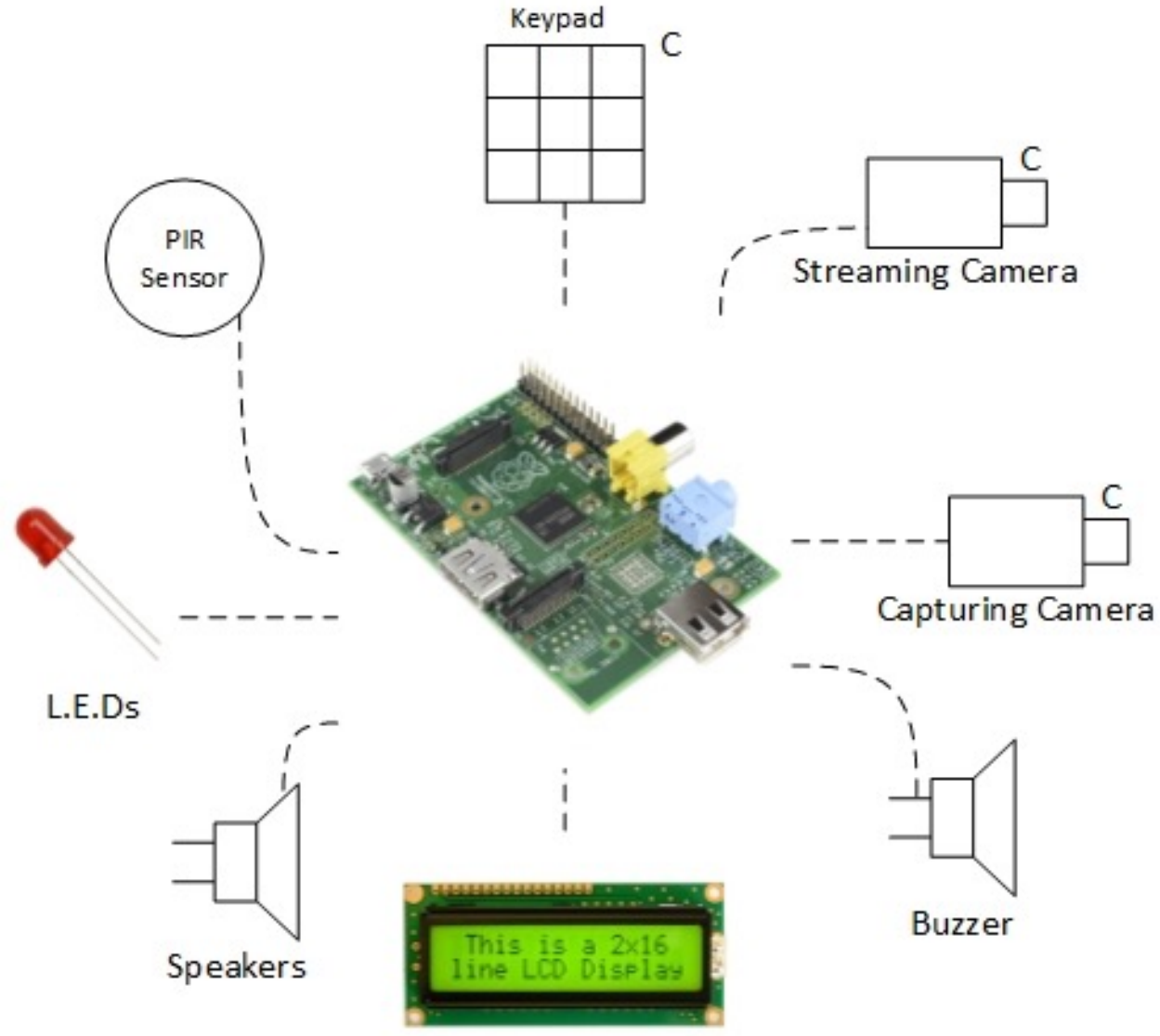


Figure 4.1 - Initial design of the hardware

The figure 4.1 shows the initial design of the hardware. The components surrounding the Raspberry Pi will be connected to it and each of them will fulfil the clients requirements. The 2 cameras will add the live streaming and picture capturing requirement, the L.E.D, LCD display, speakers and buzzer will guide the user through the management of the system, and the PIR sensor triggers the alarm of the system if movement is detected.

4.3 System Architecture

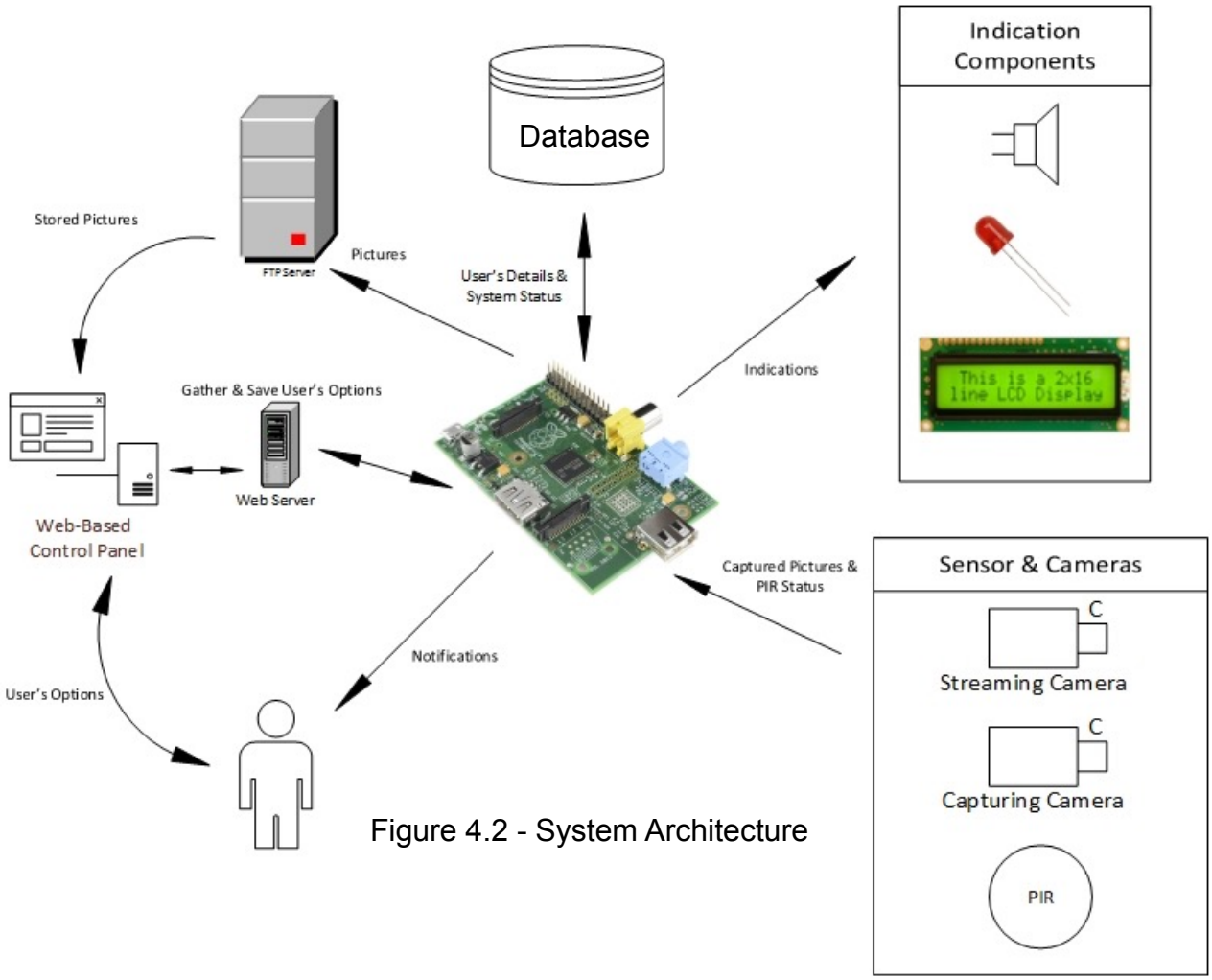


Figure 4.2 - System Architecture

The figure 4.2 shows the actual architecture of the data flow, the system will follow. The data flow is consisted of the following:

- Raspberry Pi - This is the brain of the system. Every bit of data is going through it for processing and then sends the processed data to its destination.
- Sensor & Cameras - Those components are used as inputs into the Raspberry Pi when they required.
- Indication Components - The Raspberry Pi will send indications to those components to inform the user for the system's current status.
- Database - Where user's and system's data is stored.

- FTP Server - This the external server where the captured pictures are stored and can be gathered from the web server.
- User - The user sends it is request to Flask (Ronacher, 2014) application (web server), which through Flask is going to the Raspberry Pi for further processing. Also, is receiving notifications from the system such as email and SMS.

4.4 Web-Based Control Panel Design

The design of the control panel is another important task, since is the only way the user will have full control of the system. It should be be clear and easy to navigate through. The outcome of this design should be a user-friendly interface which is not difficult to use as Mr. Gregoriou has limited knowledge of using computers. Also the control panel must includes all the required functionality that has been discussed with the client.

4.4.1 Interface Layout

The control panel has a total of 5 different pages, which the user can navigate through the menu that has placed on the top of each page. The pages are as follows:

1. A page allows the user to arm and disarm the system, take a picture and shut down the system.
2. A page, where the user is able to see the pictures have been captured.
3. A page where the user is able to watch a live streaming video of the house.
4. A page,where the user is able to see a list of the system's incidents.
5. A page where it contains the user's details, and the user is able to modify those details.

The figures bellow show the layout of the 5 pages mentioned above and the pop windows will be used in some occasions:

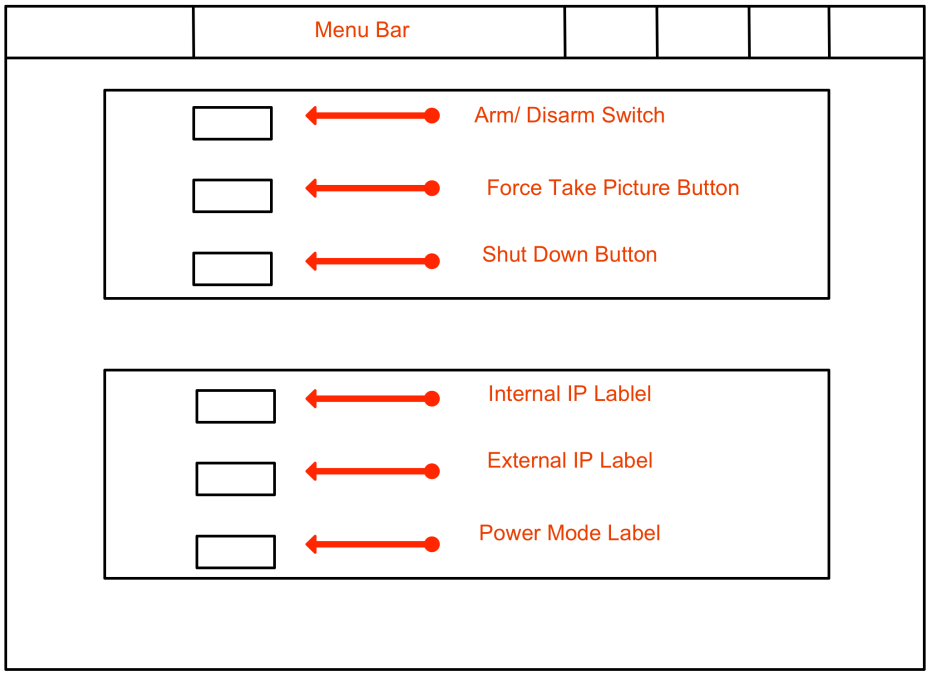


Figure 4.3 - Toggles page: System Control & Details

The toggles page shown in figure 4.3 will contain the following functions:

- Arm/Disarm Button - This button will allow the user to arm and disarm the system remotely, without the need of the embedded keypad. This button will pop up a confirmation window that asks the user to type the 4-digit passcode. The system checks the if the passcode is correct and then changes the system status on the database. Also, during this process, the L.E.D, speakers and display output indications for this changes.
- Force Take Picture Button - By pressing this button, a picture is taken by the USB camera and it is uploaded on the external FTP Server. Once the upload is finished, a window should pop up and display the captured picture.
- Shut Down Button - This button allow the user to shut down the system. When pressing it a pop up confirmation window asks the user to type the 4-digit passcode. The system checks the if the passcode is correct and then shut the system down. If the passcode is incorrect, it displays an error message.
- On the page will be 3 labels as-well that will show the internal and external IP for away from home use, as well as the current power source of the system. These information give an overview of the system's current status.

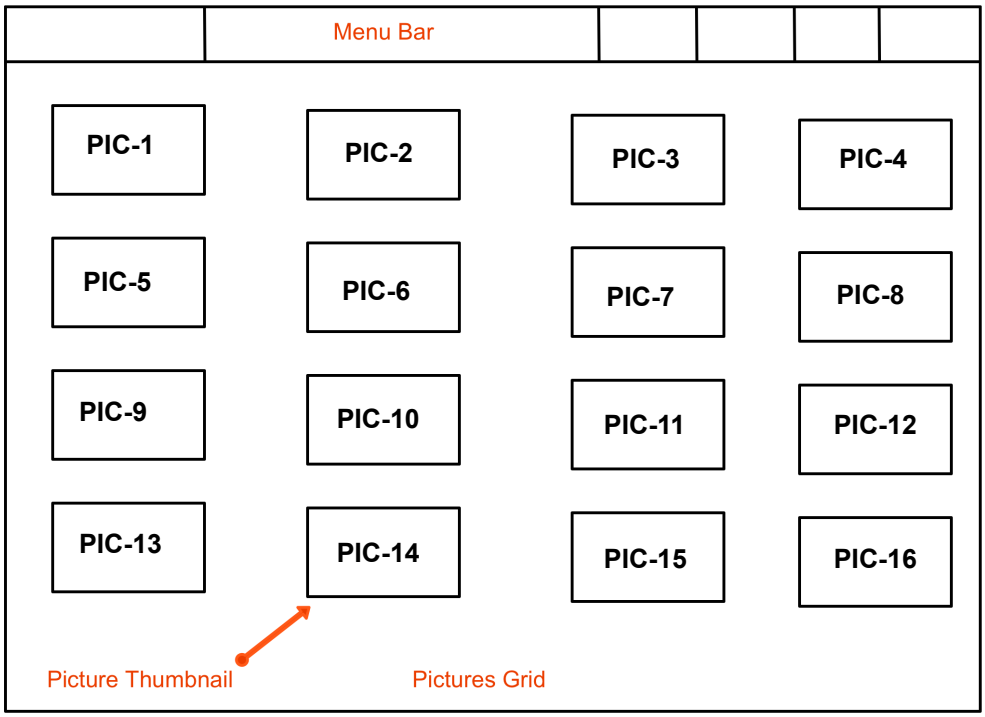


Figure 4.4 - Gallery page: Pictures grid list

The page shown on the figure 4.4 does not give any options to the user. But it will retrieve all the images that are stored on the external FTP server (Indiana University, 2014) and display the in a grid list on the page. However, the user will be able to click on the pictures thumbnails listed on the page, in order to pop out a window of the actual size of the picture.

Figure 4.5 show the streaming page. Once the user load this page, the streaming camera will be turn on and start transmitting live video on this page. The transmitted video will be actually a feed of images captured every second. The live streaming window will be refreshed every second with a new picture, so the output will seem like a video. The user has no options here as-well

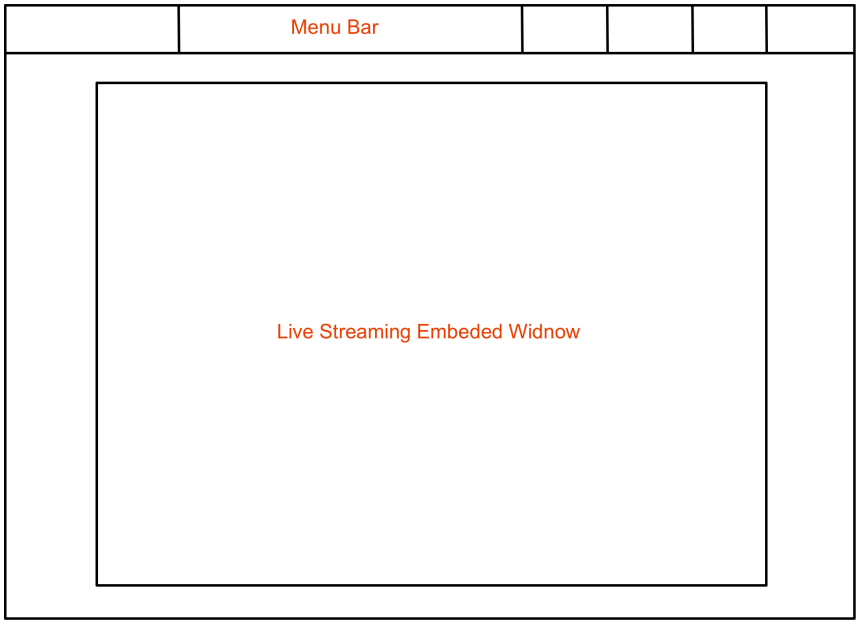


Figure 4.5 - Streaming page: Live streaming window

Figure 4.6 show the Admin page of the control panel. On this page the user is able to change any desired details or sign out from the system. This page is consisted of the following elements:

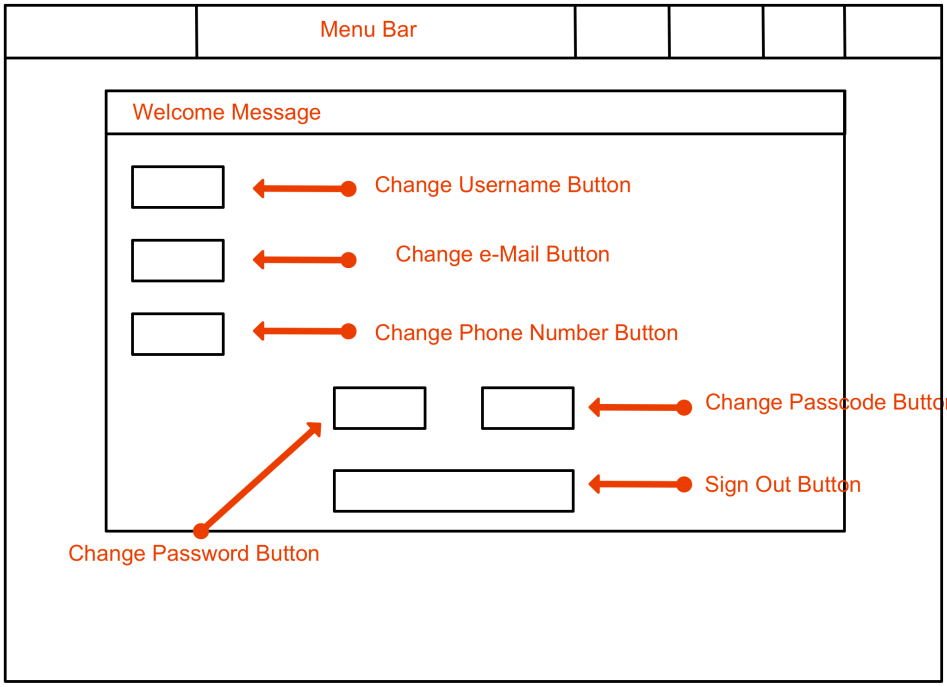


Figure 4.6 - Admin page: User's Details

- Change Username button - Once the user click the button, a popup window should popup asking for the new username and the password. Once the password is confirmed to be correct the system should update the database with the new username.
- Change E-mail button - The process will be the same as the username modification, but the user will be asked for the new e-mail address and the e-mail address will be changed instead.
- Change Password button - Once the user click the button, a popup window should popup asking for the current password, the new password and a confirmation of the new password. Once the current password is confirmed to be correct the system should update the database with the new password.
- Change Phone Number button - The process will be the same as the username modification, but the user will be asked for the new phone number and the phone number will be changed instead.
- Change Passcode button - The process will be the same as the username modification, but the user will be asked for the new passcode and the passcode will be changed instead.
- Sign Out button - This button kills the current browser's season, which leads the user to be logged out.

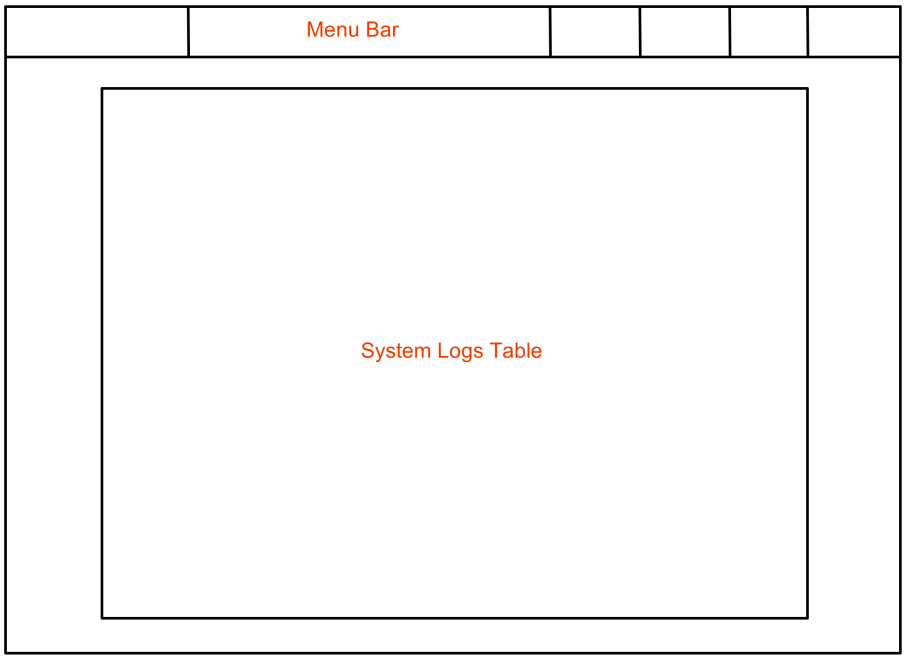


Figure 4.7 - Logs page: System Incident Logs

Figure 4.7 show the Logs page. On this page a table will be displayed, showing the latest 30 recent incidents of the system along with their timestamp. The incidents that will be logged will be the following:

- When the system was armed or disarmed
- When the user logged in or logged out
- When the power source change
- When a failed sign in attempted
- When the system started up or shouted down

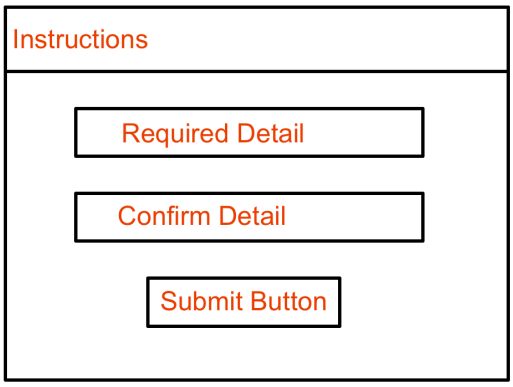


Figure 4.8 - Change User Details

Figure 4.8 shows an example of a pop-up window, in the process of changing the user details in the Admin page. Each window will contain a form, and the requested details will have some restrictions. For example the username should be 8 characters long. At the end of each form a submit button will be added, to submit the new details and perform the changes on the database.

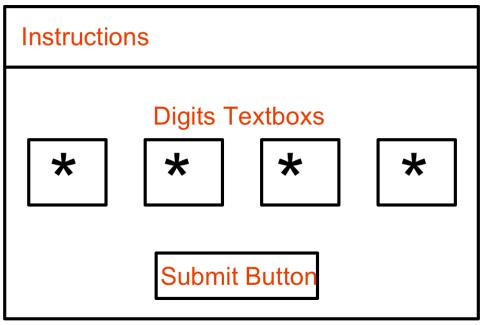


Figure 4.9 - Passcode confirmation pop-up window

Figure 4.9 shows the the pop-up window, that will ask the user to type the four digit passcode, in order to arm, disarm or shut down the system. The window will be consisted of text boxes that will ask the user to type the passcode each digit at a time, and a submit button that will submit to the system the user's request and the system will execute the corresponding command.

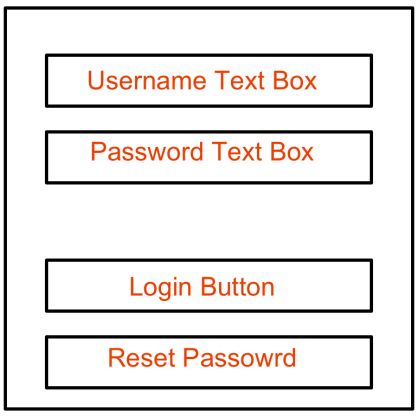


Figure 4.10 - Login window

Finally, Figure 4.10 shows the login window. The user will not be able to visit any page on the control panel, except the login page, if it is not logged in. The login page will have 2 text boxes that will request the user to type the username and the password. Also it will include a login button that will submit the username and password and the system will check if those details are correct and redirect the user to the actual control panel. If the user forgot it's password, a reset button will also be included, that will reset the password on the database, and send the new created password to the user's registered email address.

4.5 Database Design

The system needs to keep a record of the user's details, incident logs and the system status, so it needs a place to store them. Since the use of the database is not very extensive and only a limited amount of attributes are stored, the database can be small and no relationships will exist between the tables. Figure 4.11 bellow shows the database tables. The table have been designed in such way, that it can accept feature improvements on the system and each table is used for different purpose:

- Users Table - In this table the users details will be stored. The table is consisted by the attributes id, username, password, email, mNumber(mobile number) and digit which is the 4-digit password that arms/disarms and shut downs the system.
- Logs Table - In Logs table will be stored the system's incidents. This table i consisted only by 2 attributes; date time which is stores the data and time of the incident, and message which is the description of the incident. Since SQLite, do not support DateAndTime type of attribute, the data and time will be stored as text but the system will make sure that will format it to HH:MM:SS DD/MM/YYYY.
- Functions Table - On this last table will be stored the system status. This table has just the enabled attribute. Neither boolean type of field is supported by SQLite, so in its place Text type will be set. The system though will set the value to True if the systems is armed or False if it is disarmed in Text format.

USERS

Field Name	Type
id	Integer
username	Text
password	Text
email	Text
mNumber	Integer
digit	Integer

LOGS

Field Name	Type
dateTime	Text
message	Text

FUNCTIONS

Field Name	Type
enabled	Text

Figure 4.11 - Database Table used by the system

4.6 Hardware Considerations

The layout of the control panel should appear the same on various different devices, with different sized displays. This option can be provided by the Bootstrap Framework (Bootstrap, 2015). Bootstrap is designed to work on most computer, mobile and tablet displays. As the display of a mobile phone is smaller compared to a computer display, the layout of the website differs. This layout relatively repositions and resizes the elements of the page in such way that the user slightly notices the difference. For example instead of a horizontal menu on a mobile phone the menu become a drop down list menu. For the purpose of this project a smartphone and a tablet will be used for most of the testing as these devices will be used extensively from the client to manage the system.

4.7 Chosen Programming Languages

4.7.1 Python

Python is an widely used high-level programming language with large and comprehensive standard library. Since 2008, python rated the top 8 most popular programming language (TIOBE Software, 2015). What makes it so special is, it's code readability and it's syntax that allows program code to be written in fewer lines, unlike C++ and Java. On this project the main programming language will be Python which is the core programming language used on the Raspberry Pi and most of it's components libraries. Although, it allows the use of a database and it has various modules for different kind of projects.

4.7.2 HTML, CSS, Bootstrap and JavaScript

For the web based control panel, the combination of HTML, CSS and JavaScript will be used.

- HTML and CSS is the most popular and widely used programming language in the web development era. They are supported by every web browser and mobile device. The majority of web sites in the world wide web they are using HTML and CSS, which is make it easy to find sources and support by the web development communities, in case it is required.
- Bootstrap (Bootstrap, 2015) is a free open source set of tools that helps to create professionally designed websites. Actually, Bootstrap is an already written CSS file, that in order to create a website what remains to do is to write the HTML code and the Bootstrap CSS style will take care of the design. Due to time constraints for this project, Bootstrap will be used for the styling, in order to save critical time.
- JavaScript is a dynamic programming language which is commonly used for dynamic changes on a website (Flanagan, 2006, p. 1). JavaScript will be used to automatically update the system status on the web based control panel. Due to limited knowledge of the language, it will not be used for anything else.

4.7.2 SQLite

SQLite (Richard Hipp, 2015) is a rational database management system embedded in the inside of the program. SQLite is not a client server database engine, so it will need less memory resources from the system. The system it will need to store some data like system status, user's password e.t.c, and it is needed to be able to change them instantly if it is required. For that reason, SQLite determined to be the best option for the project.

4.7.3 Server Choice

- The web based control panel will be running on the Flask micro web application framework, which is written in Python. Since the main language of the project will be Python, Flask is the best option, for compatibility manners. Flask is being used from popular companies in the world, like Pinterest and LinkedIn.
- People saying prevention is better than cure. Storing the captured images on an external server will prevent memory loss and have intruder evidences somewhere safe in case of anything unexpected occur. For this feature a FTP Server (Indiana University, 2014) will be use to store the captured images. The server will be sponsored by HostPlan.gr for the purpose of this project. HostPlan.gr is a web hosting provider based in Greece and have servers located in the UK.

4.7 Conclusion

Now that the hardware is ready to be wired according to the plan, the design of the control panel has a shape, the database ready to be implemented and the development tools are chosen and ready to be used, the project will move forward to the implementation phase which is covered into the following section.

5. IMPLEMENTATION

5.1 Implementation Overview

The chosen methodology as has been discussed on section 3.5 it allows sprints that last 1 or 2 weeks. Following the chosen methodology, the implementation is divided into four iterations. At the end of each iteration, testing have been followed and the client have been asked to test the system on it is current stage and provide feedback.

The software of the project is structured in 3 different python scripts as follows:

- A script with most of the functions - This file includes most of the required features such as arm/disarm the system, send email update the database e.t.c. This file is not an executable file, the other 2 files call the required function located on this file.
- A script with checking loops - This file includes 3 loops running in threads. One loop checks whether a key is pressed on the keypad, the PIR's status, and the current power source. Also, this file calls all required functions from the first script.
- The control panel script - It is the web based control panel running on Flask. The user is able to make changes to the system through the control panel, and the required functions will be called from the first script.

The figure 5.1 bellow shows the data flow of those 3 files:

5.2 Chosen Components and Other Equipment

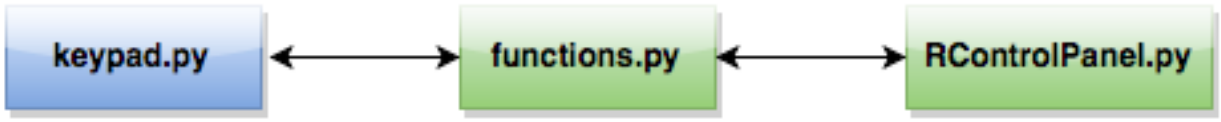


Figure 5.1 - Software Architecture

- An Adafruit Matrix Keypad which is a 3 x 4 keypad which the passcode will be entered in order to arm and disarm the system.
- An Adafruit LCD display which is a modern way for the Raspberry Foundation to display indications and make a more user friendly environment. The display is a blue and white colour and size 16x2.
- In order to wire the components on the Raspberry Pi, wires are needed. RPi Premium Jumber Wires come in three different types, Male/Male, Male/Female and Female/Female. The type of the wire is chosen according to the component's requirement and the situation.
- For soldering the components RPi Proto - Soldering Station 48W and the RPi Proto - Lead-Free Solder Wire is used.
- On a home security device the time is one of the most important things. When an image is captured the date and time must be always correct even if an internet connection failure occur. The Raspberry Pi from manufacture, has no memory on board like most personal computers. In case of internet failure it cannot be whether the

time is correct. For this case RasClock - Raspberry Pi Real Time Clock Module V3.0 is used, which is adding that extra feature on the Raspberry Pi to remember the correct time in any case.

- Raspberry Pi from manufacture has 8 I/O ports. Those ports are not enough for the purpose of this project. For that reason MyPi - Protect Your Pi - 32 I/O Point Expansion is used, which gives 32 ports, so even more components could be connected. Those 32 ports are enough and for future upgrades as-well.
- PIR Infrared Motion Sensor (HC-SR501) helps to sense movements in the room and go off the alarm.
- The Adafruit Stereo 3.7 W Class D Audio Amplifier (MAX 98306) with a combination of two 3” Diameter - 4Ohm 3 Watt speakers, gives voice indications to the system.
- The system has to be active even on a power failure case, for that reason a UPS component is needed. One of the best Raspberry Pi UPS on the market is the UPiS Advanced - Uninterruptible Power Intelligent Supply. This UPS includes a rechargeable battery which can keep the system active without power for several. It also includes Over-charge and Over-discharge protection and Over-voltage and Under-voltage protection.
- The system needs cameras to capture images of the intruder and streaming live video of the place that keeps secure. For the streaming purpose, the Raspberry Pi Camera Board is used, which is specially designed for the Raspberry Pi. It connects on the Raspberry Pi via CSI-w interface which is a High Speed Data Link, which keeps the video buffer low and the streaming timing will be more accurate. Because the streaming function is always on and uninterrupted, a second camera is used for capturing images. For this purpose Sony Playstation USB camera is used, and is provided by the client.

5.3 Iteration One

The first iteration of the project covers the wiring of the hardware following the initial design shown on the figure 4.1 and the basic requirements of the system features with priority 1 as is shown on the figure 3.1. This first iteration is the base for the future features will be added, which are depended on this initial stage.

5.3.1 Requirements

The aim of this iteration is to set up the “brain” of the system, which is the Raspberry Pi to communicate with the components. Later on, this setup has to be able to communicate with the first attempt on the coding part which is going to do the following:

- Armed and disarmed through the 4x3 keypad
- On movement detection take a picture, upload it on the external FTP and keep a copy of it locally
- On movement detection send notifications to the user through e-Mail and SMS
- Display indications on the screen

5.3.2 Implementation

5.3.2.1 Wiring the Hardware

Figure 5.2 shows the schematic that have been followed to wire the competent on the Raspberry Pi. As the design states, this is a conceptual project, so no circuit board is used but a breadboard instead. Breadboard makes wiring of the components easier and also rewiring is possible as it use jumper cables instead of soldering wires.

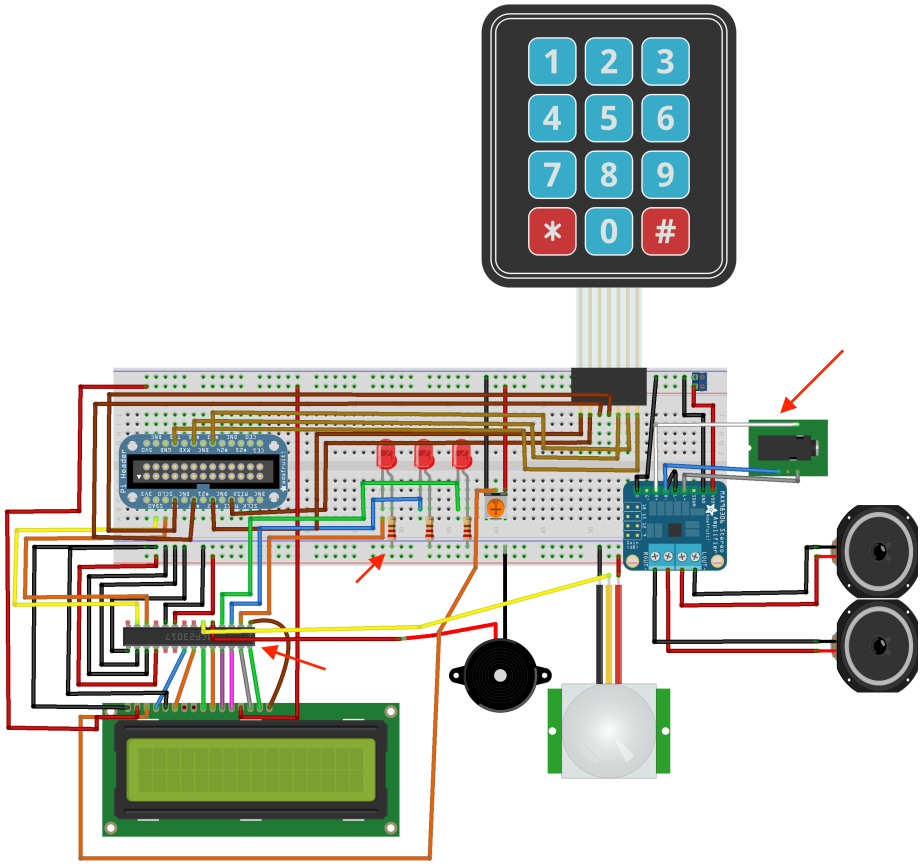


Figure 5.2 - Schematic Design of the

There are 3 pointed components on figure 5.2:

MCP Chip - The MCP23017 is the chip that increases the Raspberry Pi GPIO ports to full 16 I/O ports each. The MyPi - Protect Your Pi board contains 2 MCP23017 which increases the GPIO ports to 32 I/O ports. On this project though, only one MCP23017 will be used, but it might be useful for future improvements.

Resistor - The resistors are actually stepping down the current's power volts. The Raspberry Pi gives 3.5V, and the maximum input of each L.E.D is 2V - 2.5V, so the Raspberry Pi's current can damage the L.E.Ds. The resistor is located between the positive positive terminal of the L.E.D and the current power source and it steps down the volts, to avoid any damages on the L.E.D.

Audio In - This component is connected with the Raspberry Pi's 3.5" audio out jack, to the 2 wired speakers. This component delivers the output sound of the Raspberry Pi for example a siren sound, to the speakers which are wired on the system.

The design shown in figure 5.2 were followed for the wiring part and the result was as is shown on figure 5.3 with the addition of the USB camera which is connected on one of the available USB ports that the Raspberry Pi offers.

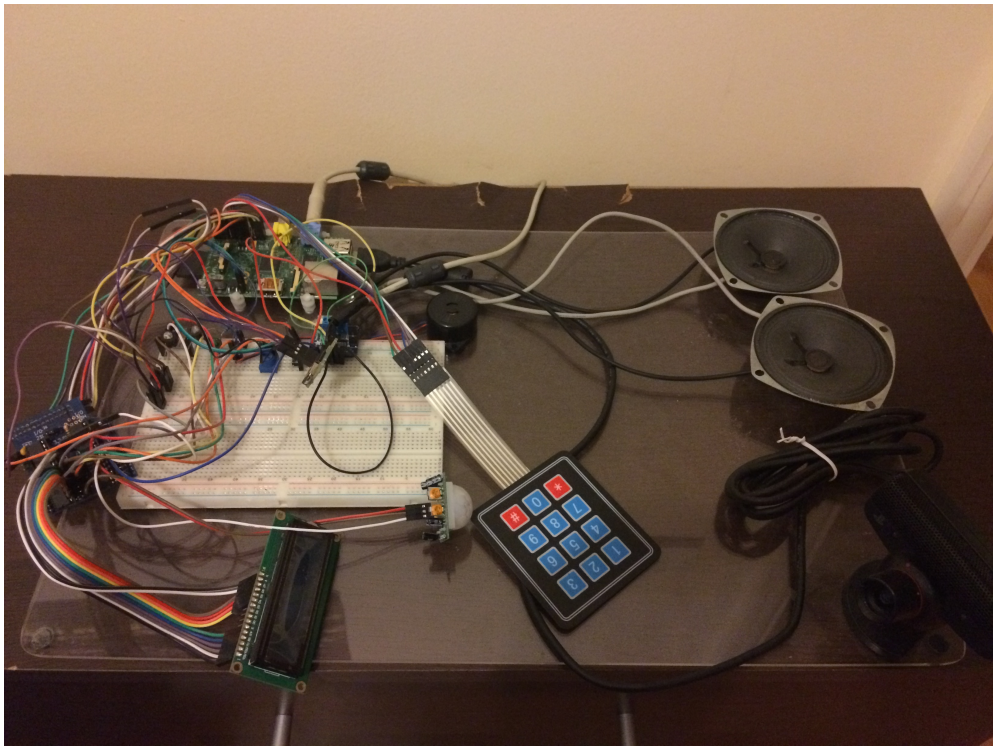


Figure 5.3 - Finishing up the wiring part

5.3.2.2 FTP Server Set-Up

As the first spring of the chosen methodology is to set up the work environment, setting up the FTP Server is the next step. On the external server it has been created a specific folder for the system to upload the captured picture. HostPlan.gr provided an FTP account which the system uses to login to the server and upload the captured pictures. Figure 5.4 show the details of the account have been provided.

Figure 5.4 - FTP account details on CPanel

5.3.2.3 Develop the Initial Features

The last task of this iteration was to develop the software features mentioned on section 5.1.1, which are as follow:

Keypad Loop

At the beginning startup.py file had 2 functions, keypad() and pir(). keypad() is a loop that checks if a key is pressed on the keypad. If 4 digits (which is the length of the passcode) is typed then the system calls the required functions from the functions.py file to check whether the passcode is correct and if its correct it checks whether the system is armed. If it is disarmed then the system will change it is status to armed and vice versa. In case the user typed the wrong passcode, after the verification the user will be prompted to try again. Figure 5.5 bellow shows the data flow chart of keypad() function.

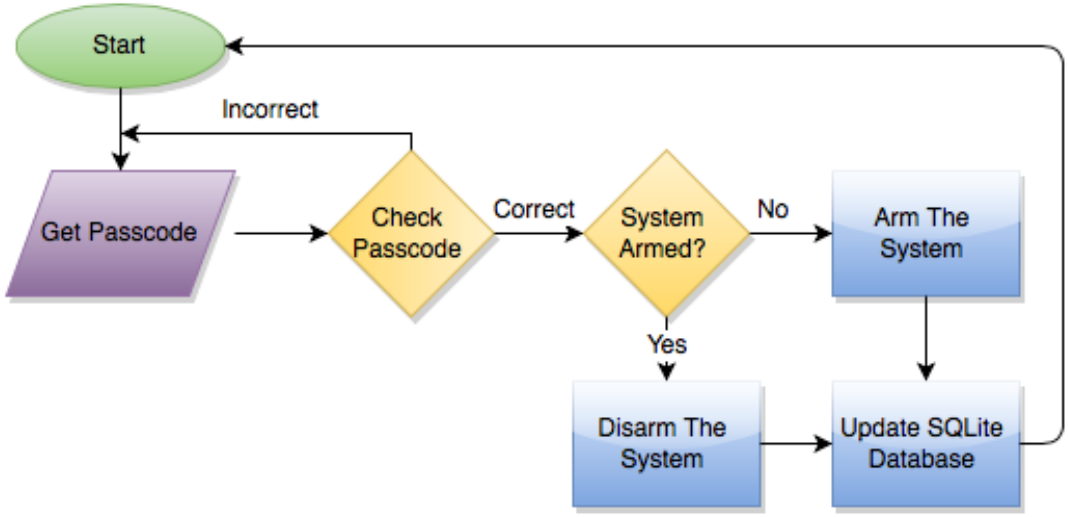


Figure 5.5 - keypad() Data flow Chart

PIR Loop

The pir() function checks whether the motion sensor detected a movement. The sensor alway returns a value to the system, if the value is 0 it means no movement detected. Therefore, the system will first check in the database whether the field enabled is True or False and determine if the system is armed. If it is, the system will check the value of the motion sensor if the value is other than 0, it will capture a picture of the environment, upload it on the external FTP Server that has been set up earlier, and then send a link of the picture via SMS and as attachment via e-Mail to the user. This process is always in an infinite loop. Figure 5.6 bellow shows the data flow chart of pir() function.

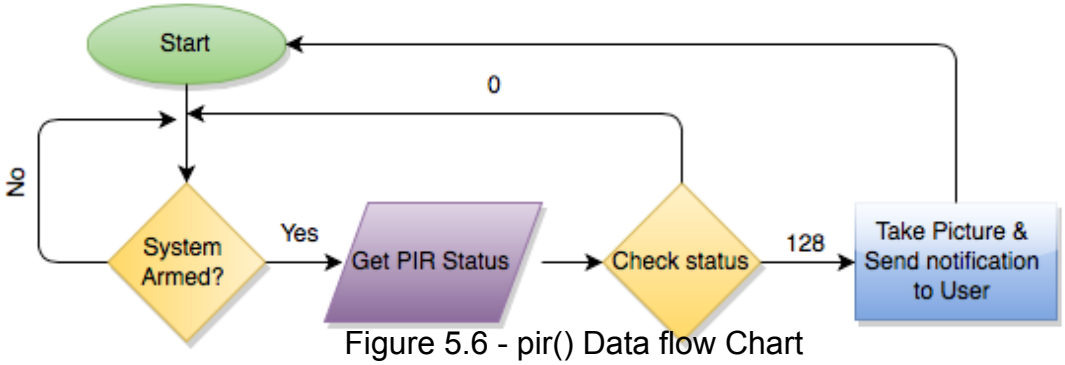


Figure 5.6 - pir() Data flow Chart

5.3.2.4 Sending Notifications

Once the function of uploading the taken image on the FTP Server is done, a link to the image is returned. That link is passed to the functions that sending notifications to the user.

For sending notification via SMS, Matt Hawkins python script (Matt Hawkins, 2012) is used. Matt's script is using TextLocal.com API (Txtlocal Limited, 2015), which is an online solution, that sends SMS through the Internet. In order to use TextLocal's API, it is required to have an account on their website and buy credits, like a normal cell phone provider. Matt's script has been set up with the user's account and it is gathering the user's phone number from the database. This script is implemented in a single function, which gathers the user's phone number from the database and sends a text message with the picture's link.

For sending notifications via e-mail, a sample code from the Python's documentation page is used (Python Software Foundation, 2015). This code code is also implemented in a single function and gathers the user's email address from the database, but instead of sending the picture as a link, like the SMS notification, it sends it as an e-Mail attachment.

5.3.2.5 Display Indications

Adafruit LCD display (Section 5.2) uses it is own python library, so that library has to be imported into the python file and then be initialised as is shown on figure 5.7.

```
from Adafruit_CharLCDPlate import Adafruit_CharLCDPlate
lcd = Adafruit_CharLCDPlate()
lcd.begin(16,1)
```

Figure 5.7 - Adafruit LCD Display initialisation

Writing on the display though, is quite straightforward, lcd.clear() clears everything from the screen and lcd.message("Hello World!") prints Hello World. An example can be shown on figure 5.8, which is the source code and the output on the display, underneath. Indications on the screen were use, to indicate to the user if a wrong passcode is given and once the status of the system has changed.

```
lcd.clear()
lcd.message("Hello World!")
```

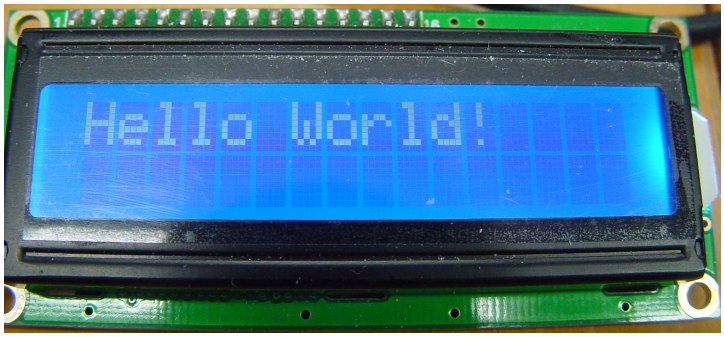


Figure 5.8 - Adafruit LCD outputs message

5.3.3 Evaluation

Finishing up the development of the first iteration, a few tests where followed to make sure that everything works as expected. The outcome of this iteration is the system to be able to detect unexpected movement and send e-Mail & SMS notifications to the user. Finally, the indications should guide the user through the system and acknowledge the system's status. The figure 5.9 shows the received e-Mail and SMS notifications, which means iteration one completed successfully. In addition, figures 5.10 and 5.11 show the indications on the Adafruit display while the user is going through the system using the keypad.

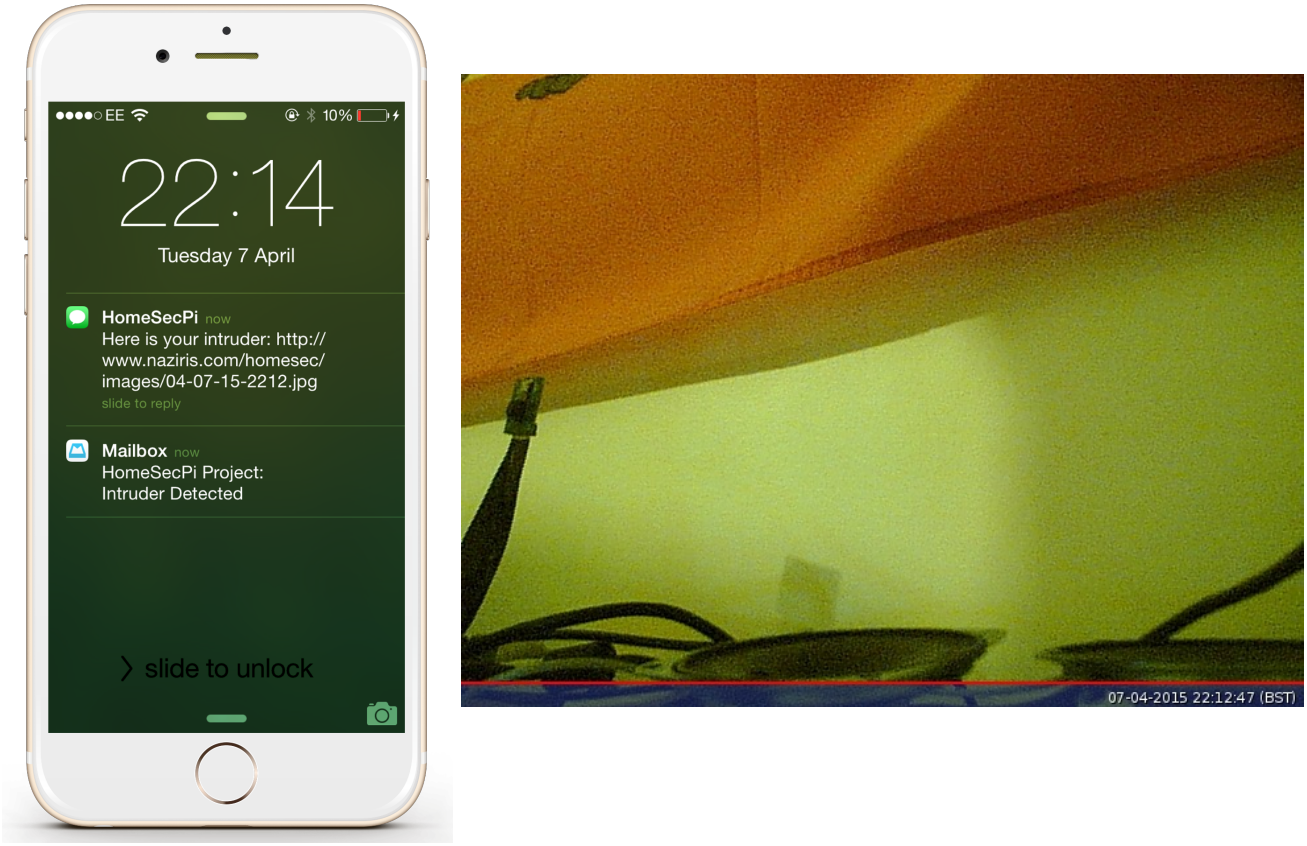


Figure 5.9 - Received notifications & Picture of the "intruder"

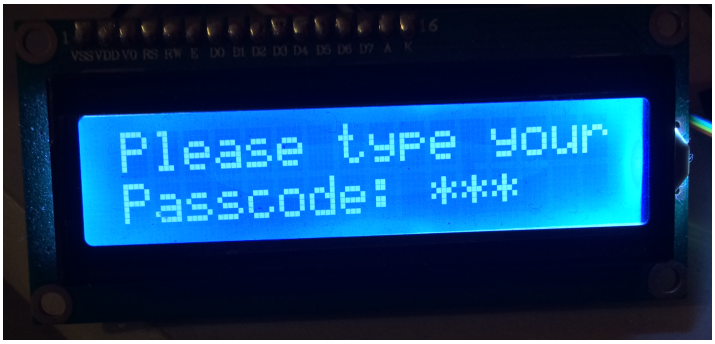


Figure 5.10 - User typed 3 digits on keypad

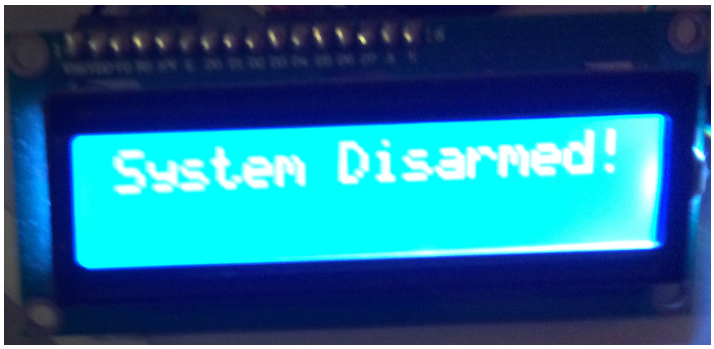


Figure 5.11 - User disarmed the system

5.4 Iteration Two

The second iteration of the project covers the implementation of the structure of the web based control panel with the requirements of the table shown on figure 3.2 with priority 1. This iteration actually sets up the environment of the Flask framework, so the rest features of the control panel expands it's functionality.

5.4.1 Requirements

The requirements of this iteration were as follows:

- Develop the structure of the web interface using Bootstrap
- The system should be able to be armed and disarmed through the online control panel.
- A live streaming page should be available on control panel.
- The control panel must have a page with a grid of all the pictures the system captured. Those pictures will be gathered from the external FTP server.

5.4.2 Implementation

5.4.2.1 Flask Framework

Flask framework as is a library of tools, it should be imported at the beginning of the python application in order to use them. Before starting building the HTML webpage, the flask application must be implemented. This application acts as a web server and loads HTML files on any web browser. It has been decided as a first step of this implementation, to build a Hello World page and then expand it to more pages. Figure 5.12 shows the code used for the simple Flask application.

```
import flask
app = flask.Flask(__name__)

class Index(flask.views.MethodView):
    def get(self):
        return flask.render_template('index.html')

app.add_url_rule('/',
                 view_func=Index.as_view('index'),
                 methods=["GET", "POST"])

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8068, debug=True)
```

Figure 5.12 - Example of Flask Code

5.4.2.2 HTML & Bootstrap

Having the Flask application ready, a HTML page must be implemented, so we can test that the application works. For this initial page Starter template of Bootstrap has been used, due to the time constrains. As Flask is sort of different than a normal web server, it has its own requirements. For example, the html files should be placed in a folder called

templates, and CSS files in another folder called static. Since Bootstrap is CSS and JavaScript files, they have to be loaded in the HTML page. In order to do that, the followed code should be written: **<link rel="stylesheet" href="{{url_for('.static', filename='bootstrap.min.css')}}">**

Once, all the required files of Bootstrap have been loaded and prepared the body of the page, such as menu, and some demo text, testing had been followed. The results of this testing can be seen in figure 5.13 bellow:

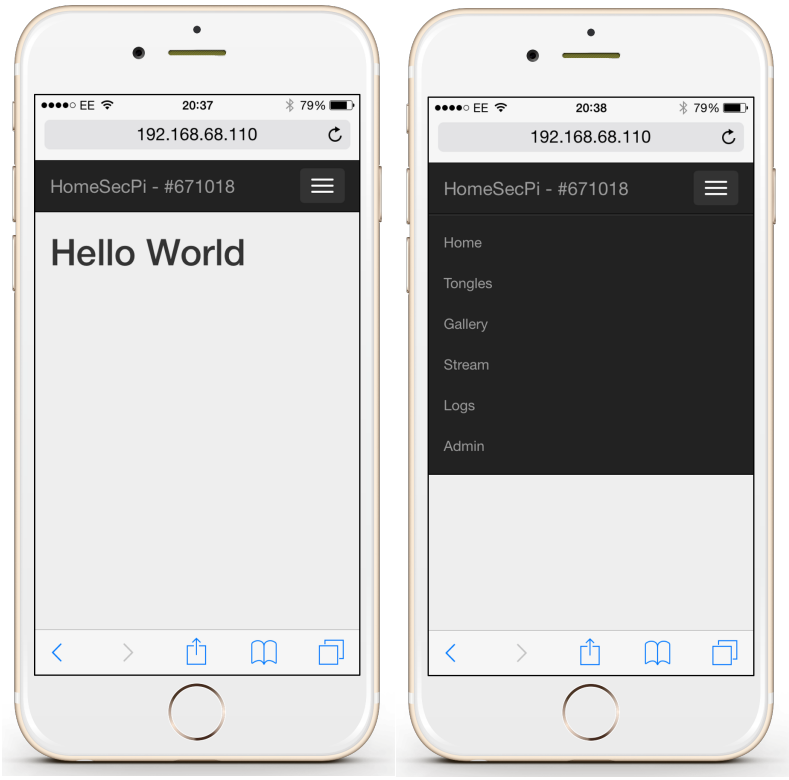


Figure 5.13 - Testing of the first implementation of the web page

5.4.2.3 The Grid of Pictures

On the initial plans of the project, it has been decided that the pictures should be loaded from the Raspberry Pi. But during the implementation it seemed more efficient to use the external server for that task, as the server is already a web server it self. This change makes the pictures load faster on the user's screen and also saves a lot of memory resources of the Raspberry Pi. For this task, David Walsh's PHP script (David Walsh, 2010) has been utilised. Dave's script automatically load a directory of pictures and display them in a grid list. The script was quite straightforward and, the only change on the initial code was to set the actual directory of the pictures. Then the script is uploaded on the web server, and created a new page on the flask application, whose body was an iframe redirecting to the link of the PHP script.

5.4.2.4 Streaming Page

Streaming live video from Raspberry Pi's camera is divided in two parts, the Python function which communicates with the camera and returns the video feed, and the HTML code that will display it on the web browser. The python function is an infinite loop that

takes one picture every second, and returns a video feed which is actually a row of pictures.This video feed is later returned on the HTML as variable. Since the web page is not updated automatically (unless user do it manually), the python function will refresh that variable with the new picture, on the back-end of the page. The whole process is not visible from the user, since the pictures refreshes every second, the user will have the impression that is watching a video.

5.4.2.5 Arm/Disarmed Button

The final task for this iteration is to create a button that will arm and disarm the system through the control panel. To prevent any unexpected events and for security reasons, by clicking that button a pop up window will appear asking the user's passcode and if the verification of the passcode pass, then it will change the system status. In addition, the user will be able to see the systems current status on the control panel as well. For displaying the system's status on the webpage the following (figure 5.14) JavaScript was implemented.

This script fetches the system's status from the database, by calling the function getSysStatus. Once the system status is fetched, it refreshes the page's element that displays it.

The button that changes the system's status follows the same data flow as is shown on the figure 5.5. User will enter the passcode and submit it, then Flask is passing the passcode to the function that does the validation and if it pass, it will call the functions that determine whether to arm or disarm the system. The actual process is the same like using the keypad, but Flask passes the passcode instead.

```
<script>
    $(document).ready(function() {
        setInterval(getSysStatus,10000); // call every 10 seconds
    });

    function getSysStatus() {
      //reload result into element with id "sysStatus"
      $("#sysStatus").load("/syStatus");

    }
</script>
```

Figure 5.14 - Javascript fetching system's status

5.4.3 Evaluation

By finishing this iteration the control panel is able to arm and disarm the system, stream live video from the the camera connected to the Raspberry Pi, and all the pictures can be fetched from the external server. Although, during the implementation several issues were raised that needed consideration and be fixed in the future iterations. The issues were as follows:

- The system must automatically clean up it self by deleting pictures older than 30 days, so it can save memory for future pictures.
- The Flask application as-well as the startup.py file must start automatically once the Raspberry Pi booted. So the user will have nothing to do than, turn on the power.

However, this iteration can be considered successful, since it passed all the tests that followed after completion and the client was pleased from the current result.

5.5 Iteration Three

The third iteration resolves the issues that have appeared in previous iterations and also develops the web based control panel even further by adding additional features.

5.5.1 Requirements

The requirements of this iteration were as follows:

- Secure the control panel by adding a login system. The user must be able to see the control panel only if its logged in with its user account.
- The user should be able modify or reset details like username and password.
- Add a “Force take picture” button where it will capture a picture and store it, no matter if the system sense unexpected movement.
- The system must automatically clean up it self by deleting pictures older than 30 days, so it can save memory for future pictures.
- The Flask application as-well as the startup.py file must start automatically once the Raspberry Pi booted. So the user will have nothing to do than, turn one the power.

5.5.2 Implementation

5.5.2.1 Login System

The login system implementation begins by creating the actual layout of the page following the initial design show on figure 4.10. Also the user should not be able to navigate to any page without a username and password verification. To achieve that with flask, the function shown on figure 5.18 should be created. This function verifies whether the user is logged by checking if the username is in an active session, if not, a redirection to the login page will occur, asking the user to login as shown on appendix H.7. When the user tries to login, the system must check the validity of the provided details. This verification achieved by comparing the given details with the details that are already stored in the database. In case the user forgot the password, a reset button is also placed on the login page, asking the user for the current username, the system checks whether the username is in the database and then resets the password and sends it via e-mail to the e-Mail address that is stored in the database.

```
def login_required(method):
    @functools.wraps(method)
    def wrapper(*args, **kwargs):
        if 'username' in flask.session:
            return method(*args, **kwargs)
        else:
            flask.flash("A login is required to see the page!")
            return flask.redirect(flask.url_for('login'))
    return wrapper
```

Figure 5.18 - Function that checks the user’s session

5.5.2.2 User Details Modification Options

The user should have the option to modify the details like phone number, e-Mail address, password e.t.c. This implementation is divided in 2 parts, the web page design, the python functions will updated the database. For the web page design the initial design showed on figure 4.6 were followed. The python part is quite straightforward. Once the user submits the new details a verification functions is called to check whether the given details are valid and meet the requirements, length for example. Once the checks have been passed, then another function is called that updates the database with the new values. To prevent any errors or issues, the user will sign out automatically from the control panel. This will make sure that the database has been updated success full with the new user's details.

5.5.2.3 Force Take Picture Button

This function takes a picture once the button is clicked, that picture should be uploaded on the external server, returns the link and appears on the user's screen. The functions for taking the picture, uploading it and return the link already existed from the first iteration, so somehow the web page should call that functions and return the link of the picture on a pop up window (modal). This has been achieved using the JavaScript code shown on figure 5.19.

```
$('#takePicturebtn').click(function() {
    $.getJSON('/picture', function(data) {
        var z = document.getElementById('pic');
        z.src = data.pictureLink;
    });
    $('#takePicture').modal('show');

});
```

Figure 5.19 - JavaScript code that takes a picture and returns the link in a modal

5.5.2.4 System Self Clean Up

This issue has been raised on the last iteration. The Memory Card attached on the Raspberry Pi filled up with pictures during testing and had no space to accept new pictures. The pictures had to be removed manually. As this is a standalone system, this have to be done automatically. To fix this issue a cronjob is created, that removes all the pictures that are 30 days old or later. Each picture is 250KB in size and the total memory of the system is 8GB, which means it can store 32,000 pictures. For a normal everyday use there is no chance for the system to capture that many pictures in a month. This cronjob will run every midnight, so it will really eliminated the chances of getting the memory filled up again. The same cronjob idea is used on the external server as-well. The figure 5.20 show the cronjob used on the Raspberry Pi.

```
#Run this as cronjon every days at 00:00 to delte pictures older than 30 days
find /home/pi/HomeSecPi/pictures/ -type f -mtime +30 -exec ls {} \;
```

Figure 5.20 - CronJob that remove pictures that are 30 days

5.5.2.5 Start on Boot

In order to start the web based control panel and startup.py application once the system boots, another CronJob should be created. This CronJob calls a .sh file on every system's boot. This .sh file contains UNIX commands that calls both required python files startup.py and RControlPanel.py. The figure 5.21 shows the .sh file and figure 5.22 the created CronJob.

5.5.3 Evaluation

```
cd /
cd home/pi/HomeSecPi
sudo python startup.py
sudo python RCControlPanel.py
```

Figure 5.21 - launcher.sh

```
#Execute launcher.sh on every reboot
@reboot sh /home/pi/HomeSecPi/launcher.sh
```

Figure 5.22 - CronJob runs on every start

By finishing the third iteration the control panel is login secured, no page than the login page can be viewed unless the user is logged in. The user's detail can be easily modified through the control panel and and instant picture can be taken through the control panel. Also, the issues raised on the second iteration are successfully solved. As the chosen methodology involves testing after every iteration, the implemented features tested and make sure no issued are raised. Once the system is tested and make sure everything is working as expected, the client was asked to test the system and give feedback before proceeding to the final iteration. The client is contented from the user friendly environment of the control panel which means that Bootstrap Framework is one of the best option it could be done.

5.6 Iteration Four

The fourth and final iteration covers the addition of the remaining features indicated on figures 3.1 and 3.2.

5.6.1 Requirements

By the end of this iteration the system has to meet the following requirements:

- Voice indication on every system interaction and voice instructions to guide the user
- The system should have L.E.D indications to indicate whether the system is powered and armed.
- A shutdown button should be available on the control panel where will allow the user to shut down the system.
- An external battery should be wired on the system to keep the system up in case of power cut.
- The system should keep system incident logs records on the database and display them on the control panel along with the time when the happened.

5.6.2 Implementation

5.6.2.1 Voice Indications

The voice indication actually are mp3 tracks playing during specific events of the system. For example, when the system is switched in the arm mode, at the end of the process a voice indication should sound from the speakers saying “System Armed”. The first phase of the development process is to decide in which occasions a voice indication is needed, what the indication will say and to play the mp3 indication on specific occasion. Figure 5.23 show the list of the voice indications have been decided. Those indications are converted from text to voice using IVONA Text to Speech software (IVONA Software, 2015). The system plays these mp3 tracks using mpg321 which is a very popular command line mp3 player. The command for playing a track is “mpg321 /home/HomeSecPi/sample.mp3”.

Voice Indication	Occasion
System ready.	When the system starts up.
System armed.	When the system switches to armed mode.
System disarmed.	When the system switches to disarmed mode.
Wrong passcode, please try again.	When a wrong passcode is given.
System is now set you have one minute to leave the building please exit now.	When the system switches to armed, and certain time space given by the system to leave the building.
System is shutting down.	When the system is going to shut down.
Motion detected please enter your passcode.	When motion is detected.
Siren sound	When motion detected and the correct passcode has not been given.
Authorities have been notified of your intrusion please surrender peacefully.	When the system notified the user.
System is running on battery powering.	When a power cut occur.
System is running on the external cable powering.	When the system switch back to normal powering.

Figure 5.23 - Voice Indications list

5.6.2.2 L.E.D Indications

The system has 3 wired L.E.Ds, a green which indicates that the system is on, a yellow which blinks when the user interacts with the system, for example press a button on the keypad. And finally a red one, which turns on when the system is armed.

5.6.2.3 Shutdown Button

User must have the ability to completely shut down the system in cases like the system must be relocated for example. For this feature a button is placed on the web based control panel which when clicked, a pop up window comes up on the user’s screen

asking the four digit passcode. Once the system verifies the validity of the passcode the required function for shutting down the system is called and the shut down process runs until the system shuts down completely. In order the button to call the shutdown function a couple of JavaScript lines are needed. The figure 5.24 show the JavaScript code is used.

```
$('#shutdown').submit(function() {
    $.getJSON("{{ url_for('index') }}");
```

Figure 5.24 - JavaScript code that calls

the shutdown function

5.6.2.4 External Backup Battery

The system should stay active even on the occasion of a power cut. The UPiS Advanced - Uninterruptible Power Intelligent Supply is used to serve this requirement. This UPS is actually a plug and play component, so no further development is needed for that to work. However, an issue was raised when the USB cable was powering the UPS and the system at the same time. The issue was inefficient voltage power. The USB cable was providing a maximum output of 0.1A - 1.3A and 5.00±0.25 V. The system it self without the UPS needs approximately 4.90V on idle mode. The UPS was impossible to keep charging and powering up the components at the same time, so for safety reasons was kicking the battery charging out. As it is already known the current is not always stable. During power drops on the USB Cable, the “juice” was provided on the Raspberry Pi was not enough, so it was keep reseting to protect it self. To solve this issue an external DC power supply with maximum power output 2A 12V was provided by the client to power the UPS through the external EPR module. Now the 12V±0.25 V is more than enough to keep the system alive and charge the battery at the same time.

5.6.2.5 System Logs Records

The system should keep logs of incidents along with the time happened so the user can keep track of what is going on into the system. For example what time the system has been disarmed, what time the user logged in into the system e.t.c. The process of this requirement is quite straightforward as-well. The idea is at the end of specific function to store in the database a record of what happened along with the time. For example when the user arm the system, at the end of the process the row “10/03/2015 11:43:24”, “System armed” will be be stored in the logs table. A new page is added into the control panel that it is fetching the last 30 rows of the table logs and display them on the screen.

5.6.3 Evolution

The fourth and final iteration is completed successfully and solved all the issues raised during the implementation process. The system developed, is now fully functional and stable. The next step is to take the system into an extensive testing procedure to make sure that the system is working as expected and no failures are raise once the system is deployed into the clients environment. The testing procedure is covered into the next section.

6. TESTING AND DEBUGGING

6.1 Testing Overview

This section covers the testing procedure carried out to determine the functionality of the final product. The testing methodology is followed was the Acceptance Tests (Agile Alliance, 2013). These tests test the application from the end user's perspective by checking each feature and mark it as accepted or rejected. To make sure that the results are reliable, the client tested the product for a week.

6.2 Testing Strategy and Results

The testing procedures is planned. The database is initialised at the beginning with dummy user details and at the end of the testing the actual user's details should take place into the database. Mr. Gregoriou was provided with system's guidelines to guide him through the system, and was asked to test the system for a week. Then he reported back the results of the testing. The provided guidelines can be found on the appendix J. The following figures show the results of the functionality of the system's hardware as well as the control panel. This test does not cover just the functionality of the product, but also the system's stability and reliability.

Test ID	Test Description	Expected Result	Result
T1	Arm the system using keypad	Red L.E.D should turn on and the system switch to armed mode. Display and voice indications should notify the incident.	Pass
T2	Disarm the system using keypad	Red L.E.D should turn off and the system switch to disarmed mode. Display and voice indications should also appear.	Pass
T3	Unplug the main power source	Display and voice indication should notify the incident.	Pass
T4	Plug the main power source	Display and voice indication should notify the incident.	Pass
T5	Alarm the system and walk into the room	System should ask the pass and if is not given to send SMS and e-Mail notification	Pass

Figure 6.1 - Hardware Testing Plan

Test ID	Test Description	Expected Result	Result
T7	Sign in into the control panel.	Redirect to the admin page.	Pass
T8	Arm the system using control panel's button.	Red L.E.D should turn on and the system switch to armed mode. Display and voice indications should notify the incident.	Pass
T9	Disarm the system using control panel's button.	Red L.E.D should turn off and the system switch to disarmed mode. Display and voice indications should also appear.	Pass
T10	Force take picture using control panel's button.	A picture should be taken and appear on the screen.	Pass
T11	Shut down system using control panel's button.	The system should turn off.	Pass
T12	Visit the gallery page.	A grid of the taken pictures should appear on the screen.	Pass
T13	Visit streaming page.	A live streaming video should start on the screen.	Pass
T14	Visit logs page.	The last 30 system's incidents should appear on the screen.	Pass
T15	Change all the details of the account.	The database should be updated with the new details, and each detail change a force sign out procedure should occur.	Pass
T16	Sign out from the control panel.	Redirect to the login page, and have no access to the rest control panel pages.	Pass

Figure 6.2 - Web Based Control Panel Testing Plan

6.3 Evaluation of Results

Upon the completion of the testing following the testing plan, was determined that the system is fully functional and also is reacting as expected to invalid and valid inputs. During the testing the system was stable and reliable as no crash occurred. Testing procedures during each iteration helped a lot the system to achieve this result. The systems does exactly what the client needs, however some small issues are identified and analysed in the next section.

7. EVALUATION

7.1 Evaluation Overview

This section covers the evaluation of the home security system against the functional and non functional requirements initialised in section 3. This section also covers limitations have been discovered and how they affect the system. In addition, this section will evaluate the approach and the chosen methodology and determine whether the right decisions were made.

7.2 Evaluation Against Functional Requirements

Figure 7.1 shows the list of the functional requirements determined in section 3. This table evaluated the software has been implemented against the functional requirements by testing the functionality of each requirement. Evidences of this testing process can be found on appendix H which are linked to the appendix reference table.

ID	Description	Status	Appendix Ref.
FR1	The system should be able to be armed and disarmed through the connected 3x4 keypad.	Satisfied	**Not Available**
FR2	In case of unexpected movement the system should take a picture store it locally and on an external FTP Server	Satisfied	**Not Available**
FR3	In case of unexpected movement the system should send e-Mail and SMS notifications.	Satisfied	**H.1**
FR4	The system should have voice indications - A voice will inform the user after the passcode is entered if the system is armed or disarmed.	Satisfied	**Not Available**
FR5	The system should have L.E.D indications - The L.E.D will indicate whether the system is powered and armed.	Satisfied	**H.2**
FR6	The system should display indications on a screen - The indications will be instructions to the user or current system status.	Satisfied	**H.3**
FR7	The system should have a UPS battery as an external power source.	Satisfied	**H.4**
FR8	The system should be able to be armed and disarmed through the online control panel.	Satisfied	**H.5**
FR9	A live streaming page should be available on control panel.	Satisfied	**H.6**
FR10	The control panel should be secured - The user should be able to see the control panel only if its logged in with its user account.	Satisfied	**H.7**

FR11	The user should be able modify or reset details like username and password.	Satisfied	**H.8**
FR12	On the control panel it should be a force take picture button where it will capture a picture and store it, no matter if the system sense unexpected movement.	Satisfied	**H.9**
FR13	A shutdown button should be available where will allow the user to shut down the system.	Satisfied	**H.10**
FR14	A page with a grid of all the images the system captured should be available - The images will be gathered from the external FTP server,	Satisfied	**H.11**
FR15	The system should keep a logs records - Those records will be systems incidents and what time they happened.	Satisfied	**H.12**

Figure 7.1 - Tested Functional

As the figure states, all of the functional requirements initialised in section 3 has been satisfied from the system.

7.3 Evaluation Against Non-Functional Requirements

This section goes through each of the non functional requirements and determines whether the project has satisfied them.

Platform

- The web based control panel must be supported on most modern browsers and mobile devices.
- Must be as compatible as possible with future browsers and mobile devices releases.
- The system should be easily expandable for desired added feature in the future.

Fully satisfied - The control panel is during testing confirmed that it can be supported on different web browsers like Apple Safari and Google Chrome, as well as different mobile devices like Apple iPhone, iPad and Samsung Galaxy S5 (See Appendix K). Also the way the system is designed it can be expanded easily by adding more sensors and cameras.

Usability

- The system must be as user-friendly as possible.
- The English language should be the main language of the system.

Fully satisfied - The system uses English as is the main and only language for now. And the client's feedback after testing the systems confirms that he found it very easy to use .

Performance

- The system should be stable and able to work 24/7 uninterrupted.
- The web based control panel should perform in a timely manner on most modern browsers and mobile devices.

Fully satisfied - Client's feedback after testing the system, confirms that the system's performance is satisfied as all the required functionalities listed on figures 6.1 and 6.2 passed the tests.

Portability

- The system must be able to work using Ethernet or Wi-Fi network.

Fully satisfied - The system fully operates using Ethernet or Wi-Fi connection and the IP address us known to the user.

7.4 Limitations

Battery Duration

The system can run on the external battery for certain hours. In order to determine the duration of the battery, the current mAh usage from the system should be known. UPiS Advanced - Uninterruptible Power Intelligent Supply has the ability to communicate via serial, so a simple script was written to keep records of the mAh usage on a text file every 10 seconds. From the thousand records the script exported the average mAh usage was 830mAh. Having the value of the system's mAh usage and the capacity of battery was easy to determine the duration. The appendix G show the calculations in detail. So calculated duration was 2 hours. This limitation may affect the usage of the system if the user will be away from a longer time. This issue can be easily fixed though, if the user connect a battery with more capacity

7.5 Approach and Methodology

The choice of the FDD methodology proof to be beneficial to the overall outcome as was making it easier to identify small issues resolve the with minimal effort. Also it was allowed modifications onto initial design during the implementation without compromising any requirements. However, this methodology was very time consuming as the design, implementation and testing were repeated 4 times. But, this is a price you pay and a risk you will take for a functional, stable and reliable result at the end.

Good time management and planing led the project being completed on time; however the plan could not always be followed due to other University's commitments. However, the plan was always the leader of the whole development. In addition, the time spent on researching different home security systems and hardware led to correct decisions and the choice of the Raspberry Pi turned out to be reliable and easy to work on. The final plan is located in appendix F.

The following section will conclude the whole prices of the project development and, add some future developments ideas which could be added if needed

8.CONCLUSION

8.1 Project Aims and Objectives

The first objective was to make a research and analyse various home security systems currently on the market and other different DIY approaches. Chapter 2 of this report covers the research process that were carried out before undertaking of this project. The research carried out led to the decision of using the Raspberry as the main board of the system which is the “brain” of the system. This objective can be considered achieved.

The second objective was “Discuss with the client and determine the systems requirements”. The requirements were discussed with the clients and a list of 15 functional requirements and 4 non functional requirements were determined for this project. Those requirements needed to be made clear as posable, so the developer and the client will expect the same outcome. It was also critical to aware the client that this is a conceptual and not the finished product, which can later be further developed. For these reasons, this objective can also be considered as achieved.

The third objective was the selection of the development life cycle methodology will be used to develop the product. The chosen methodology was Feature Driven Development, and the reasons of this choice can be found in section 3, the methodology is has also been evaluated in section 7. The chosen methodology can be concluded that it is one of the best options for this type of development because it allows the product to evolve with every iteration, and fix any issue that have been raised in the process. However, choosing the FDD methodology for larger developments will cost time and money which will led to serious issues, but for small individual projects like the current one there is no looking back on this option.

The fourth objective was “Select suitable components for the hardware part of the project”. The components were selected very carefully and make sure that are supported by the Raspberry Pi. The list of the components along with the cost can be found appendix K.

The fifth objective was “Select suitable software development tools for the web based control panel”. Python was the chosen main development language for this project as it is widely used on various Raspberry Pi projects. With a combination of python and the Flask module it was possible to run the web server that hosted the web based control panel. For the development of the control panel though, HTML, CSS and Javascript has been used, which are the 3 widely used tools for web development projects. Although, the system needed a place to store data, which is actually the “system’s memory”. For that SQLite has been chosen. As all the needed development tools were successfully chosen, this objective can be considered as achieved.

The sixth objective was “Put the hardware of the system together and prepare the hardware to run the software”. The components were carefully wired onto the Raspberry Pi having in mind the initial design of the system which is located in section 4 and the outcome of can be found in section 5. Therefore, this objective can be considered as achieved.

The seventh objective was “Develop a stand-alone application and a web-based application which can control the systems components”. The developed software have been developed, can operate the system Busing the wired keypad and also gives remote

access to the system which can be used to operate the system remotely. All the wired components communicate perfectly with the software, therefore this objective can also be considered as achieved.

The final objective was to document the whole process of the project. The objective is considered as achieved, as the development process of the home security system has been documented throughout this report.

The aims of this project were to “go through the development process of a home security system and produce a conceptual system, so it can be developed further as a commercial product for the use of the client”. Both aims and objectives were successfully met, therefore the development of this project can be considered as success.

8.2 Future Approach

The FDD methodology used has been evaluated in the previous section and can be concluded as a satisfactory methodology for this type of projects. The choice in methodology would not be different if the same project was done again or developed further

8.3 Future Developments

As the current system is a conceptual and it has only basic home security functionality, there is a lot of scope for future developments. Some of the possible added functionality and upgrades are as follows:

Prefboard - The breadboard is usually used to design a circuit, test it out and make sure that everything is wired correctly. In addition, a breadboard needs a lot space and a massive amounts of cabling. For more professional products, a good idea is a prefboard. Prefboard is a DIY circuit board that different electronic components can be wired on it and no wires are needed. The design of the prefboard have been actually made, but due to time constraints and availability of the prefboard it has not been implemented. Figure 8.1 show the design of the prefboard.

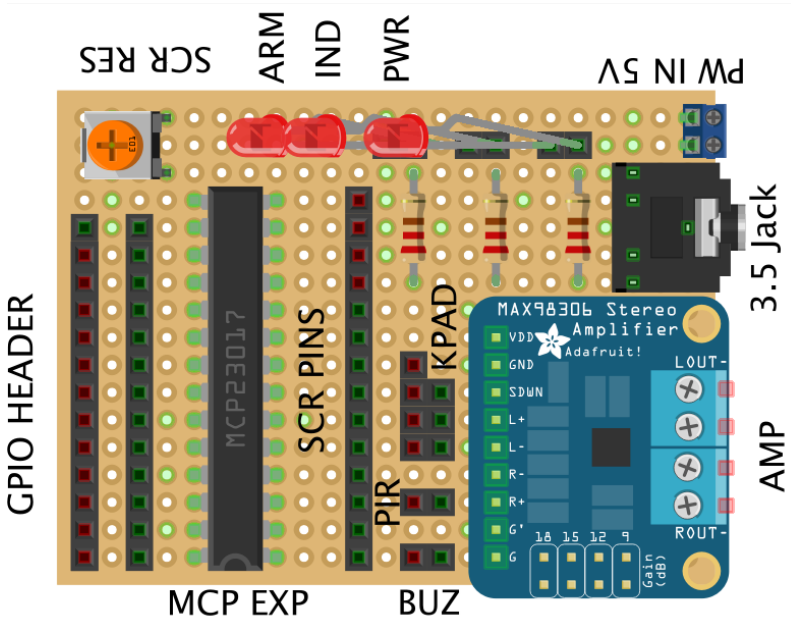


Figure 8.1 - Prefboard Design

Enclosure - The system needs an enclosure to protect the electronic components from humidity, liquids and any possible danger may cost damage to the system. Also it could be used as a mounter to keep the system in a desired place. The enclosure has also been designed but due to the time constrains and the availability of a 3D printer, it was not possible the actual implementation of the design. The enclosure design can be found on the appendix D.

Security - The whole development process covered just the implementation of such as project but not how to secure it. From the moment the system communicates outside from the local network (Remote Access) the system needs extensive security measures to be taken. The measures will make sure no other than the owner of the system could have access on it. Also, it could prevent sensitive data such as user details and pictures to be in the wrong hands like crackers. In order to secure the system a lot research should be curried out, but it is the first task of the list of the future developments.

System expansion and multi user capabilities - The current system is design to work just with one user and one movement sensor. A new feature could easily be added, is to allow different type of sensors an multiple users. Each user could also have a number of sensors. This system expansion is not easy but it is possible to be achieved using multiple Raspberry Pi devices in each room with a motion sensor and a camera. Each Raspberry Pi will run the same software that checks for movement and capture pictures, as the current system, but all the data like database values and images will be stored into a central server. The central server could be something more powerful than a Raspberry Pi, due to the extensive processes will make. On that server, all the Raspberry Pi will be connected using the local network, and they could be managed from the web based control panel that will also run on the central server. The whole system expansion is just a concept and nothing has been researched yet but, in case the client need this expansion the research, design and implementation will curried out.

8.4 Personal Conclusion

The project has been successfully developed on time by, trying staying on track on the initial plan. The process had a lot of challenges and risks like dealing with stuff that have never been associated with before, and the risk of failing meet the client's requirements. The development process not only developed the product but and my general knowledge as well. It gave me basic electronics knowledge and helped me to understand the back-end of a security system. If I had to undertake another project, if not the same I would definitely choose something similar, because the knowledge and the emotional challenges it offers, prepares you for real time experience in the outer world.

8.5 Final Conclusion

The project successfully covers the development process of a home security system which led to the a conceptual home security system to be produced. The project objectives and aims are met successfully; therefore the project can be considered a success. This section also covered future developments discussions and considerations. As technology evolves year by year and the price of the hardware decrease, soon home security systems will be used in every home. Not only is making the people live in the house to feel secure, but is fun to use and housed with security systems, most of the time are those that are out of the buglers list.

REFERENCES

Chowdhury, A. F., & Huda, M. N. (2011). Comparison between adaptive software development and feature driven development. Proceedings of the 2011 International Conference on Computer Science and Network Technology (ICCSNT) , 363-367.

Arduino Software. (2015). *What is Arduino?*. Retrieved from http://www.arduino.cc

Bootstrap. (2015). The world's most popular mobile-first and responsive front-end framework. Retrieved from http://getbootstrap.com

D-Link UK. (2012). DCS-6004L HD PoE Mini Dome Cloud Camera. Retrieved from http://www.dlink.com/uk/en/business-solutions/ip-surveillance/business-ip-cameras/dome-cameras/dcs-6004l-hd-poe-day-night-mini-dome-camera-with-ir-led

Flanagan, D. (2006). JavaScript: The definitive guide. Sebastopol, CA: O'Reilly.

Indiana University. (2014, November 6). What is FTP, and how do I use it to transfer files? Retrieved from https://kb.iu.edu/d/aerg

IngGaro. (2014, September). *Arduino anti-theft alarm shield - Hackaday.io*. Retrieved from https://hackaday.io/post/7360

Jeff Highsmith. (2013, August 2). *Making Fun | PiLarm: How to Build a Raspberry Pi Room Alarm*. Retrieved from http://www.makezine.com/video/pilarm-how-to-build-a-raspberry-pi-room-alarm/

Lemaker. (2014). Banana Pi - A Highend Single-Board Computer. Retrieved from www.bananapi.org

Panoptic Dev. (2014, March 11). *The Iterative Development Process* [Video file]. Retrieved from https://www.youtube.com/watch?v=7UTywn4vKLs

PrivateEyePi. (2013). Home Alarm System for Raspberry Pi - How it works - PrivateEyePi Project. Retrieved from http://www.projects.privateeyepi.com/home/home-alarm-system-project/installation/how-it-works

Raspberry Pi Foundation. (2015). What is a Raspberry Pi?. Retrieved from https://www.raspberrypi.org/help/what-is-a-raspberry-pi/

Richard Hipp, D. (2015, January). About SQLite. Retrieved from http://www.sqlite.org/about.html

Sheehan, L. (2013, September 2). [Image]. Retrieved from http://lexsheehan.blogspot.co.uk/2013/09/21-cfr-part-11-and-sdlc.html

TIOBE Software. (2015, April). The Coding Standards Company. Retrieved from www.tiobe.com

Van Cauwenberghe, P. (n.d.). Another look at incremental and iterative development [Web log post]. Retrieved from http://www.methodsandtools.com/archive/archive.php?id=14

David Walsh. (2010, April 6). Automatically Generate a Photo Gallery from a Directory of Images. Retrieved from http://davidwalsh.name/generate-photo-gallery

IVONA Software. (2015). IVONA Text-to-Speech. Retrieved from http://www.ivona.com

Matt Hawkins. (2012, August 30). Sending SMS Text Messages Using Python. Retrieved from http://www.raspberrypi-spy.co.uk/2012/08/sending-sms-text-messages-using-python/

Python Software Foundation. (2015, April 3). email: Examples - Python 2.7.10rc0 documentation. Retrieved from https://docs.python.org/2/library/email-examples.html

Txtlocal Limited. (2015). Bulk SMS Marketing Online for UK & International Texts. Retrieved from http://www.textlocal.com

Ronacher, A. (2014). Flask (A Python Microframework). Retrieved from http://flask.pocoo.org

Agile Alliance. (2013). Acceptance Testing. Retrieved from http://guide.agilealliance.org/guide/acceptance.html

Appendix A - PID

School of Computing final year project

Kyriakos Naziris

(PJE40)

Project Initiation Document

Home Security System based on Raspberry Pi

Project Initiation Document

1. Basic details

Student name:	Kyriakos Naziris
Draft project title:	Home Security System based on Raspberry Pi
Course:	Computer Science
Client organisation:	-
Client contact name:	Gregoris Gregoriou
Project supervisor:	Athanasios Paraskelidis

2. Outline of the project environment and problem to be solved

The client would like to set up a home security system in their storage. The system will allow the client to remotely and locally operate the system using a web based control panel or an embedded control system. The system must guard and monitor the storage and notify the client if there is any unexpected movement. The web based control panel should be compatible on desktop and mobile environments, so the client will be able to control the automation system on the move.
The client has asked that they should be able to watch live web streaming video, which will allow them to monitor the area at anytime and in case that they informed about an unwelcome intruder.
In addition, in order to prevent power and internet connection failures, a backup battery will be wired on the system, which will be able to charge and discharge when is needed.

3. Project aim and objectives

The overall aim of this project is to develop a home security system, which can guard and monitor the client's storage. To meet this aim I have to meet the following objectives:

- Set up the hardware and prepare it to accept the software will be developed.
- Develop the software will run on the system.
- Develop the appearance of the web based control panel.
- Program the software of the system and the web based control panel to communicate with the hardware of the system.
- Develop the database the system will use.
- Deploy the system at the client's storage

4. Project deliverables

I am going to develop an artefact which will be a home security system which will guard and monitor a specific room. The user will be able to operate the system either from home or remotely.

The project report will be made up of several sections:

- Analysis and research – This will be report that will show in detail the user specification, a schedule, a literacy review, the methodologies I may use, testing approaches, etc.
- Design – In this stage I will determine what parts of the system will need to be completed first, how the system will interact with the components, what the web control centre will look like etc.
- Implementation – In this stage the system artefacts are developed.
- Testing - Documents will be produced on everything that is tested, whether it was user tested or tested by myself. These will describe what went wrong, why it went wrong, or if nothing went wrong.

5. Project constraints

There are several constraints during the implementation of the artefact, and they are as follows:

- Time constraints – The project has to be completed and handed in by 26 April 2015
- Other unit commitments – I have to work on the project alongside doing courseworks and examinations for other units
- Some components might be hard to be found or expensive
- User-friendly – The application and the system's control centre must be user-friendly. In terms of usability and readability.

6. Project approach

At the beginning of the development some time will be spend on researching on various home security systems on the market and DIY approaches. Also, I will discuss with the client the needs and requirements of the project and make sure I understand the client's needs. Failing to produce a system that the client needs would be disastrous.

Later on, I must decide the choice of the development board (Raspberry Pi, Arduino, Banana Pi). This will depend on the features required by the system and if it supports the needed components.

I have decided to use the iterative and incremental development model. The reason of this selection is because is suited for small individual project that have to be developed in very short period of time. It is similar to the waterfall model but in addition it allows backtracking, which is not possible to waterfall methodology. Backtracking allows changes to the initial design and implementation, if needed, allowing more flexibility on the development process.

7. Facilities and resources

For the coding part, I am going to use the Sublime Text editor for Mac, which support various types of mark-up coding, such as Python, HTML, CSS etc. This tool is free to use and can be accessed immediately. In addition, I will use the new mobile application lab, that has opened up at the university which will be really useful to test my web application on various mobile and tablets devices to make sure that it will work on most of the personal mobile devices.

8. Log of risks

There are several risks I may encounter while doing my project.

- Producing a system that does not meet the client's needs. Discussing the requirements of the system with the client will hopefully prevent this.
- I could fall sick while doing this project, disrupting the amount of time spent on the project. To reduce the effect on my project, I will allow time in the schedule for such problems.
- Wiring/Components risk – The components or the wiring might fail and replacements will be needed. Which could also disrupting the amount of time spent on the project.
- The client cannot use the system – The system will be as user-friendly as possible. I am also going to produce a user guide.

9. Starting point for research

I can find many journals/papers using ACM and IEE Xplore Digital libraries. There are also many books and online resources for the following:

- Python software development tools. Deciding what board I should use based on features and usage statistics.
- How previous/current home security systems work.
- How a software communicates and controls hardware

10. Breakdown of tasks

- Research analysis techniques – How to get the most from clients.
- Gather the requirements from the client and get a clear overview of the project.
- Research similar home security systems available on the market and in the DIY era.
- Study the hardware will be used and manage to get familiar with it.
- Research on what makes a software environment user friendly.
- Design, implement and test home security system
- Deploy the completed artefact in the client's place

11. Project plan

Please see attachment.

12. Legal, ethical, professional, social issues

- I will ensure that the artefact is to a good working standard before releasing it.
- Copyright and plagiarism issues will be prevented by ensuring that the work I produce is my own or referenced correctly.

Signatures

	Signature:	Date:
Student		
Client		
Project supervisor		

Appendix B - Ethics Certificate

University of Portsmouth

Certificate of Ethics Review

Project Title:	Home Security System based on the Raspberry Pi
User ID:	671018
Name:	Kyriakos Naziris
Application Date:	21/04/2015 13:57:34

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative for the School of Computing is Carl Adams

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

All projects involving human participants need to offer sufficient information to potential participants to enable them to make a decision. Template participant information sheets are available from the:

- Univeristy's Ethics Site (Participant information template).

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

SchoolOrDepartment: SOC
PrimaryRole: UndergraduateStudent
SupervisorName: Athanasios Paraskelidis
HumanParticipants: Yes
ParticipationBeyondAnsweringQuestionsOrInterviews: Yes
ParticipantInformationSheets: The participants will answer questionnaire or give feedback based on their experience with the product.
ParticipantConfidentiality: The only published details will be their name, all other details will be private and will be used only for research and development purposes.
InvolvesNHSPatientsOrStaff: No
NoConsentOrDeception: No
CollectingOrAnalysingPersonalInfoWithoutConsent: No
InvolvesUninformedOrDependents: No
DrugsPlacebosOrOtherSubstances: No
BloodOrTissueSamples: No
PainOrMildDiscomfort: No
PsychologicalStressOrAnxiety: No
ProlongedOrRepetitiveTesting: No

Certificate Code: 4814-8AB6-ED0F-78EA-7CED-DD53-0D02-C9AF Page 1

PsychologicalStressOrAnxiety: No
ProlongedOrRepetitiveTesting: No
FinancialInducements: No
PhysicalEcologicalDamage: No
HistoricalOrCulturalDamage: No
HarmToAnimal: No
HarmfulToThirdParties: No
OutputsPotentiallyAdaptedAndMisused: No
Confirmation-ConsideredDataUse: Confirmed
Confirmation-ConsideredImpactAndMitigationOfPontentialMisuse: Confirmed
Confirmation-ActingEthicallyAndHonestly: Confirmed

Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy.

Supervisor signature:

Date:

Certificate Code: EED3-1D59-3CD4-A8BB-143D-F398-3E0C-69AD Page 2

Appendix C - Questionnaire

Sample client satisfaction questions

1. **What is the exact name of the company you are representing?**

   Multitech L.T.D

2. **What is your name and your position in this company?**

   My name is Vasilios Irodotou and I am a Security systems sales manager

3. **Has the company you are representing, a franchise in the UK?**

   Not at the moment

4. **What is your current best selling home security system ?**

   Is the D-Link DCS-6004L HD PoE Mini Dome Cloud Camera

5. **Which of the following hardware features does it offer?**

   - ☑Movement Sensor
   - ☐ Keypad
   - ☐ Display
   - ☐ Speaker
   - ☑ L.E.D Indicators
   - ☐ Power redundancy
   - ☐ Internet Connection Redundancy
   - ☑ Camera

6. **Which of the following software features does it offer?**

   - ☐ Arm and disarm the system from home using an embedded keypad
   - ☑ Arm and disarm the system through an online control panel
   - ☑ Force takes picture
   - ☑ Live video streaming
   - ☐ Voice indication
   - ☐ SMS notifications
   - ☑ e-Mail Notifications
   - ☑ Store picture locally
   - ☑ Store picture on external FTP Server

7. **How does it work?**

The system captures a picture of the place that monitors, and it keeps checking were the live picture is different form the one it already captured. In reality it is comparing pixels, which is, it is “movement sensor”. In case of unexpected movement sends to the owner an email informing about the incident. In the meanwhile, it stores the picture in the SD memory card and it can upload it in an external FTP server if the owner whish to.

8. **Does it work offline? Please explain?**

No, it does not. This product has been design to work only with an active Internet connection. In order to be able to arm or disarm it an active Internet connection is required.

9. **Does this product remain active in case of power cut?**

No, it does not. The main source of power is through the home’s central power source. If that source for a reason is not able to provide the needed form the product power, then the system will shut down instantly.

10. **How much does this product costs for an individual client?**

The total cost of this product is around 250 pounds, including 2 year International warranty.

11. **Does the price include the installation fee?**

No it does not. The installation fee is dependent on the person will install it. Multitech L.T.D, does not provide such as services. Although, the installation process is pretty simple as it has been design for DIY installation.

12. **Is this system easily expendable?**

This product cannot be expanded. Each system work independently.

13. **How can this product be controlled?**

It can be controlled through the online control panel or the mobile application provided for iOS and Android devices. In case there are more than one of those product connected into the same network, you can manage all of them through the same mobile application

14. **Does the client required to have any kind of pre-installation at home in order this product work?**

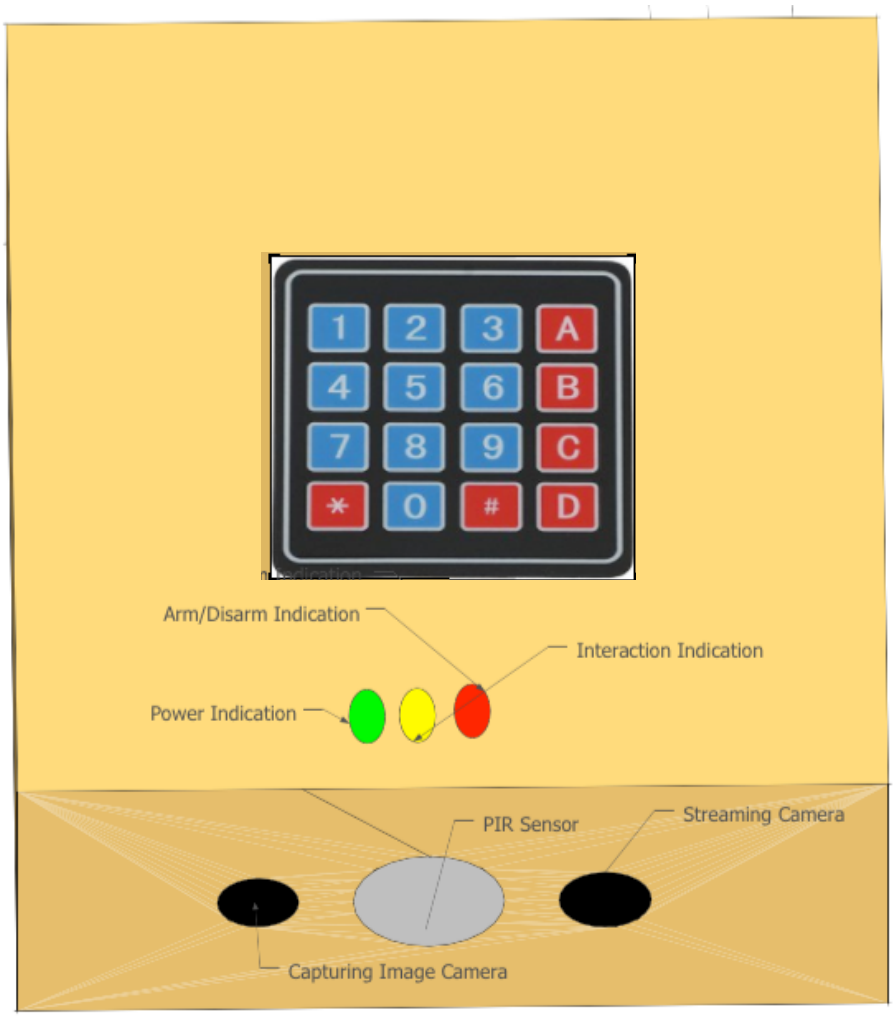
Not really. Just an active Internet connection, a power socket and a SD Memory card are enough.

15. **What feature do your client occasionally ask about?**

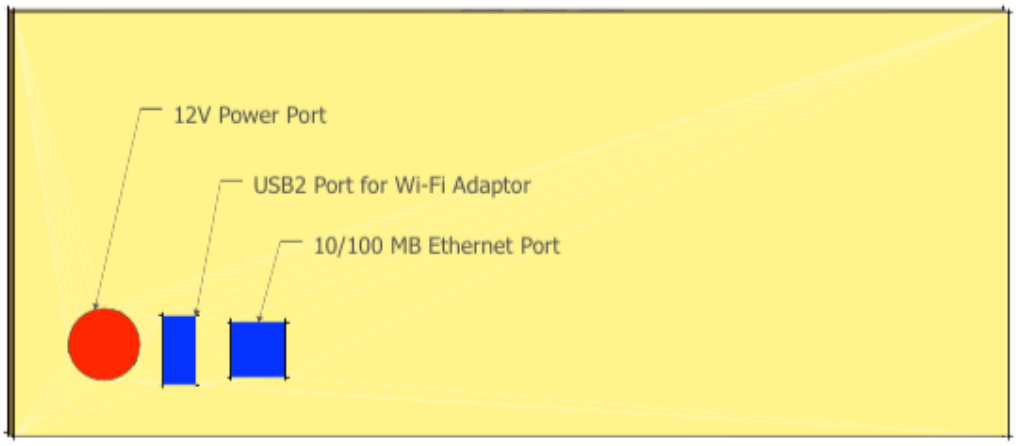
A good sensor and camera quality (12MP), the new colour night vision that is price is still expensive at the moment and microphone to capture voice.

Thank you for your time!

Appendix D - Enclosure

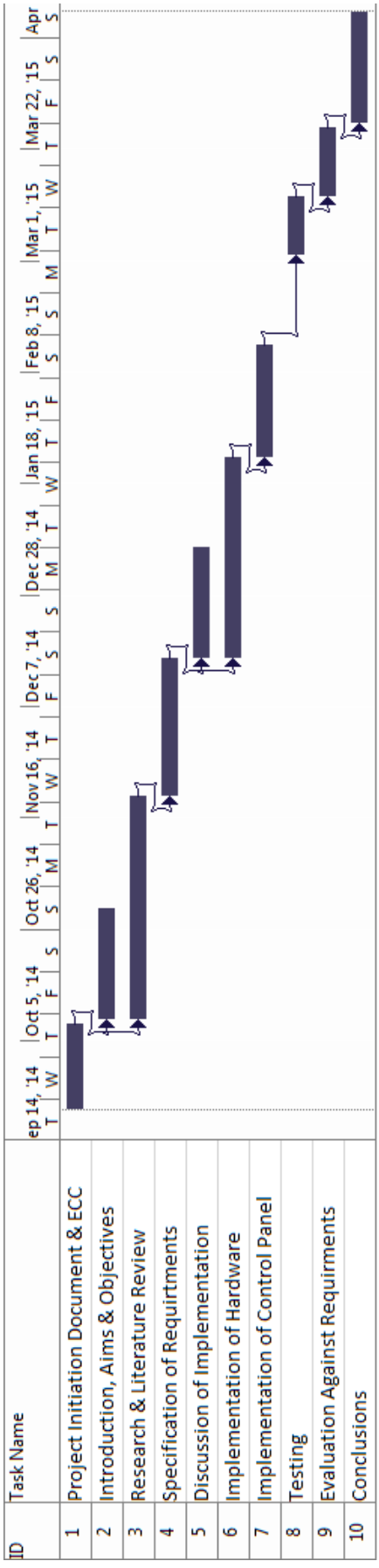


Front and Upper side of the

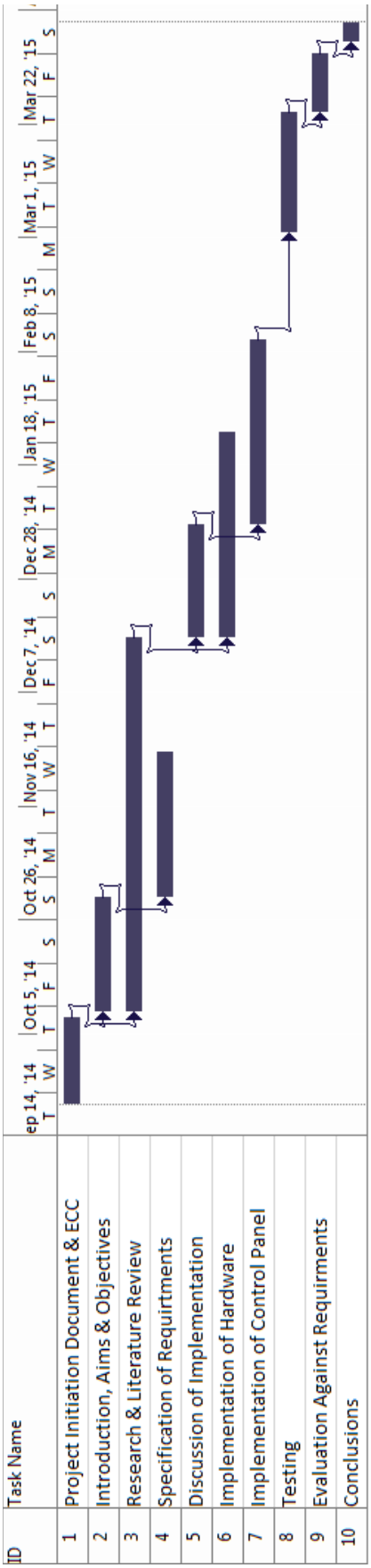


Back side of the

Appendix E - Initial Gantt Chart



Appendix F - Final Gantt Chart



Appendix G - Battery Calculation Appendix

As the script was running game an average of the power consumption of the system which was 830mAh, and the battery capacity was known from factor that is 2600mAh, it was possible to calculate the battery life in hours using the following formula:

$$BatteryLife(hours) = \frac{BatteryCapacity}{AverageConsumption} \times Efficiency$$

Battery life calculation formula

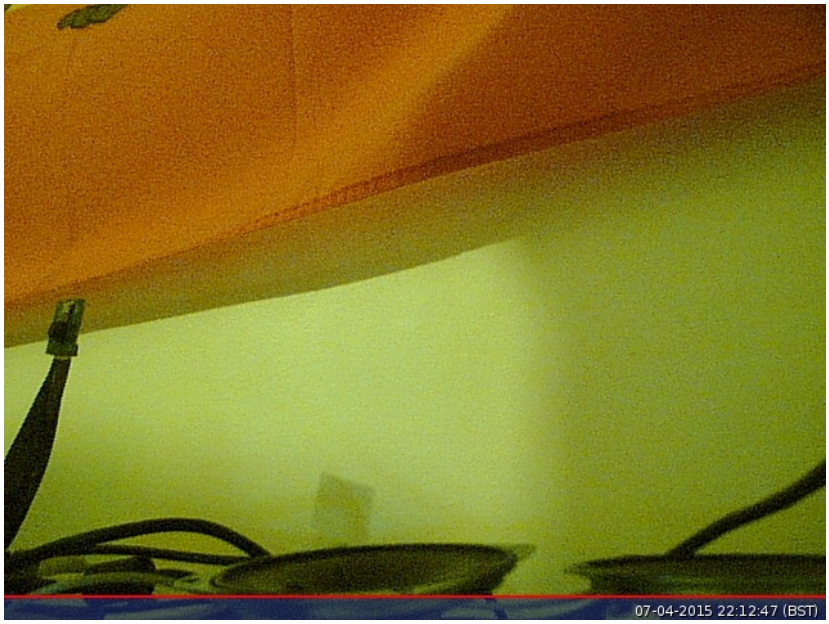
The Efficiency which is stated on the formula is t he factor of 0.7 makes allowances for external factors which can affect battery life. By multiplying the devision of battery capacity and average power consumption, allows external factor such us system power consumption more than 830mAh in case the system needs it. So the battery life calculation stills be valid in such as cases. The formula stated above followed for the calculation and the result was that the system can last 2 hours running on battery.

$$BatteryLife(hours) = \frac{2600}{830} \times 0.7 = 2.06818 \Rightarrow Round(2.06818) = 2hours$$

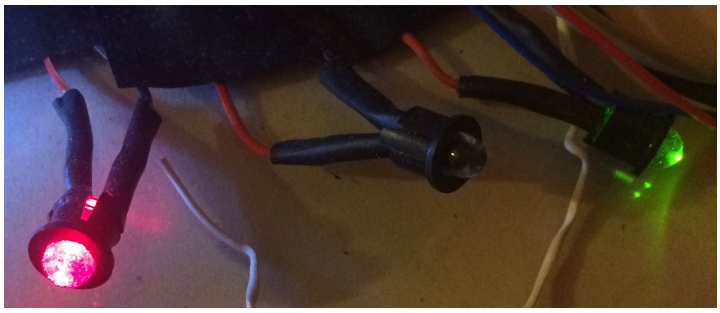
Implementation of the battery life calculation formula

Appendix H - Testing Evidences

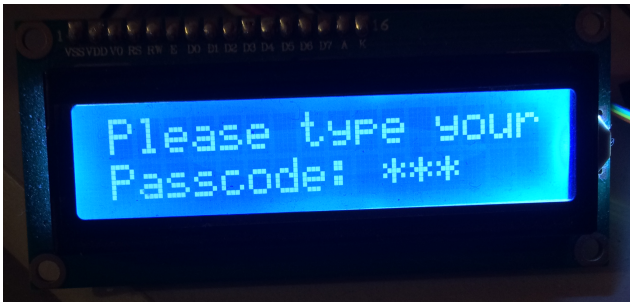
H.1 - When unexpected movement was detected while the system was in armed mode, a picture has been taken, SMS and e-Mail notifications were sent.



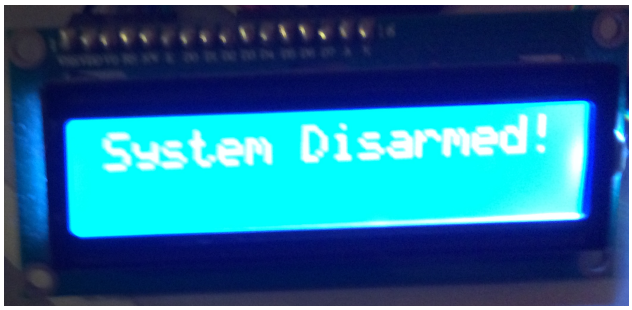
H.2 - The L.E.D indicators indicate that the system is on and armed. The red L.E.D means armed and the green one that the system is currently on. The L.E.D in the middle, indicates that the system is currently processing smoothing. Since nothing is under processing, the indicator is off.



H.3 - The pictures bellow indicate some of the display indication that the system indicate to inform the user for the current status of the system and guide him through the system.



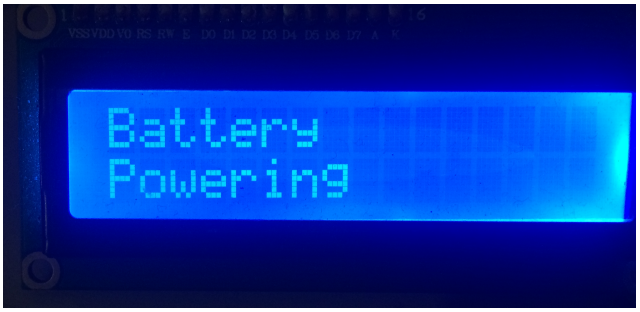
User typed 3 digits on keypad



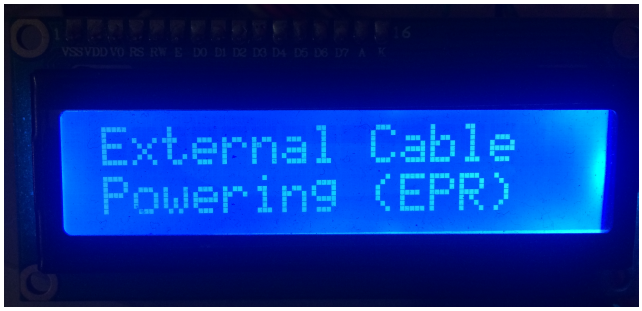
User disarmed the system

Initializing system before is set to armed

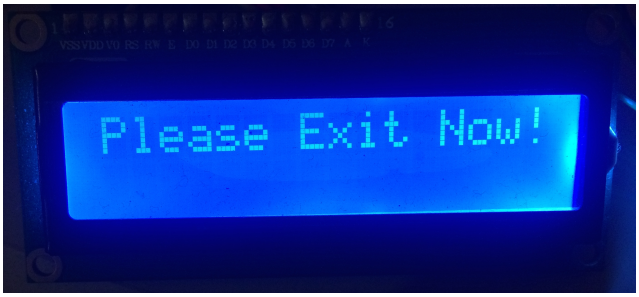
User armed the system



When a power cut occur

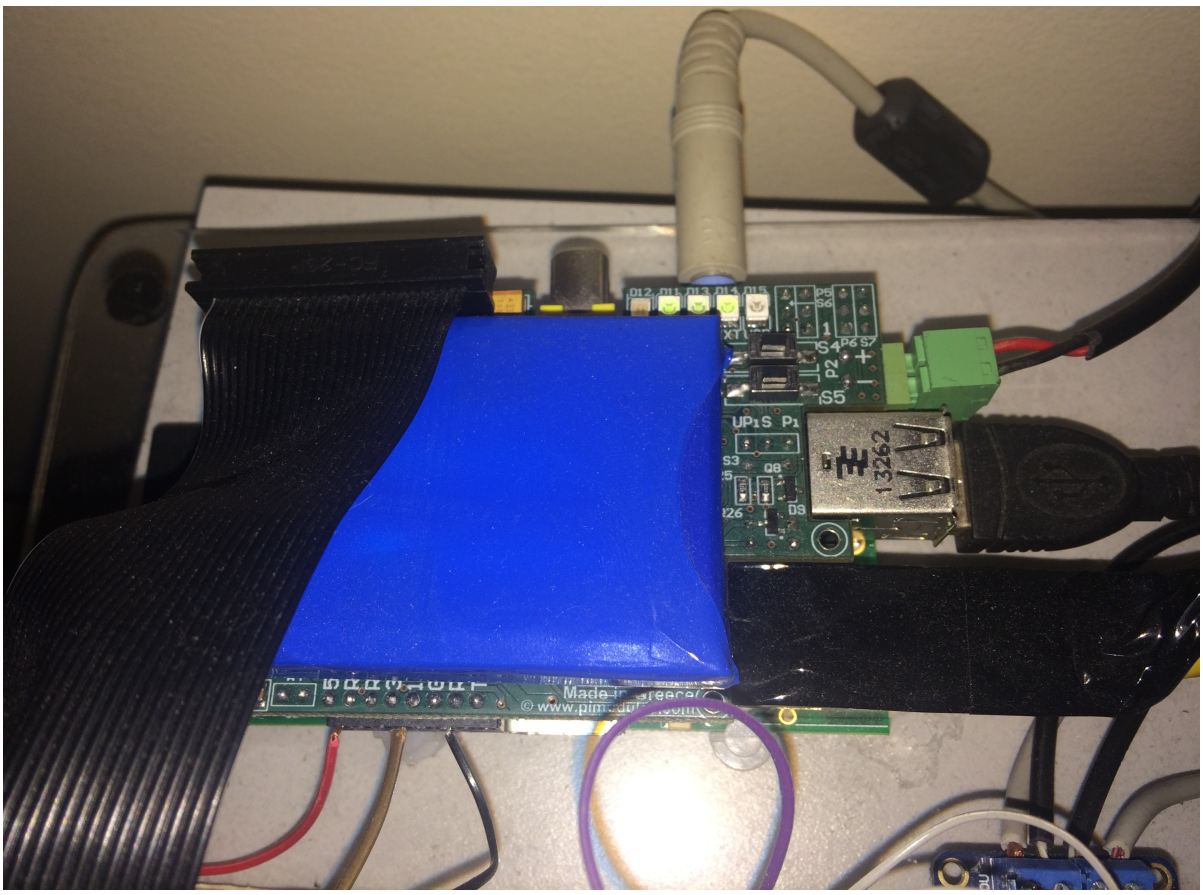


When the system is back from a power

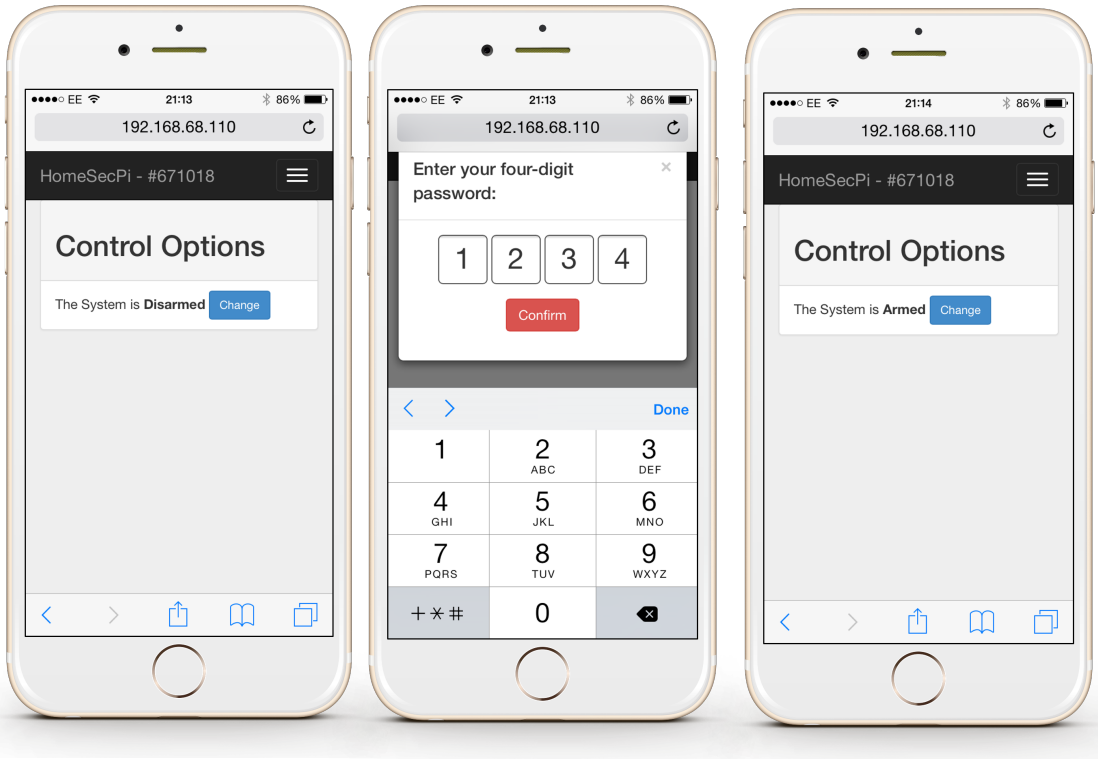


Indicates to the user to leave the building before the system is set to armed.

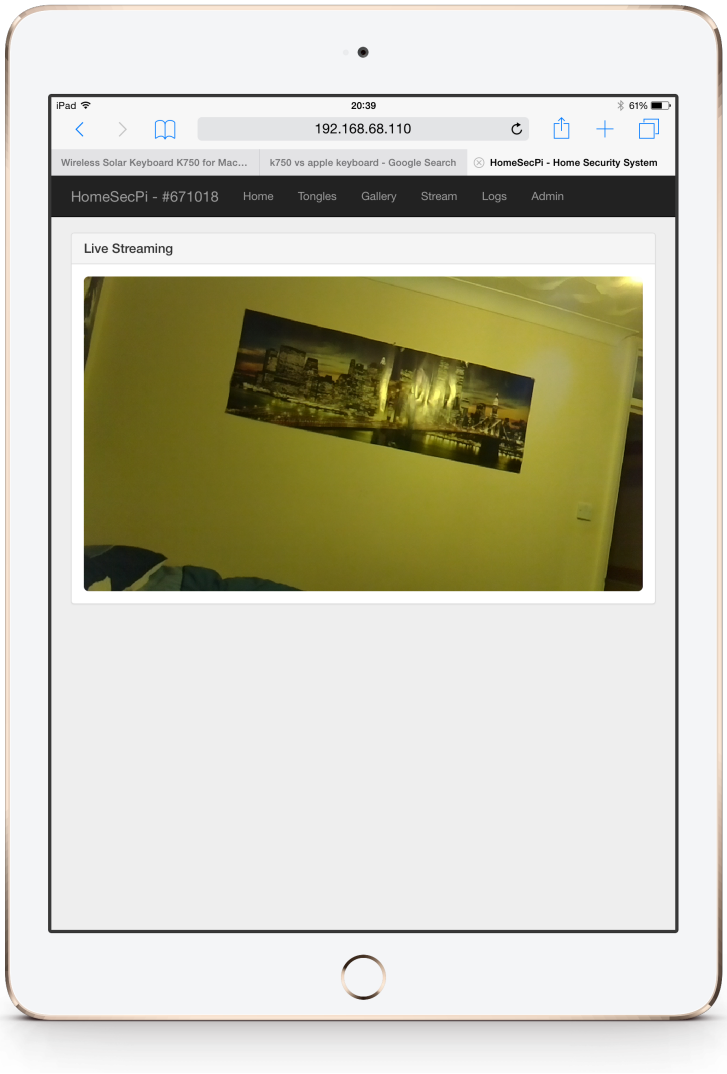
H.4 - This is the external UPS/battery connected on the Raspberry Pi. The blue back is the actual battery sitting on the UPS board. The black cable with the green plug is the 12V DC Power supply the feeds the UPS with power.



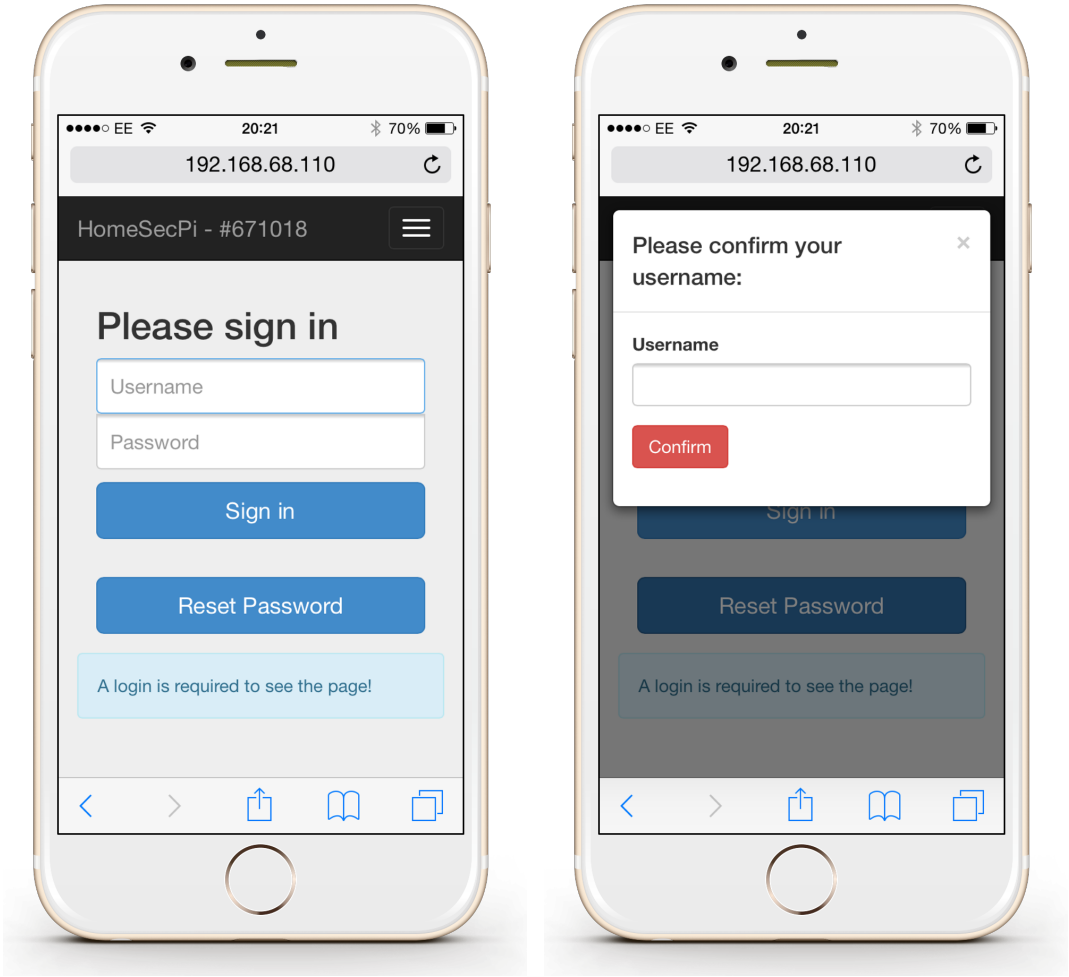
H.5 - Setting the system to “Armed” using the remote control panel.



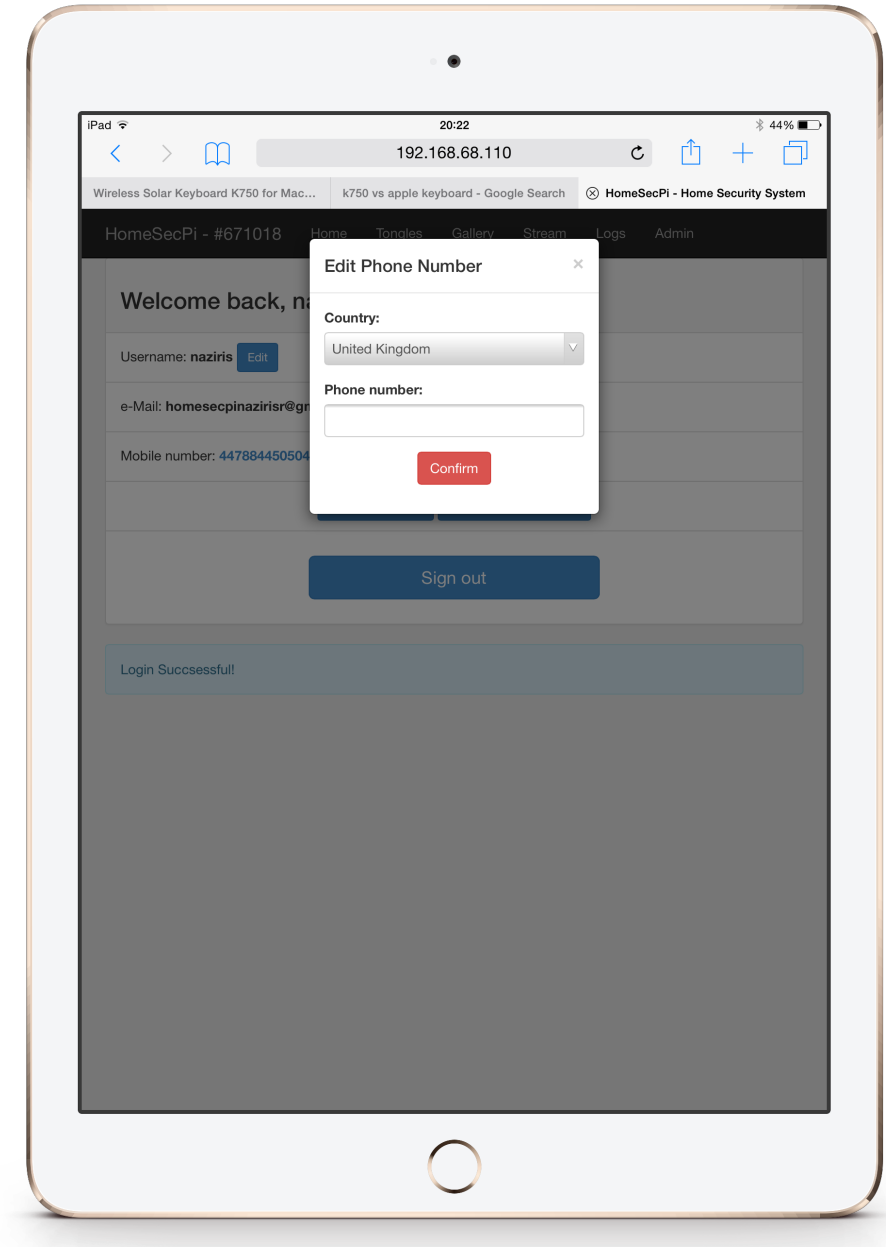
H.6 - The live streaming video page on a tablet device.

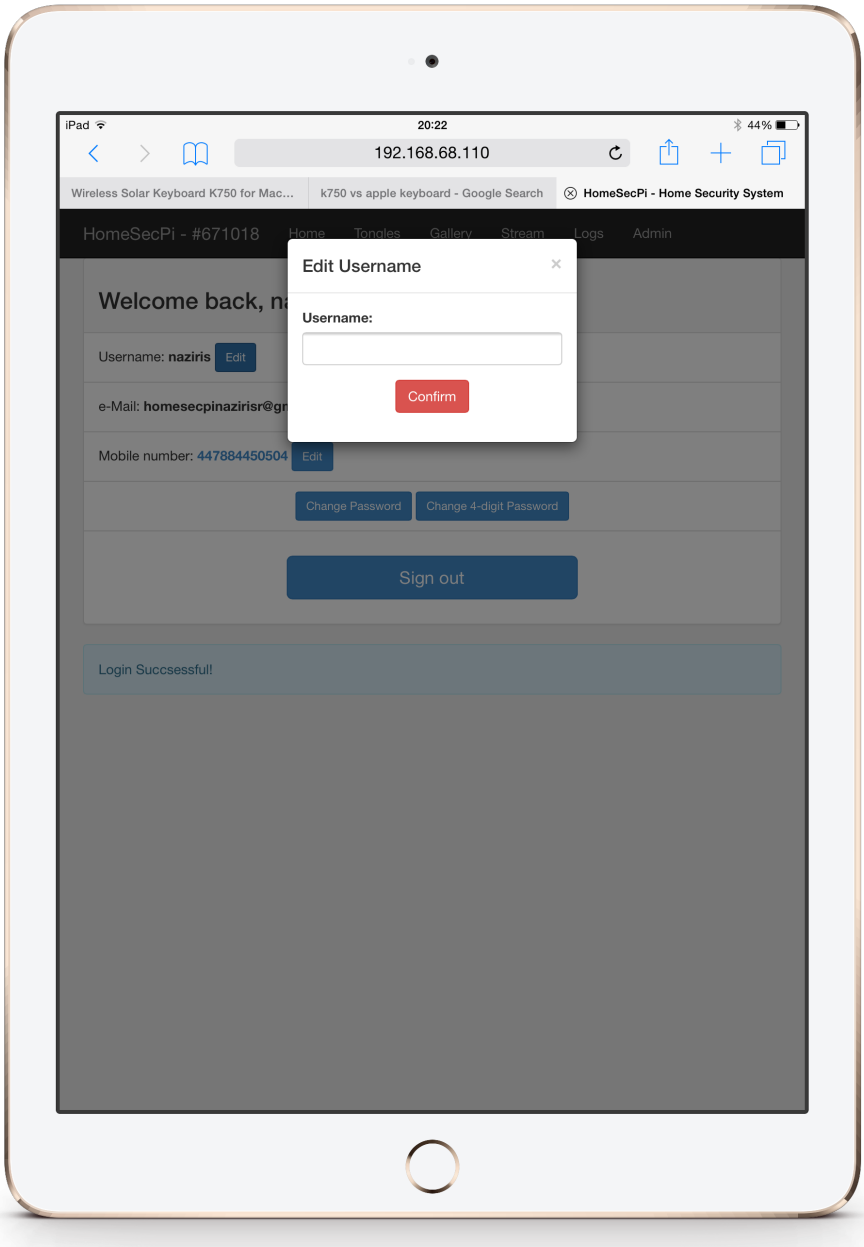


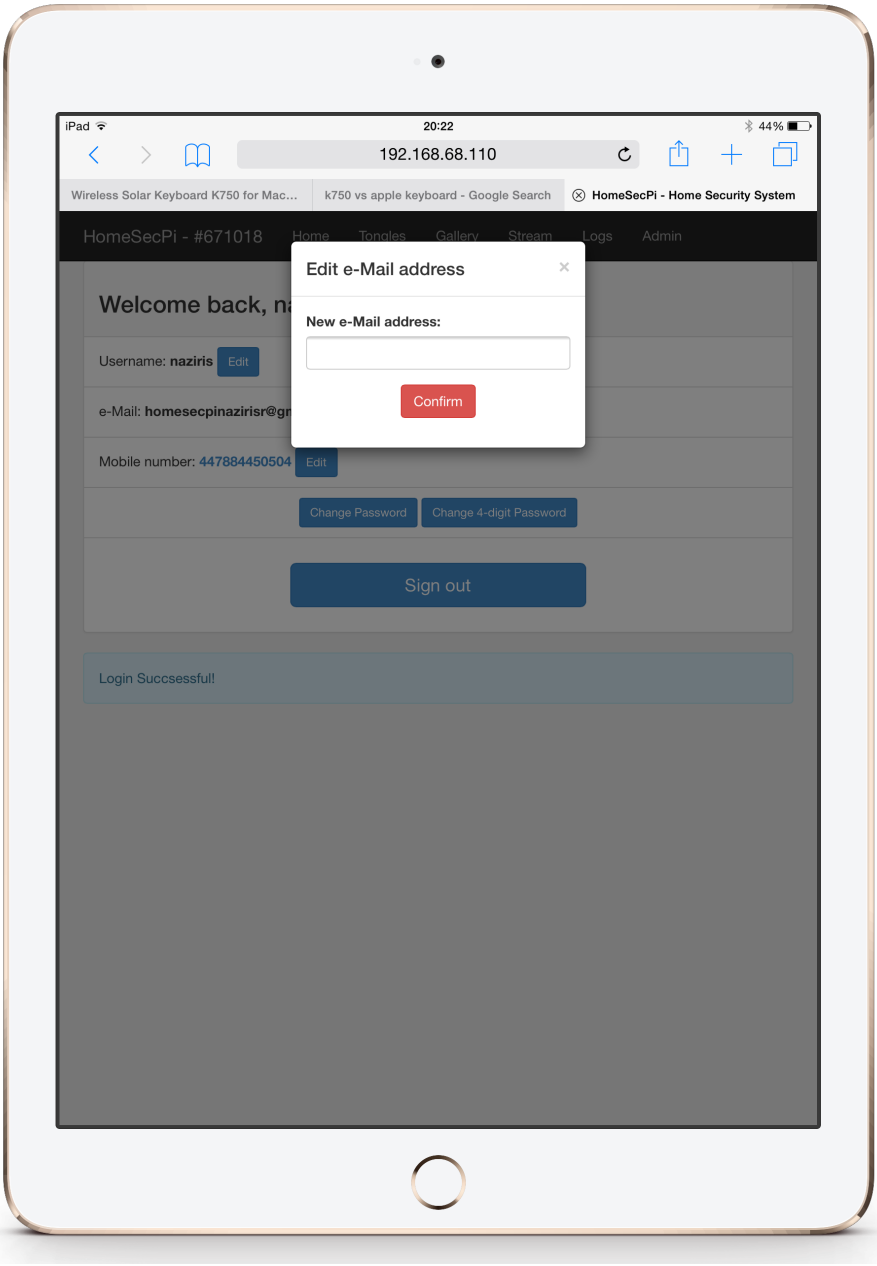
H.7 - The login page which is the only accessible page, if the user is not logged in. From this page the user have the options to log in using the account details or reset the password, which the reseted password will be emailed.

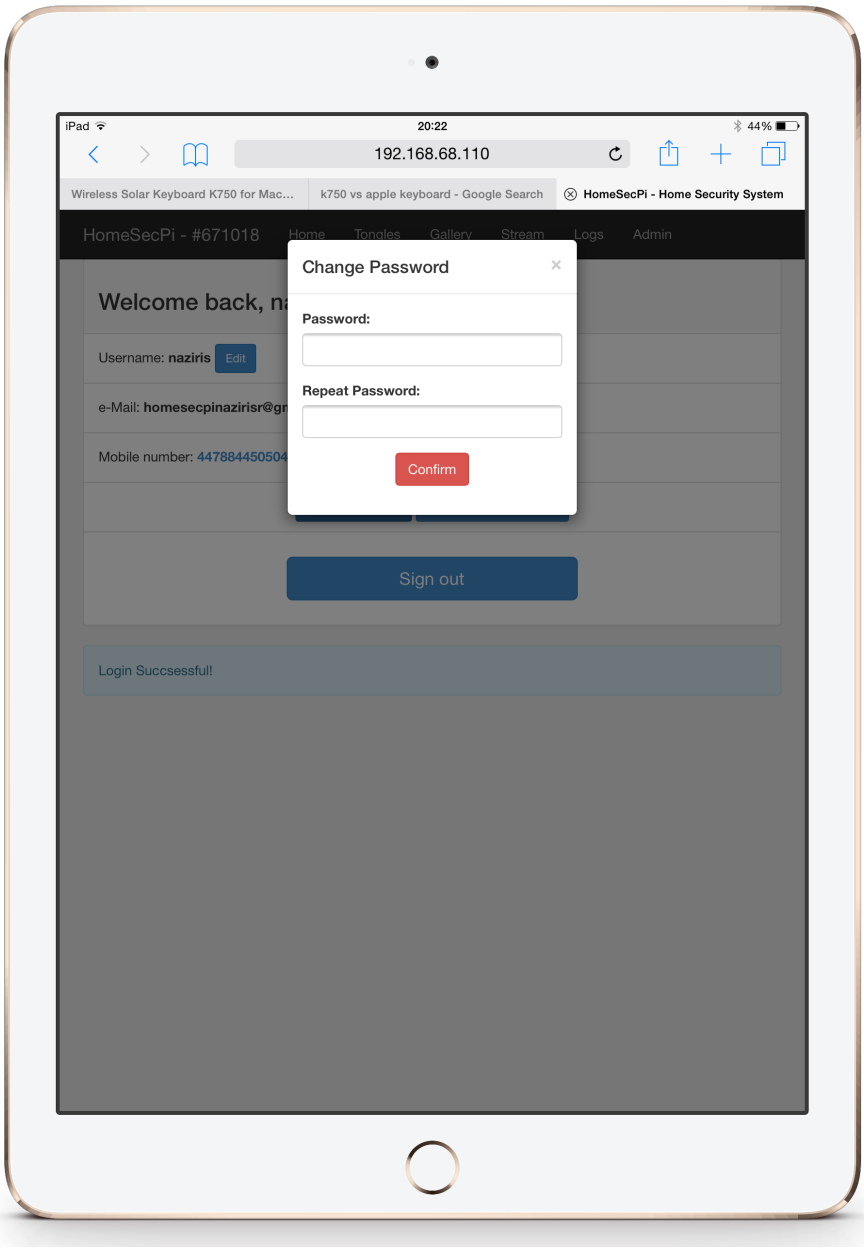


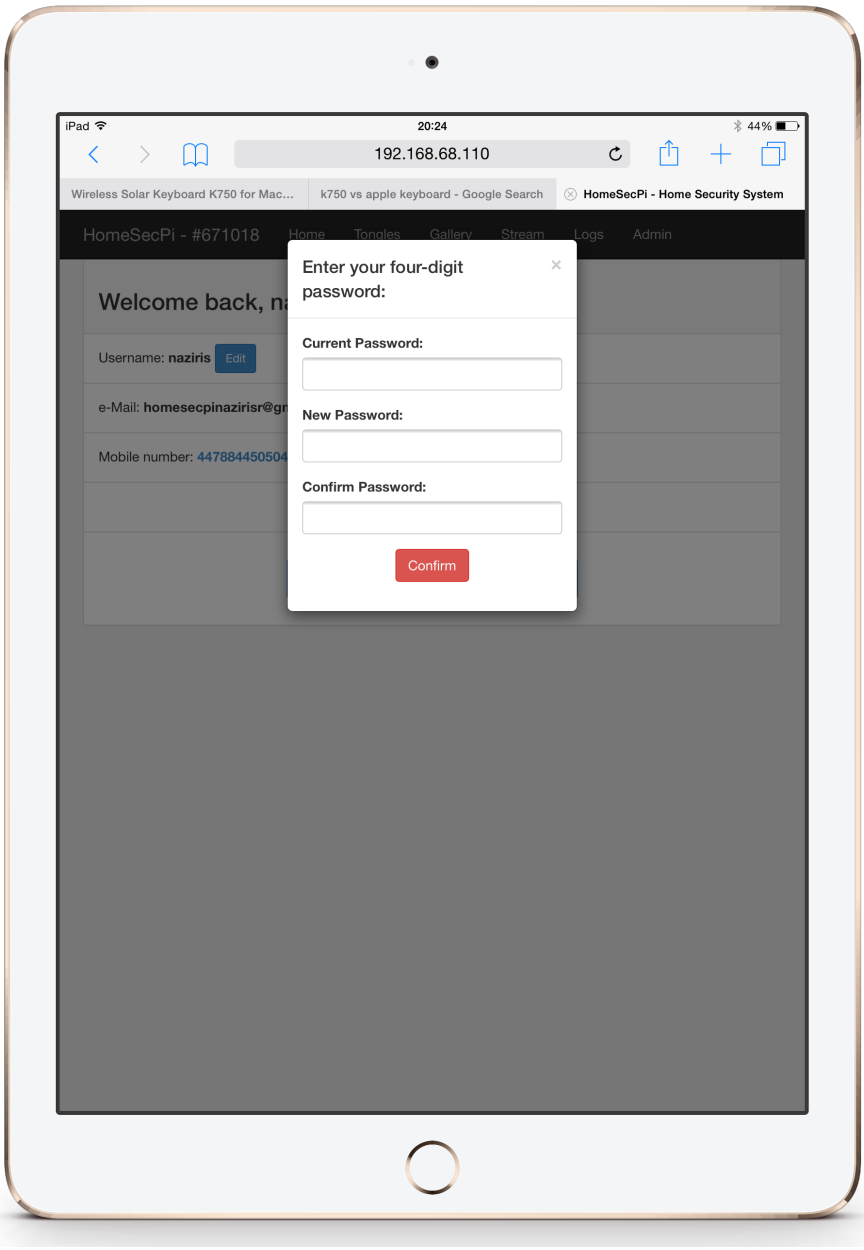
H.8 - The following screenshots indicates the process the user has to follow in order to change the user's details.




iPad
20:22
44%
192.168.68.110
Wireless Solar Keyboard K750 for Mac...
k750 vs apple keyboard - Google Search
HomeSecPi - Home Security System
HomeSecPi - #671018
Home
Tongles
Gallery
Stream
Logs
Admin
Edit Username
Username:
Confirm
Welcome back, na
Username: naziris
Edit
e-Mail: homesecpinazirisr@gr
Mobile number: 447884450504
Edit
Change Password
Change 4-digit Password
Sign out
Login Succsessful!

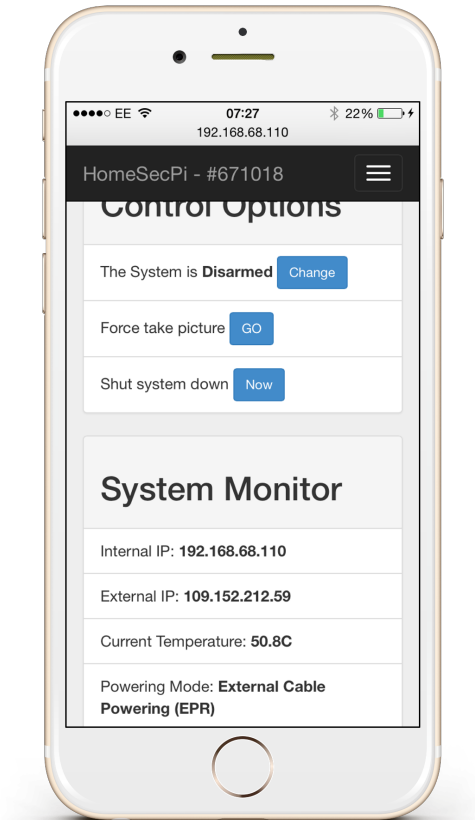

iPad
20:22
44%
192.168.68.110
Wireless Solar Keyboard K750 for Mac...
k750 vs apple keyboard - Google Search
HomeSecPi - Home Security System
HomeSecPi - #671018
Home
Toggles
Gallery
Stream
Logs
Admin
Edit e-Mail address
New e-Mail address:
Confirm
Welcome back, na
Username: naziris
Edit
e-Mail: homesecpinazirisr@gr
Mobile number: 447884450504
Edit
Change Password
Change 4-digit Password
Sign out
Login Succsessful!


iPad
20:22
44%
192.168.68.110
Wireless Solar Keyboard K750 for Mac...
k750 vs apple keyboard - Google Search
HomeSecPi - Home Security System
HomeSecPi - #671018
Home
Tongles
Gallery
Stream
Logs
Admin
Welcome back, na
Username: naziris
Edit
e-Mail: homesecpinazirisr@gr
Mobile number: 447884450504
Change Password
Password:
Repeat Password:
Confirm
Sign out
Login Succsessful!

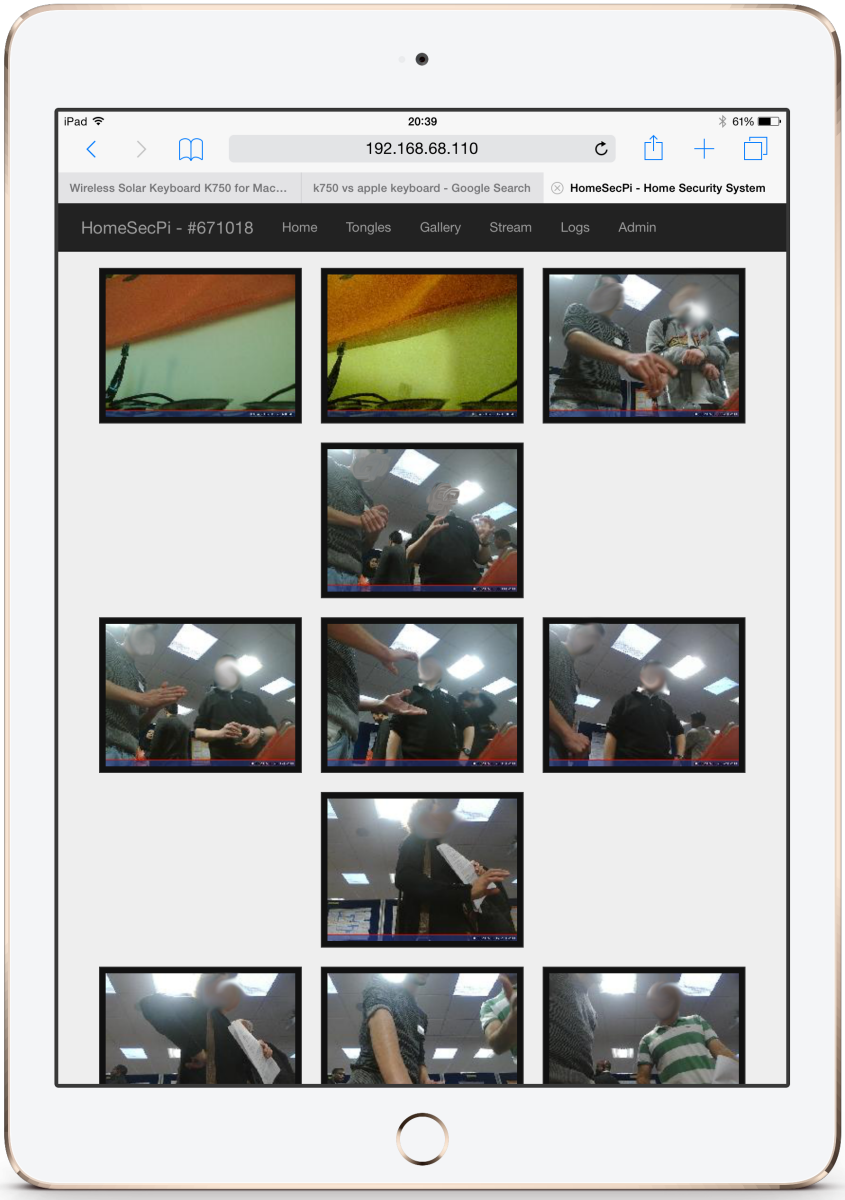

iPad
20:24
44%
192.168.68.110
Wireless Solar Keyboard K750 for Mac...
k750 vs apple keyboard - Google Search
HomeSecPi - Home Security System
HomeSecPi - #671018
Home
Toggles
Gallery
Stream
Logs
Admin
Welcome back, na
Username: naziris
Edit
e-Mail: homesecpinazirisr@gr
Mobile number: 447884450504
Enter your four-digit password:
Current Password:
New Password:
Confirm Password:
Confirm

H.9 - The screenshot indicates the result when the user click the force take picture button.

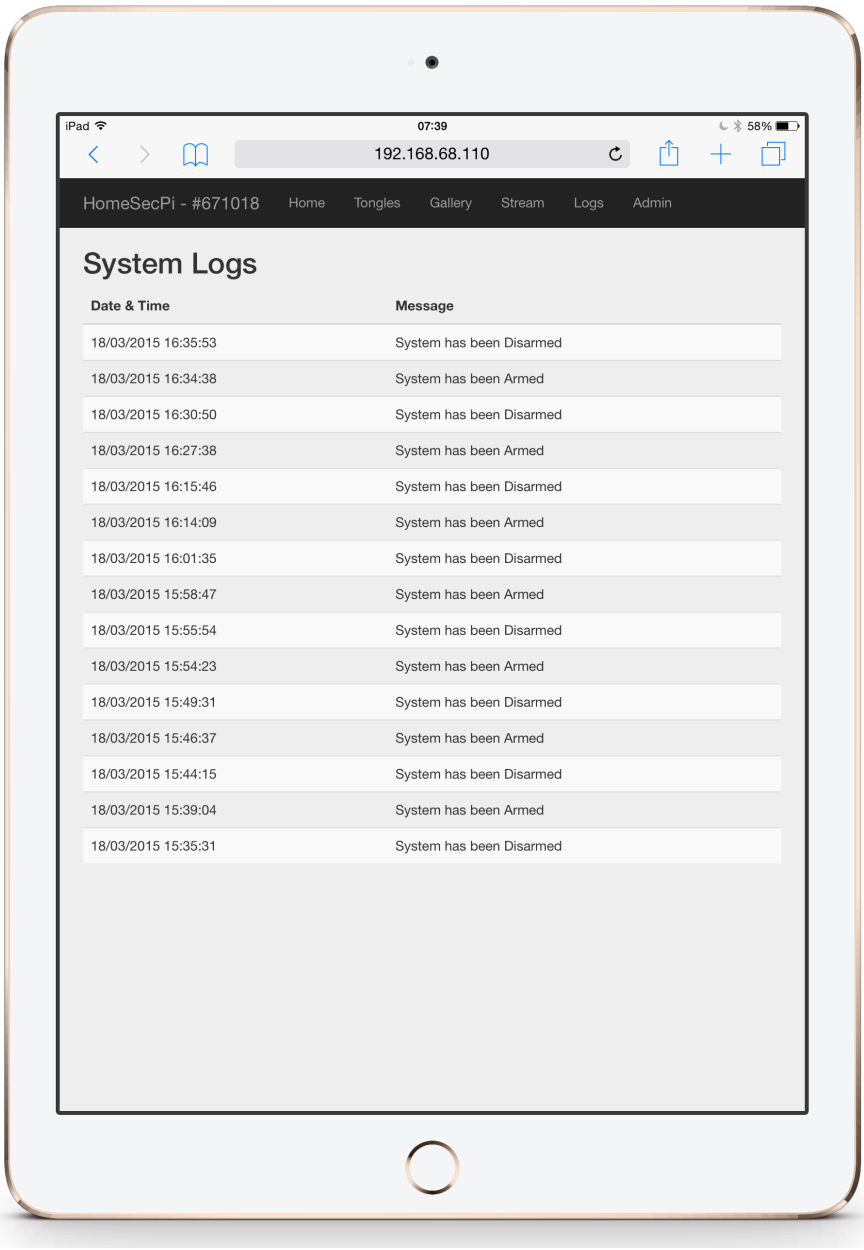
H.10 - The screenshot indicates the toggles page which includes the arm/disarm, force take picture and shut down button. When the shutdown button is pressed the same pop up window appears as is indicated in H.5.



H.11 - The gallery page with the grid of pictures, fetched from the external FTP Server.



H.12 - The logs page with the list of most recent system's incidents.



Appendix I - Price List

This cost list shows only the components that have been used in order to built the system. Also use a few other components and tools have been used, that happened to have at home or provided by the client, which will not be considered in the price list below.

Item Description	Quantity	Unit Price	Cost
Raspberry Pi -Model B & 8GB Memory Card	1	£ 32.99	£ 32.99
Adafruit Matrix Keypad	1	£ 2.99	£ 2.99
Adafruit Blue and White 16x2 LCD	1	£ 19.99	£ 19.99
RPi Premium Jumber Wires 40kp	2	£ 4.00	£ 8.00
Ultra Bright LED - Blue (10 Pack)	1	£ 0.99	£ 0.99
RPi Proto - Soldering Station 48W	1	£ 14.99	£ 14.99
RPi Proto - Solder Wire 0.7mm 100g (Lead-Free)	1	£ 5.99	£ 5.99
RasClock - Raspberry Pi Real Time Clock Module V3.0	1	£ 8.99	£ 8.99
MyPi - Protect Your Pi - 32 I/O Point Expansion	1	£ 9.99	£ 9.99
PIR Infrared Motion Sensor (HC-SR501)	1	£ 2.99	£ 2.99
Adafruit Stereo 3.7 W Class D Audio Amplifier (MAX 98306)	1	£ 6.99	£ 6.99
Breadboard (Full-Size) - White	1	£ 5.99	£ 5.99
UPiS Advanced - Uninterruptible Power Intelligent Supply	1	£ 40.83	£ 40.83
PiFlex Camera Mount - Black	1	£ 2.49	£ 2.49
Raspberry Pi -Camera Board (5MP, 1080p, v1.3	1	£ 16.66	£ 16.66
Speaker - 3” Diameter - 4Ohm 3 Watt	2	£ 1.99	£ 3.98
Shipping			£ 14.12
20% VAT			£ 39.79
Total Cost			**£ 238.76**

Appendix J - Testing User Manual

J.1 - Arm the system using the keypad

Type the four digit password on embedded keypad, while the system is disarmed in order to set it into arm mode. The system's default passcode is "1234". The system will count 60 seconds backwards allowing some time to evacuate the storage.

J.2 - Disarm the system using the keypad

Type the four digit password on embedded keypad, while the system is armed in order to set it into disarm mode. The system's default passcode is "1234".

J.3 - Plug/Unplug the main power source

To test the backup battery and the UPS indications, remove the system's plug for the wall socket while the system is running. Once the warning indications appear wait 30 seconds and insert the plug into the wall socket again.

J.4 - Alarm the system and walk into the room

Following the guidelines given on J.1, arm the system and walk away. Wait a couple of minutes and walk into the storage. In a few seconds the system will detect the movement ask for the four digit passcode. If the passcode is not given, soon notifications will be received.

J.5 - Sign in into the web based control panel

When the system is starting up, on the display the IP that the system is using it appears for 30 seconds. This IP will be used to logged in into the system, and in front of that IP the port should be included, for example IP:8068. In this case is 192.168.68.110:8068.

Type the IP:8068 and navigate through the page. The system should soon ask for username and password. The default username and password is "naziris" and "password" respectively. Click the sign in button and you will be redirected to the Admin Page.

J.6 - Arm the system using the control panel

Following the guidelines given on J.5, log into the system. Once you are logged in, navigate to the menu bar on the top of the page and select the Toggles page. Once the page is loaded click on the button next to the label "The system is Disarmed". A pop up window will appear asking for the four digit passcode. Type it and then click submit. In a while the window will disappear and the indication will now show "The system is Armed"

J.7 - Disarm the system using the control panel

Following the guidelines given on J.5, log into the system. Once you are logged in, navigate to the menu bar on the top of the page and select the Toggles page. Once the page is loaded click on the button next to the label "The system is Armed". A pop up window will appear asking for the four digit passcode. Type it and then click submit. In a

while the window will disappear and the indication will now show “The system is Disarmed”

J.8 - Force take picture using the control panel

Following the guidelines given on J.5, log into the system. Once you are logged in, navigate to the menu bar on the top of the page and select the Toggles page. Once the page is loaded click on the button next to the label “Force Take Picture”. A pop up window will appear with the captured picture.

J.9 - Shut down the system using the control panel

Following the guidelines given on J.5, log into the system. Once you are logged in, navigate to the menu bar on the top of the page and select the Toggles page. Once the page is loaded click on the button next to the label “Shut System Down”. A pop up window will appear asking for the four digit passcode. Type it and then click submit. In a while the control panel will no logger respond and the system will be shut. A manual reset will be necessary to restart the system (Press the reset button located on the top of the UPS board).

J.10 - Visit the gallery page

Following the guidelines given on J.5, log into the system. Once you are logged in, navigate to the menu bar on the top of the page and select the Gallery page. A grid list of pictures will appear on the screen.

J.11 - Visit the logs page

Following the guidelines given on J.5, log into the system. Once you are logged in, navigate to the menu bar on the top of the page and select the Logs page. A list of incidents will appear on the screen.

J.12 - Visit the streaming page

Following the guidelines given on J.5, log into the system. Once you are logged in, navigate to the menu bar on the top of the page and select the Stream page. A live streaming video will appear on the screen.

J.13 - Modify all the details of the account

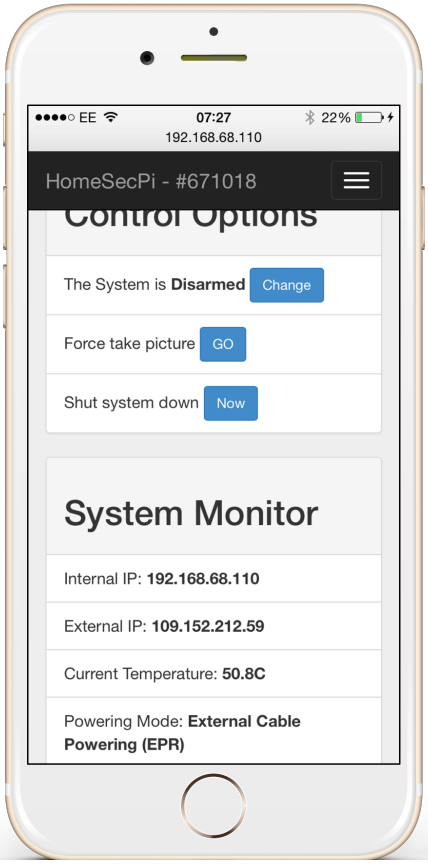
Following the guidelines given on J.5, log into the system. Once you are logged in, navigate to the menu bar on the top of the page and select the Admin page. Select the preferred detail you would like to modify and type the required details on the the appeared pop up window. Click submit and soon you will be force logged out. Log in again, and the change should take place now.

J.14 - Sign out from the control panel

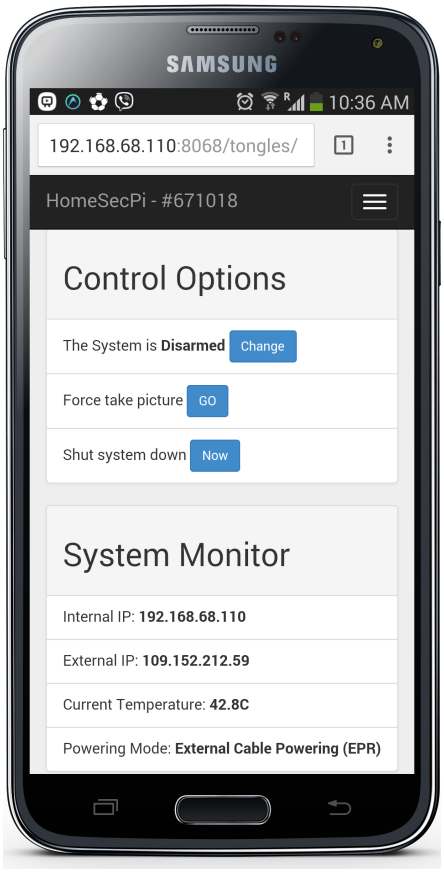
Following the guidelines given on J.5, log into the system. Once you are logged in, navigate to the menu bar on the top of the page and select the Admin page. Click the sign out button on the bottom of the page and soon you will be logged out.

Appendix K - Testing Control Panel on Various Platforms

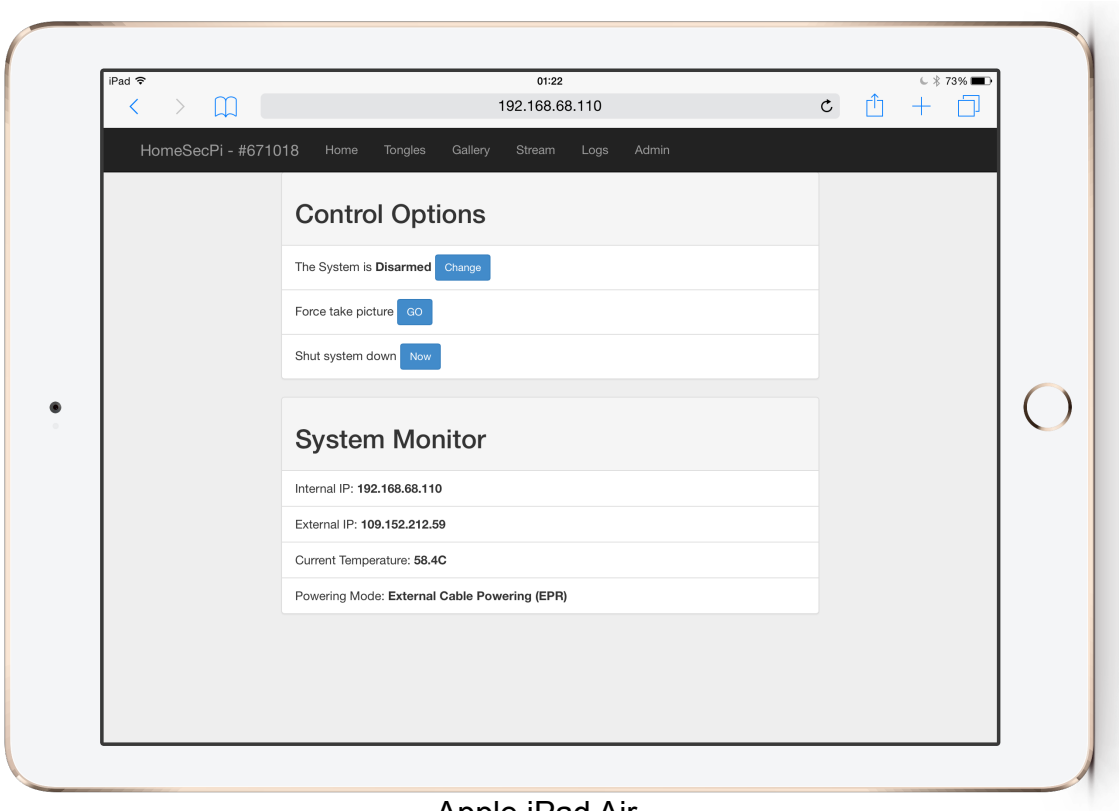
Mobile Devices



Apple iPhone 6

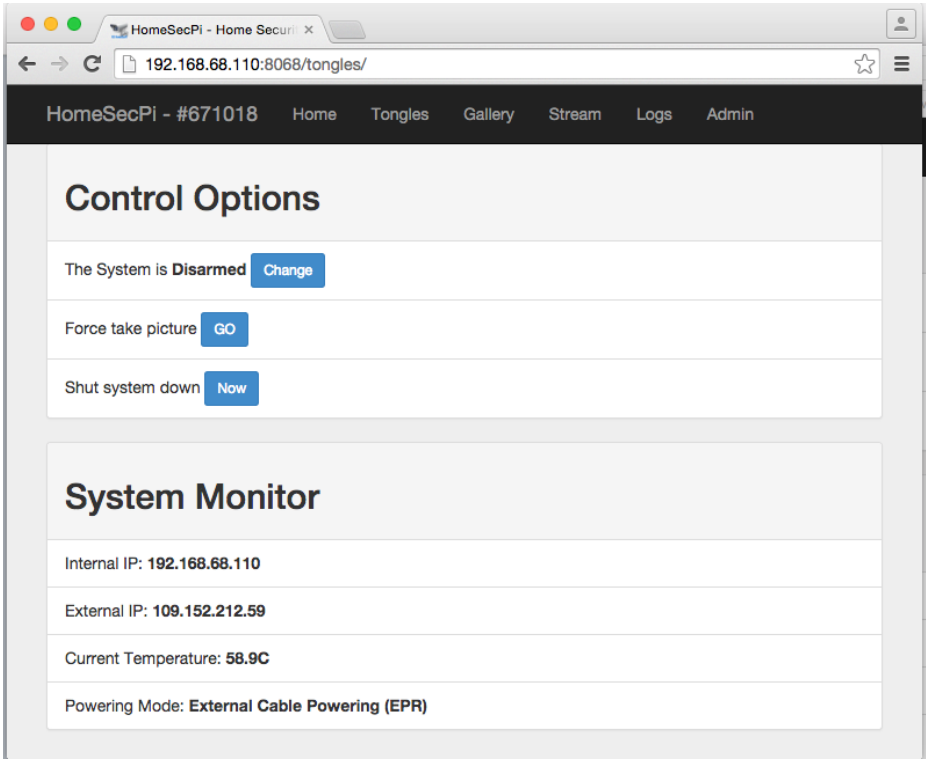


Samsung Galaxy S5

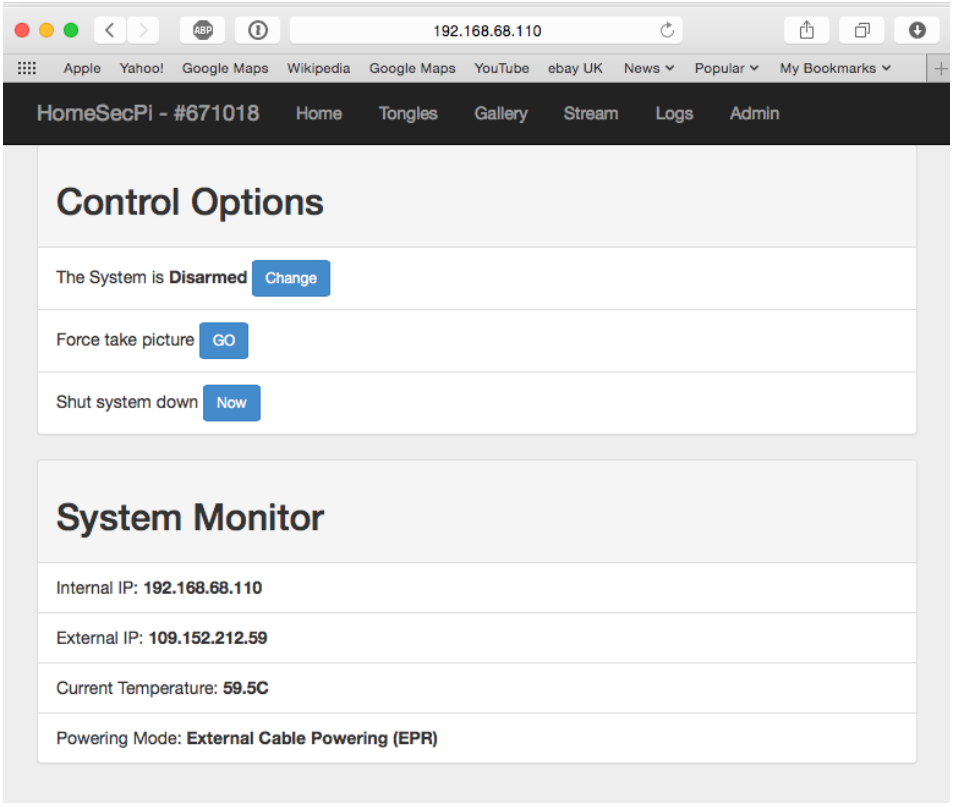


Apple iPad Air

Desktop Browsers



Google Chrome for Mac



Apple Safari for Mac