



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Υπηρεσίες ανάλυσης δεδομένων σε περιβάλλοντα αυτοματισμών και έξυπνου σπιτιού.

*Πτυχιακή εργασία
Δερβίσης Βασίλης*



Αθήνα, 2020

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τριμελής Εξεταστική Επιτροπή

Επιβλέπων

Καραγιώργου Σοφία

Επισκέπτης Καθηγήτρια, Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Μέλη

Ηρακλής Βαρλάμης

Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Γεώργιος Κουσιουρής

Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Ο Δερβίσης Βασίλης δηλώνω υπεύθυνα ότι:

1. Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.
2. Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

Αφιέρωση

Ευχαριστίες

Η παρούσα πτυχιακή εργασία υλοποιήθηκε στα πλαίσια του προγράμματος προπτυχιακών σπουδών του τμήματος Πληροφορικής και Τηλεματικής στο Χαροκόπειο Πανεπιστήμιο Αθηνών. Η υλοποίηση της πτυχιακής εργασίας ξεκίνησε τον Μάρτιο του 2020 και ολοκληρώθηκε τον Αύγουστο του 2020. Ολοκληρώνοντας τη πτυχιακή μου εργασία, θα ήθελα να ευχαριστήσω όλους όσους βοήθησαν σε όλα τα στάδια ανάπτυξής της.

Αρχικά, θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτρια κ. Καραγιώργου Σοφία για την άριστη συνεργασία καθώς και τις εύστοχες υποδείξεις της σε όλα τα στάδια ανάπτυξης της πτυχιακής. Θα ήθελα, επίσης, να ευχαριστήσω τα μέλη της εξεταστικής επιτροπής κ. Ηρακλή Βαρλάμη, και κ. Γεώργιο Κουσιουρή που μου έκαναν την τιμή να αξιολογήσουν την προσπάθειά μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για τη διαρκή υποστήριξη και βοήθεια κατά τη διάρκεια της εργασίας καθώς και την παροχή ερεθισμάτων και συνεργασίας καθ' όλη τη διάρκεια των σπουδών μου.

Περίληψη

Στην εποχή μας που η τεχνολογία αναπτύσσεται με ραγδαίους ρυθμούς ο άνθρωπος ψάχνει έξυπνες εφαρμογές και τεχνολογίες που θα του παρέχουν διευκολύνσεις στην καθημερινότητά του και κυρίως στην οικία του. Αρχικά δημιουργήθηκαν μεμονωμένες έξυπνες συσκευές (για παράδειγμα έξυπνοι διακόπτες με δυνατότητα απομακρυσμένου ελέγχου). Με τον καιρό δημιουργήθηκε η ανάγκη ενοποίησης συσκευών και συστημάτων κάτω από ένα ενιαίο έξυπνο σύστημα (smart-home). Η τεχνολογία έξυπνου σπιτιού παρέχει στους χρήστες ασφάλεια άνεση , ενεργειακή απόδοση και άλλες ευκολίες. Συγκεκριμένα, έχει την δυνατότητα να ελέγξει από όπου και αν βρίσκεται φωτισμό, θέρμανση, τηλεοράσεις, κλιματιστικά, κάμερες και συστήματα ασφάλειας μέσω κινητού τηλεφώνου ή ιστοσελίδας στο διαδίκτυο.

Συγκεκριμένα ο αριθμός των χρηστών έξυπνων σπιτιών παγκοσμίως σήμερα ξεπερνά 189 εκατομμύρια και προβλέπεται ο αριθμός αυτός τα επόμενα χρόνια να είναι πολλαπλάσιος (*Smart home worldwide market users*, n.d.). Η ραγδαία αυτή αύξηση οφείλεται και στο γεγονός ότι τα έξυπνα αυτά συστήματα προσφέρουν βελτιωμένη εξοικονόμηση ενέργειας.

Είναι συνεπώς φανερό η ανάγκη ύπαρξης κατάλληλων συστημάτων που να μπορούν να προσφέρουν τις προαναφερθείσες λειτουργικότητες με φιλικά προς τον χρήστη περιβάλλοντα .

Λέξεις κλειδιά: [έξυπνο σπίτι, ασφάλεια, εξοικονόμηση ενέργειας]

Abstract

Nowadays technology is developing rapidly and people are looking for smart applications and technologies that will provide facilities in their daily life and especially in their home. Initially, individual smart devices were created (for example, smart switches with remote control capabilities). Over time, the need arose to integrate devices and systems under a single smart-home system. Smart home technology provides users with security, comfort, energy efficiency and other facilities. In particular, it has the ability to control from anywhere the lighting, heating, TVs, air conditioners, cameras and security systems via mobile phone or web page applications.

In particular, the number of smart home users worldwide today exceeds 189 million and this number is expected to multiply in the coming years (*Smart home worldwide market users*, n.d.). This rapid increase is due to the fact that the smart systems offer improved energy savings.

Therefore, it is obvious that there is a need for appropriate systems that can provide the aforementioned functionalities with user-friendly environments.

Keywords: [smarthome, security, energy saving]

Συντομογραφίες

API	Application Programming Interface
USB	Universal Serial Bus
SQL	Structured Query Language
MVC	Model View Controller
AMQP	Advanced Message Queuing Protocol
MQTT	Message Queuing Telemetry Transport
JDBC	Java Database Connectivity
WIFI	Wireless Fidelity
LAN	Local Area Network
IP	Internet Protocol
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
JSON	JavaScript Object Notation
GPIO	General Purpose Input Output
UI	User Interface
JSP	Java Server Pages
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
RTSP	Real Time Streaming Protocol
FPS	Frames Per Second

Περιεχόμενα

Περίληψη	6
Abstract	7
1 Εισαγωγή	11
1.1 Αντικείμενο πτυχιακής	11
1.2 Περιγραφή κεφαλαίων	11
2 Επισκόπηση βιβλιογραφίας	12
2.1 Παρόμοια project και ήδη υπάρχουσες εφαρμογές	12
2.2 Προβλήματα προς αντιμετώπιση	12
3 Ανάλυση και σχεδίαση	13
3.1 Ανάλυση και περιγραφή αρχιτεκτονικής	13
3.1.1 Raspberry pi	13
3.1.2 Node-red	13
3.1.3 Maria Db	13
3.1.4 RabbitMQ	14
3.1.5 Spring framework	14
3.1.6 Διαγράμματα αρχιτεκτονικής	15
3.2 Περιγραφή συστημάτων	17
3.2.1 Alarm system	17
3.2.2 Smart home system	18
4 Υλοποίηση	19
4.1 Λεπτομέρειες υλοποίησης συστήματος alarm	19
4.1.1 Βασικά nodes	19
4.1.2 Nodes ενημερώσεων AMQP	20
Υπηρεσίες ανάλυσης δεδομένων σε περιβάλλοντα αυτοματισμών και έξυπνου σπιτιού. Δερβίσης Βασίλης	9

4.1.3	Nodes ενεργειών	20
4.1.4	Other nodes	20
4.1.5	Android client	21
4.2	Λεπτομέρειες υλοποίησης συστήματος smarthome	24
4.2.1	Spring controller	24
4.2.2	Spring security	24
4.2.3	Spring mqtt	25
4.2.4	Tasmota	25
4.2.5	User Interface	26
5	Επίλογος	44
5.1	Συμπεράσματα	44
5.2	Μελλοντικές Επεκτάσεις	44
	Αναφορές	45

1 Εισαγωγή

1.1 Αντικείμενο πτυχιακής

Σκοπός αυτής της πτυχιακής είναι να υλοποιηθεί ένα ενιαίο έξυπνο σύστημα διαχείρισης συσκευών καθώς και αισθητήρων που χρησιμοποιούνται σε καθημερινή βάση σε ένα σπίτι. Κύριος στόχος είναι η υποστήριξη μεγάλου εύρους συσκευών και ήδη υπάρχοντων συστημάτων, η εξοικονόμηση ενέργειας με την εμφάνιση προτάσεων και σημαντικών πληροφοριών καθώς και η διευκόλυνση του χρήστη στην καθημερινότητά του μέσω ενός εύκολου και φιλικού περιβάλλοντος.

1.2 Περιγραφή κεφαλαίων

Η παρούσα πτυχιακή εργασία είναι δομημένη με βάση τα εξής κεφάλαια:

- Στο 1ο κεφάλαιο γίνεται μια εισαγωγή στο αντικείμενο της πτυχιακής.
- Στο 2ο κεφάλαιο αναφέρονται παρόμοιες εφαρμογές και τα τεχνικά προβλήματα που προέκυψαν.
- Στο 3ο κεφάλαιο περιγράφονται βασικές έννοιες της αρχιτεκτονικής που αναπτύχθηκε και υλοποιήθηκε. Επίσης γίνεται περιγραφή των δύο (2) σχετικών συστημάτων.
- Στο 4ο κεφάλαιο αναφέρονται οι λεπτομέρειες υλοποίησης των δύο συστημάτων. Σημειώνεται ότι στα πλαίσια της παρούσας πτυχιακής εργασίας θα γίνει επίδειξη Live Demo που να αναδεικνύει τη διαλειτουργικότητα των συσκευών και υπηρεσιών που αναπτύχθηκαν.
- Στο 5ο κεφάλαιο αναφέρονται τα γενικά συμπεράσματα της πτυχιακής καθώς και μελλοντικές επεκτάσεις αυτής.

2 Επισκόπηση βιβλιογραφίας

2.1 Παρόμοια project και ήδη υπάρχουσες εφαρμογές

Για την υλοποίηση της πτυχιακής εξετάστηκαν αρκετές υπάρχουσες εφαρμογές εκ των οποίων οι σημαντικότερες είναι:

Knx Visualisation Software : Παρέχει στον χρήστη σύνθετο γραφικό περιβάλλον των κτιρίων για τα knx συστήματα. Ακόμα παρέχει χάρτη με το σχέδιο του κτιρίου και τις έξυπνες συσκευές

Google home : Παρέχει στον χρήστη ηχητικές εντολές για να ελέγξει περιορισμένο πλήθος συσκευών.

Vivint Smart Home : Παρέχει στον χρήστη εκτός από την διαχείριση συσκευών και την ασφάλεια του κτηρίου με χρήση αισθητήρων με μια εφαρμογή. Ακόμα συνεργάζεται με Google-home και Amazon-echo. Ο χειρισμός του συστήματος γίνεται μόνο από εφαρμογή κινητού.

2.2 Προβλήματα προς αντιμετώπιση

Ένα από τα κύρια προβλήματα που παρουσιάζονται στα προαναφερθέντα project, είναι η έλλειψη συμβατότητας με αρκετές συσκευές και η δυσκολία στον τομέα εγκατάστασης των συστημάτων αυτών. Συγκεκριμένα, πολλές έξυπνες συσκευές καθώς και αρκετά συστήματα ασφαλείας δεν διαθέτουν κάποιο API που μπορεί να χρησιμοποιηθεί από εφαρμογές τρίτων, με αποτέλεσμα ο χρήστης να έχει μια εφαρμογή για κάθε ξεχωριστό σύστημα ή διαφορετική εταιρεία. Ακόμα σημαντικός παράγοντας είναι η ευελιξία του συστήματος ώστε να μπορεί να εγκατασταθεί σε όλους τους χώρους ανεξαρτήτως υπάρχουσας καλωδίωσης. Τέλος θα πρέπει να ληφθεί υπόψη και το πρόβλημα του κόστους εγκατάστασης συστημάτων έξυπνου σπιτιού ώστε ο χρήστης να έχει την δυνατότητα με χαμηλό κόστος να το αναβαθμίζει όσο επιθυμεί και να μην χρεώνεται από την αρχή για πράγματα που δεν χρησιμοποιεί.

3 Ανάλυση και σχεδίαση

3.1 Ανάλυση και περιγραφή αρχιτεκτονικής

3.1.1 Raspberry pi

Το raspberry pi είναι ένας πολύ μικρός υπολογιστής χαμηλού κόστους που μπορεί να χρησιμοποιηθεί ως server. Το πρώτο pi κατασκευάστηκε από την Raspberry Pi Foundation το 2012. Έκτοτε η εταιρεία έχει δημιουργήσει νεότερα μοντέλα με μεγαλύτερη υπολογιστική ισχύ και περισσότερες δυνατότητες. Προσφέρει συνδεσιμότητα με WIFI, Bluetooth και LAN και άλλες δυνατότητες με τις οποίες αυξάνονται τα σενάρια χρήσης του. Αυτό που το καθιστά τόσο σημαντικό είναι το λειτουργικό του. Το raspbian είναι βασισμένο σε Debian linux. Ακόμα μπορεί να δεχτεί και άλλα λειτουργικά συστήματα όπως Windows και Android. Ο αποθηκευτικός χώρος σε αυτήν την περίπτωση δεν είναι δίσκος αλλά μια microSD card με θετικό αποτέλεσμα την φορητότητα του συστήματος. Τέλος με τα GPIO pins του δίνεται η δυνατότητα σύνδεσης με οποιοδήποτε αισθητήρα με ευκολία ([Raspberry pi official page](#), n.d.).

3.1.2 Node-red

Το Node-RED είναι ένα εργαλείο ανάπτυξης που βασίζεται στη ροή για τον οπτικό προγραμματισμό. Η οπτική γλώσσα προγραμματισμού επιτρέπει στους χρήστες να δημιουργούν προγράμματα κατασκευάζοντας τα στοιχεία του προγράμματος γραφικά αντί να τα ορίσουν με κείμενο. Αναπτύχθηκε αρχικά από την IBM για τη σύνδεση συσκευών υλικού, API και διαδικτυακών υπηρεσιών, ως μέρος του Διαδικτύου των πραγμάτων. Το Node-RED παρέχει έναν επεξεργαστή ροής που βασίζεται σε πρόγραμμα περιήγησης ιστού, ο οποίος μπορεί να χρησιμοποιηθεί για τη δημιουργία λειτουργιών JavaScript. Στοιχεία εφαρμογών μπορούν να αποθηκευτούν ή να κοινοποιηθούν για επαναχρησιμοποίηση. Οι ροές που δημιουργούνται στο Node-RED αποθηκεύονται σε μορφή JSON ([Node-red official page](#), n.d.) και ([Node-red wiki page](#), n.d.). Ο κύριος λόγος που χρησιμοποιήθηκε για το σύστημα συναγερμού, ήταν η εύκολη παραμετροποίηση και ανάγνωση του κώδικα σε περιπτώσεις συντήρησης και αναβάθμισης αλλά και η ύπαρξη node για την χρήση των gpio του pi.

3.1.3 Maria Db

Η SQL είναι μια βασική γλώσσα για πρόσβαση και χειρισμό βάσεων δεδομένων. Η βάση δεδομένων του SQL Server αποτελείται από μια συλλογή πινάκων που αποθηκεύουν ένα συγκεκριμένο σύνολο δομημένων δεδομένων. Ένας πίνακας περιέχει μια συλλογή γραμμών, που αναφέρονται επίσης ως εγγραφές και στήλες, καθώς και ως χαρακτηριστικά ([What is sql](#), n.d.). Ο MariaDB server είναι ένας από τους πιο διαδεδομένους servers στον κόσμο. Είναι ανοιχτού κώδικα και κατασκευάστηκε από τους προγραμματιστές της Mysql. Έχει αρκετούς σημαντικούς χρή-

στες όπως Wikipedia και Google ([MariaDb official page](#), n.d.). Η επιλογή της MariaDb για server βάσης δεδομένων έγινε καθαρά με βάση κριτήρια όπως η ταχύτητα και η επεκτασιμότητα, αλλά και στο γεγονός ότι έχει υποστήριξη παλαιότερων εκδόσεων με αποτέλεσμα να κάνει την αναβάθμιση πιο εύκολη ([Why mariaDb over mysql](#), n.d.).

3.1.4 RabbitMQ

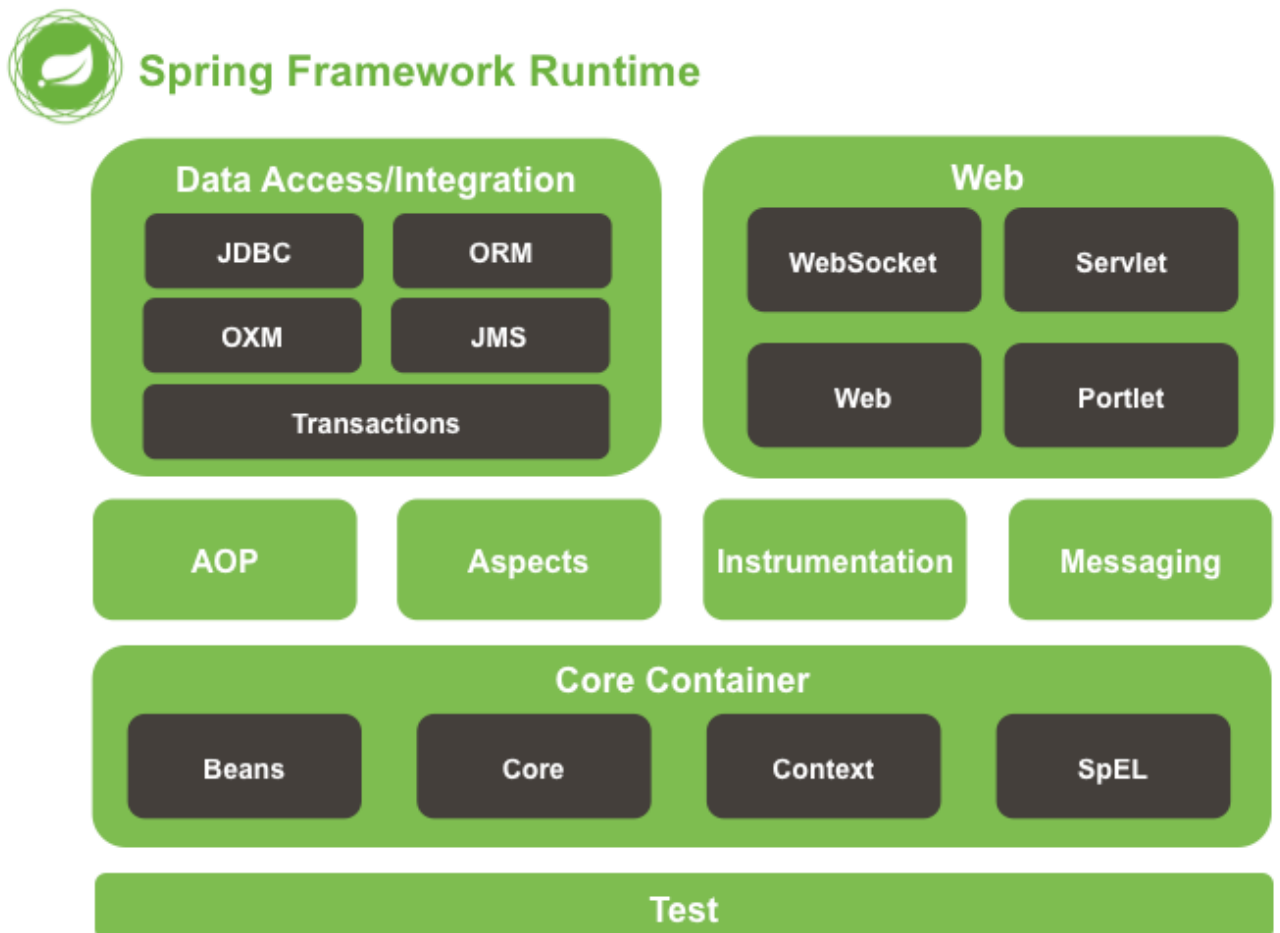
Το RabbitMQ είναι ένα λογισμικό ανοιχτού κώδικα μεσολαβητή μηνυμάτων ή message-oriented middleware που αρχικά εφάρμοσε το Advanced Message Queuing Protocol (AMQP) και από τότε επεκτάθηκε με μια αρχιτεκτονική plug-in για την υποστήριξη Streaming Text Oriented Messaging Protocol (STOMP), MQ Telemetry Transport (MQTT) και άλλα πρωτόκολλα. Πρωτοεμφανίστηκε το 2007 από την Rabbit Technologies και αργότερα το 2013 πέρασε στα χέρια της Pivotal Software. Το πρόγραμμα διακομιστή RabbitMQ είναι γραμμένο στη γλώσσα προγραμματισμού Erlang και βασίζεται στο Open Telecom Platform framework clustering and failover. Οι βιβλιοθήκες πελατών για διασύνδεση με τον μεσολαβητή είναι διαθέσιμες για όλες τις μεγάλες γλώσσες προγραμματισμού ([What is rabbitMq](#), n.d.).

Το προηγμένο πρωτόκολλο ουράς μηνυμάτων ή αλλιώς AMQP είναι ένα ανοιχτό πρότυπο πρωτόκολλο επιπέδου εφαρμογής για μεσαίο λογισμικό προσανατολισμένο σε μηνύματα. Δομικά στοιχεία του AMQP είναι η message queue ή αλλιώς ουρά μηνυμάτων, τα exchanges, το binding, το μήνυμα, την σύνδεση και το κανάλι. Η ουρά μηνυμάτων λειτουργεί σαν buffer ο οποίος αποθηκεύει τα μηνύματα μέχρι να δρομολογηθούν. Υπάρχουν τέσσερα είδη exchanges, direct exchange, fanout exchange, topic exchange, και header exchange. Binding είναι μια σχέση μεταξύ μιας ουράς και ενός exchange που αποτελείται από ένα σύνολο κανόνων που χρησιμοποιεί το exchange για τη δρομολόγηση μηνυμάτων σε ουρές. Όσον αφορά το μήνυμα είναι μια οντότητα που αποστέλλεται από τον publisher στην ουρά και γίνεται subscribed τελικά από τον consumer. Η AMQP σύνδεση είναι η σύνδεση δικτύου μεταξύ της εφαρμογής και του AMQP broker. Το κανάλι είναι μια οικονομική σύνδεση μέσα σε μια σύνδεση, μεταξύ δύο AMQP peers ([What is amqp](#), n.d.). Το RabbitMQ έχει πρόσθετο (MQTT Plugin) που υποστηρίζει το MQTT πρωτόκολλο. Το mqtt είναι ένα πρωτόκολλο δικτύου δημοσίευσης-εγγραφής που μεταφέρει μηνύματα μεταξύ συσκευών μέσω TCP / IP. Η πρώτη έκδοση του πρωτοκόλλου ήταν το 1999 ([RabbitMQ official MQTT plugin](#), n.d.).

3.1.5 Spring framework

Το Spring Framework είναι ένα ανοιχτού κώδικα πλαίσιο εφαρμογής για την πλατφόρμα της Java. Το Spring Core μπορεί να χρησιμοποιηθεί από οποιαδήποτε Java εφαρμογή. Επίσης με πρόσθετα καθίσταται δυνατή η δημιουργία web εφαρμογών. Το Spring framework αποτελείται από είκοσι modules. Το inversion of control container παρέχει μέσα για την ρύθμιση και την διαχείριση αντικειμένων Java χρησιμοποιώντας την ικανότητα reflection. Reflection είναι η ικανότητα μιας διεργασίας να ελέγχει και να μεταποιεί την δικιά της δομή και συμπεριφορά. Το Spring jdbc module παρέχει ένα επίπεδο αφαίρεσης jdbc που καθιστά μη αναγκαία την ανάπτυξη κώδικα αλλά και την προσπέλαση λαθών που αφορά την εκάστοτε βάση δεδομένων. Τέλος το Web layer αποτελείται από

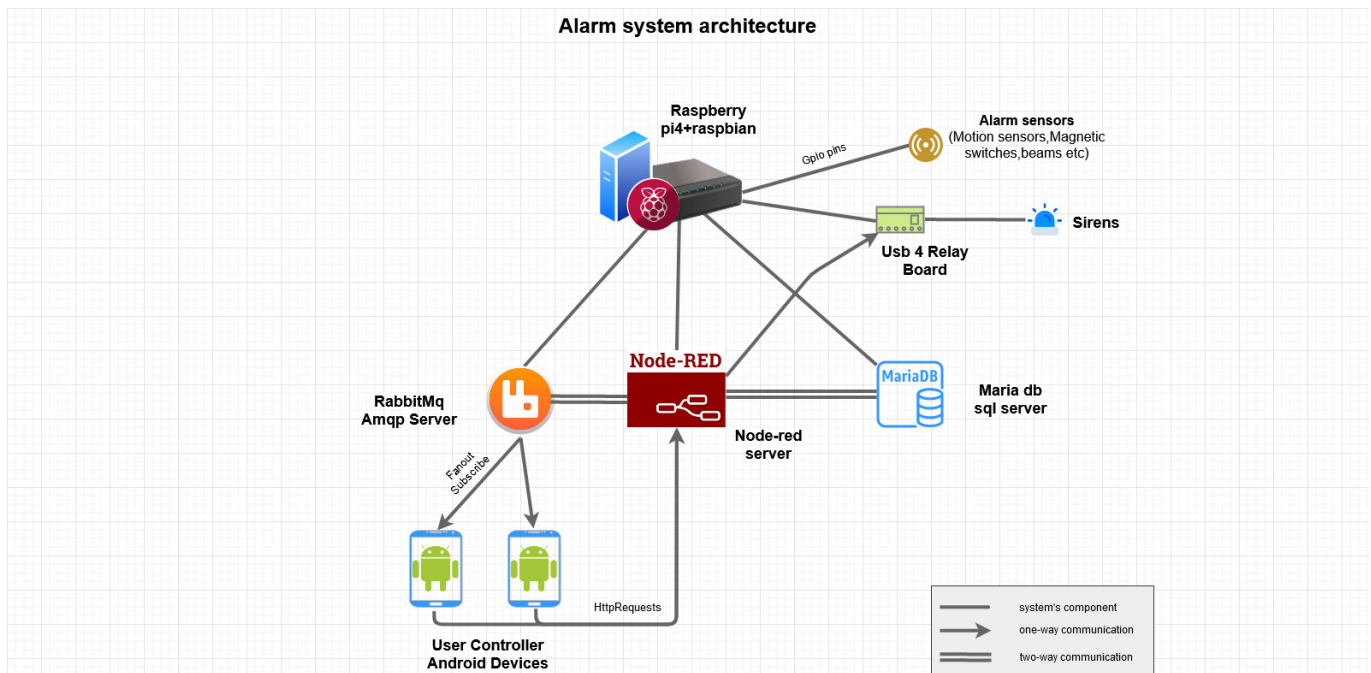
τα web και webmvc modules. Η ενότητα spring-web παρέχει βασικές δυνατότητες ενσωμάτωσης στο διαδίκτυο. Ενώ η μονάδα spring-webmvc περιέχει τον model-view-controller (MVC) και τα REST Web Services για τις εφαρμογές Web ([Spring framework official page](#), n.d.). Στο Σχήμα 1 φαίνεται η σφαιρική εικόνα του spring framework.



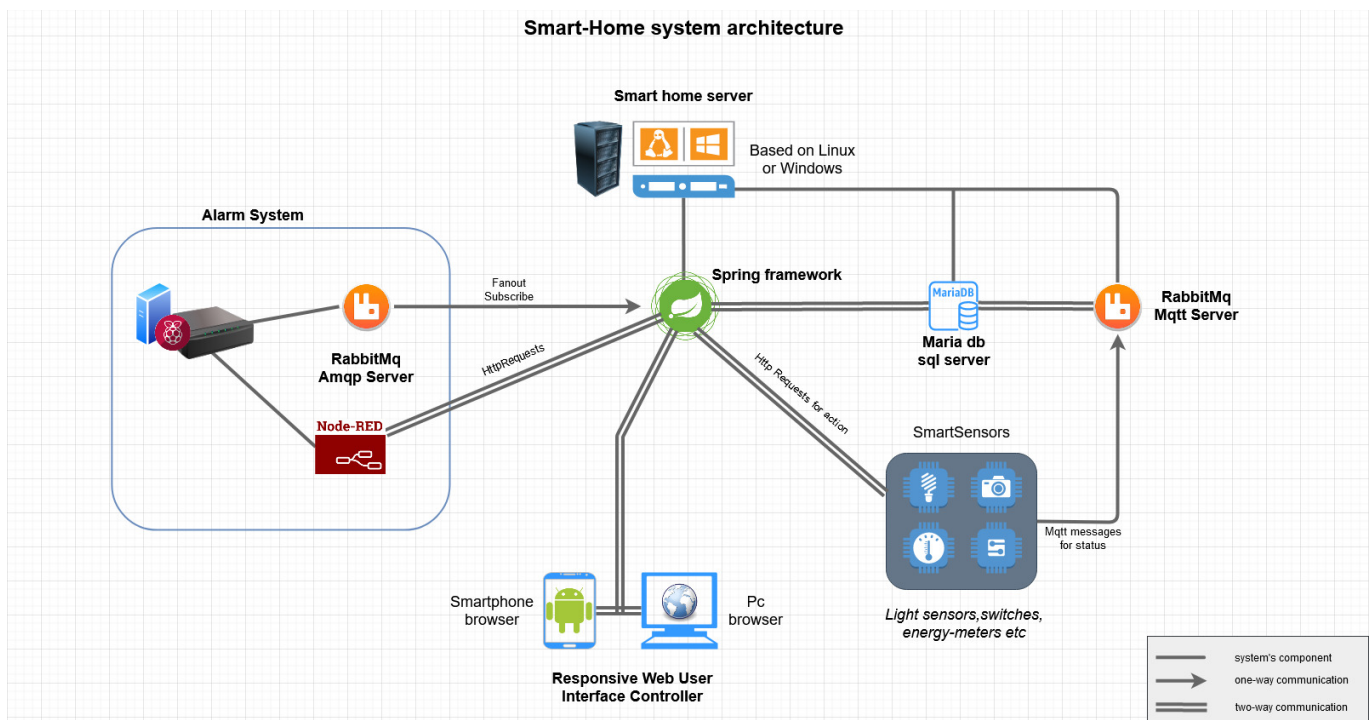
Σχήμα 1 : Spring framework

3.1.6 Διαγράμματα αρχιτεκτονικής

Παρακάτω φαίνεται η διαγραμματική απεικόνιση των δύο συστημάτων.



Σχήμα 2 : Αρχιτεκτονική συστήματος συναγερμού



Σχήμα 3 : Αρχιτεκτονική συστήματος έξυπνου σπιτιού

3.2 Περιγραφή συστημάτων

3.2.1 Alarm system

Αρχικά υλοποιήθηκε το σύστημα συναγερμού. Βασικές αρχές που οδήγησαν την συγκεκριμένη υλοποίηση είναι οι απαιτήσεις των χρηστών για ένα σύστημα που θα έχει χαμηλή κατανάλωση ώστε να μπορεί να αντέξει τουλάχιστον μια ημέρα χρήσης σε μπαταρία αλλά και για ένα σύστημα το οποίο θα δουλεύει είκοσι τέσσερις ώρες το εικοσιτετράωρο χωρίς προβλήματα. Το σύστημα λοιπόν αυτό προσφέρει βασικές δυνατότητες όπως arm, disarm καθώς και πιο σύνθετα σενάρια όπως Stay και exit. Επίσης ο χρήστης έχει την δυνατότητα να επιλέξει να κάνει bypass τις ζώνες που επιθυμεί. Σε περίπτωση συναγερμού ενεργοποιείται η εξωτερική και εσωτερική σειρήνα του κτηρίου και ταυτόχρονα ο χρήστης ειδοποιείται με κατάλληλη ένδειξη στην εφαρμογή να βάλει κωδικό για να γίνει disarm. Το σύστημα αποτελείται από δύο ρόλους χρηστών. Έναν διαχειριστή που μπορεί να προσθέτει και να διαγράφει χρήστες και έναν ή περισσότερους απλούς χρήστες. Τέλος ο χειρισμός του συστήματος γίνεται με δύο τρόπους. Είτε με την Android εφαρμογή είτε με την εφαρμογή smart home του 2ου συστήματος.

3.2.2 Smart home system

Η υλοποίηση του συστήματος έξυπνου σπιτιού έχει ως κύριο στόχο την διευκόλυνση του χρήστη σε καθημερινή βάση καθώς την ευρεία υποστήριξη πολλών αισθητήρων και έξυπνων συσκευών. Συγκεκριμένα το σύστημα χωρίζεται σε δύο ρόλους χρηστών.

Αρχικά ο διαχειριστής admin του συστήματος έχει τις εξής δυνατότητες:

- Επισκόπηση κατάστασης συστήματος
- Διαχείριση χρηστών
- Ρυθμίσεις συστήματος συναγερμού
- Διαχείριση διαδικτυακών καμερών
- Διαχείριση οικιακών κ.α. έξυπνων συσκευών και διακοπών
- Διαχείριση συσκευών μέτρησης ενέργειας και προσθήκη ενεργειακού πλάνου
- Διαχείριση χώρου και προσθήκη έξυπνων συσκευών σε αυτούς

Τέλος ο χρήστης έχει δυνατότητα :

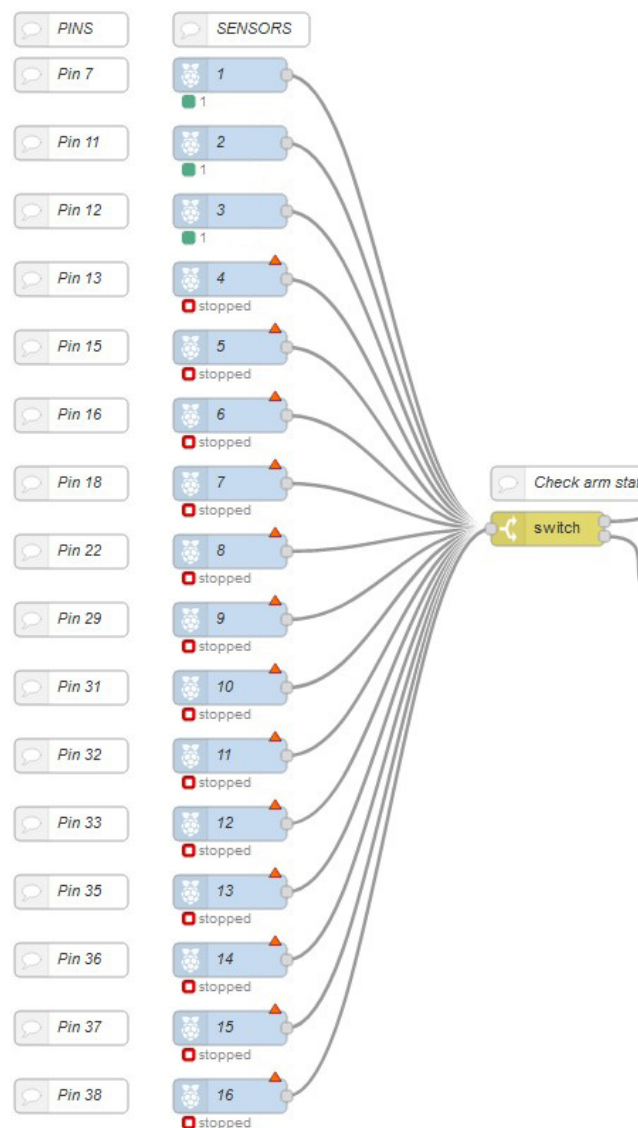
- Επισκόπησης κατάστασης συστήματος
- Διαχείρισης του λογαριασμού του
- Προβολής και χειρισμού συστήματος συναγερμού (οπλισμός, αφοπλισμός κλπ)
- Ζωντανής προβολής διαδικτυακών καμερών
- Χειρισμού έξυπνων συσκευών και προβολής κατάστασής τους
- Προβολής και διαχείρισης ενεργειακού πλάνου και διαγραμμάτων ενέργειας
- Προβολής συσκευών σε χάρτη δωματίων

4 Υλοποίηση

4.1 Λεπτομέρειες υλοποίησης συστήματος alarm

4.1.1 Βασικά nodes

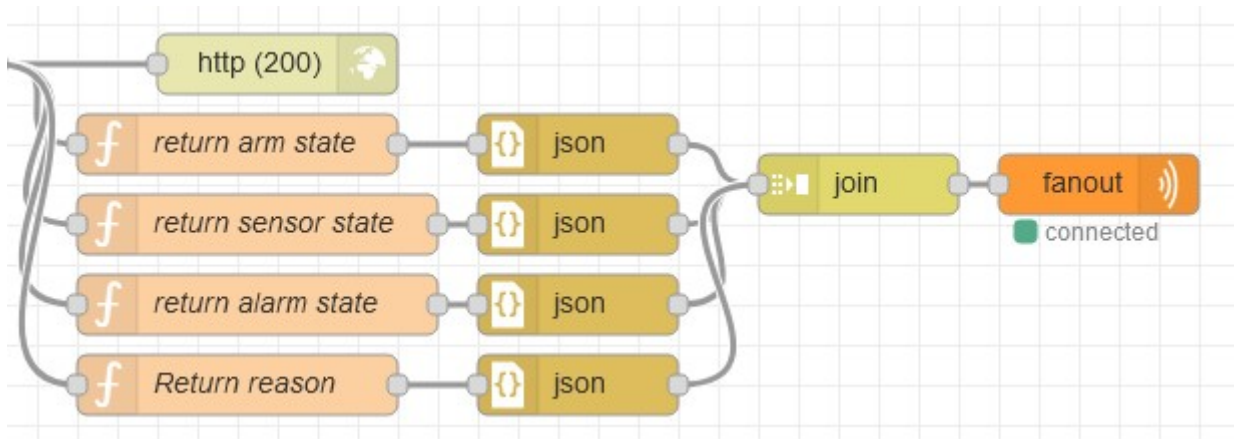
Πρώτο και βασικό μέρος του συστήματος είναι να γνωρίζουμε την κατάσταση ενός αισθητήρα δηλαδή αν είναι ανοιχτός ή κλειστός. Για τον λόγο αυτό χρησιμοποιήθηκε το node "rpi gpio in" το οποίο επιστρέφει τιμή 0 ή 1 κάθε φορά που αλλάζει η κατάσταση. Ύστερα αφού έρθει μια κατάσταση οποιουδήποτε αισθητήρα γίνεται έλεγχος για το αν θα πρέπει να σημάνει συναγερμός. Αυτό γίνεται με απλούς ελέγχους global μεταβλητών που γίνονται initiated κατά την εκκίνηση του server.



Σχήμα 4 : Node-Red "rpi gpio in" nodes

4.1.2 Nodes ενημερώσεων AMQP

Σε οποιαδήποτε αλλαγή κατάστασης γίνεται ενημέρωση όλων των χρηστών με fan out διαδικασία. Συγκεκριμένα ενημερώνονται για την κατάσταση όλων των αισθητήρων, την κατάσταση arm (δηλαδή αν είναι οπλισμένο ή όχι το σύστημα), την κατάσταση alarm (δηλαδή αν υπάρχει συναγερμός ή όχι) και τέλος τον λόγο ενημέρωσης (πχ "updateStatus") ώστε να γνωρίζουν πως να χρησιμοποιήσουν οι clients το μήνυμα. Όλα αυτά τα states και το reason μετατρέπονται σε μορφή json και με το node "join" συνδιάζονται σε ένα ενιαίο μήνυμα που θα σταλθεί με το "amqp out" node. Αυτό φαίνεται στο Σχήμα 5.



Σχήμα 5 : Amqp

4.1.3 Nodes ενεργειών

Για οποιαδήποτε ενέργεια του χρήστη να οπλίσει, αφοπλίσει να κάνει bypass κάποιες ζώνες (δηλαδή να εξαιρέσει από τον συναγερμό κάποιες ζώνες), να αλλάξει ρυθμίσεις, γίνεται χρήση του "http in" node. Στην ουσία κάθε client για να αλληλεπιδράσει με τον συναγερμό στέλνει http requests στον server.

4.1.4 Other nodes

Η σύνδεση της σειρήνας γίνεται με την χρήση ενός usb 4-relay της denkovi. Η ενεργοποίηση της γίνεται με το node "exec" στο οποίο σαν είσοδο δίνονται εντολές terminal τις οποίες παρέχει ο κατασκευαστής του ([Denkovi usb Relay Board official page](#), n.d.). Το σύστημα ελέγχει ανά δευτερόλεπτο το alarm state με την χρήση του node "mytimeout" ([Node-Red mytimeout node](#), n.d.) και αν αλλάξει ενεργοποιεί ή απενεργοποιεί αντίστοιχα τη σειρήνα. Τέλος όλες οι καταστάσεις αποθηκεύονται στην βάση με απλά sql queries με την χρήση του node "mysql".

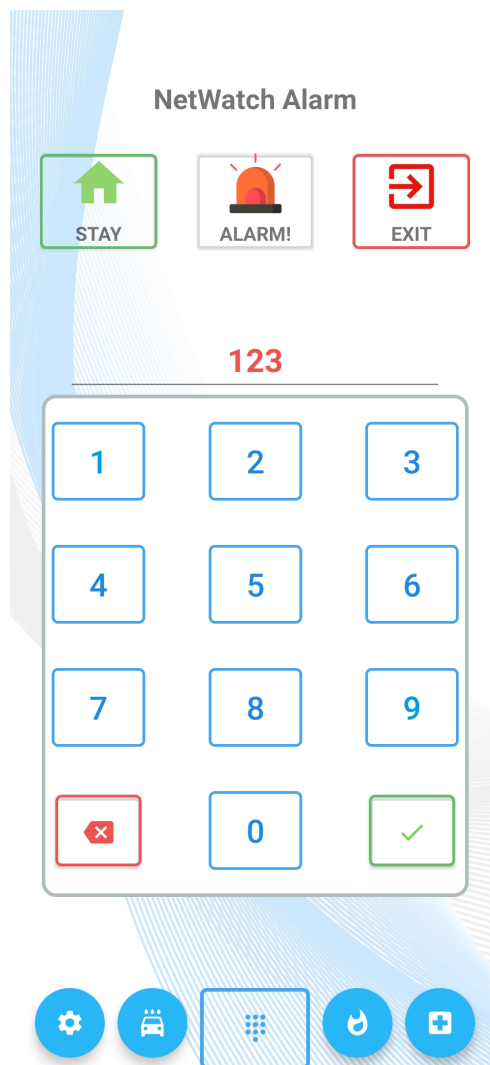
4.1.5 Android client

Για τον χειρισμό του συστήματος υλοποιήθηκε Android εφαρμογή. Συγκεκριμένα χρησιμοποιήθηκε η βιβλιοθήκη της rabbitmq ([RabbitMq official java library](#), n.d.) ώστε κάθε client να κάνει subscribe και να δέχεται amqp ειδοποιήσεις. Παρακάτω στο Σχήμα 6 φαίνεται η κυρία οθόνη της εφαρμογής.



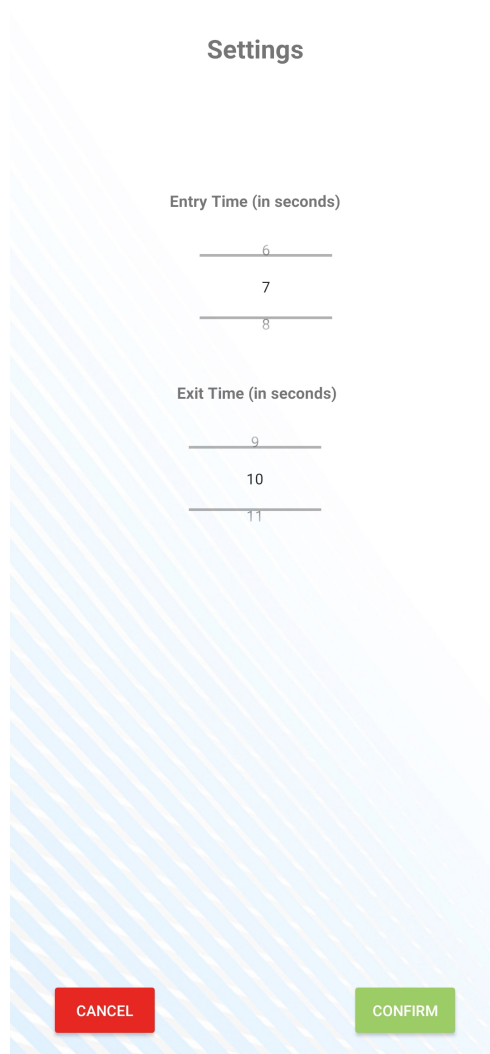
Σχήμα 6 : Android "NetWatch-Alarm" dashboard.

Όπως φαίνεται στο Σχήμα 6 με πράσινο χρώμα εμφανίζονται οι κλειστές ζώνες, με κίτρινο οι ζώνες οι οποίες έχουν εξαιρεθεί και με γκρι χρώμα οι ζώνες που δεν αντιστοιχούν σε κάποιο αισθητήρα. Στην περίπτωση συναγερμού εμφανίζεται ένα πληκτρολόγιο για να γράψει τον κωδικό ώστε να αποπλίσσει το σύστημα (Σχήμα 7).



Σχήμα 7 : Android "NetWatch-Alarm" show keypad on alarm state.

Για να το οπλίσει έχει δύο επιλογές στο πάνω μέρος της οθόνης, Home και Away. Με το πάτημα του κουμπιού home το σύστημα συναγερμού οπλίζει περιμετρικά το σπίτι αγνοώντας τους αισθητήρες κίνησης καθώς ο χρήστης θα παραμείνει μέσα στον χώρο και θέτει την κατάστασή του σε άμεση στην περίπτωση διάρρηξης. Αντίθετα αν πατήσει το κουμπί away το σύστημα του δίνει κάποια δευτερόλεπτα που έχει ο ίδιος επιλέξει στις ρυθμίσεις για να φύγει. Στην περίπτωση αυτή ενεργοποιούνται και τα radar. Στην περίπτωση που ανοίξει η πόρτα πάλι δίνεται στο χρήστη χρονικό περιθώριο να βάλει τον κωδικό του. Τέλος με το κουμπί settings ο χρήστης μεταφέρετε σε ένα 2ο activity στο οποίο μπορεί να αλλάξει τους χρόνους εισόδου και εξόδου (Σχήμα 8).



Σχήμα 8 : Android "NetWatch-Alarm" settings menu.

4.2 Λεπτομέρειες υλοποίησης συστήματος smarthome

4.2.1 Spring controller

Για το mapping όλων των διευθύνσεων της εφαρμογής χρησιμοποιήθηκε μία κλάση "homecontroller" η οποία έγινε annotate ως controller. Εκεί διαχειριζόμαστε όλα τα δεδομένα από και προς το χρήστη, τα οποία προέρχονται από http requests. Η κλάση αυτή αποτελείται από μεθόδους που έχουν annotation RequestMapping παρέχοντας ένα path συνδέσμου. Στις μεθόδους αυτές γίνεται έλεγχος των δεδομένων που υποβάλλει ο χρήστης πριν αποθηκευτούν στην βάση δεδομένων. Τέλος κάθε μέθοδος επιστρέφει το όνομα ενός view το οποίο είναι υπεύθυνο για την απόδοση των jsp σελίδων.

4.2.2 Spring security

Για την ασφάλεια του συστήματος χρησιμοποιήθηκε το spring security framework. Με αυτό γίνεται η αυθεντικοποίηση και η εξουσιοδότηση του χρήστη. Συγκεκριμένα για λόγους πειραματικούς στην κλάση "WebSecurityConfig" δημιουργήθηκαν δύο app configuration adapters. Στον πρώτο που αφορά τον admin χρησιμοποιήθηκε annotation order με παράμετρο 1. Σε αυτόν χρησιμοποιήθηκε in Memory Authentication και ρόλο "admin". Στον δεύτερο adapter χρησιμοποιήθηκε annotation order με παράμετρο 2. Αντίθετα σε αυτόν τον adapter που αφορά στον χρήστη και έχει role "user" χρησιμοποιήθηκε αυθεντικοποίηση μέσω βάσης δεδομένων (jdbcAuthentication) αφού πρώτα ρυθμίστηκε το datasource στην κλάση "AuthenticationProviderConfig". Το πρόβλημα όμως ήταν ότι το σχήμα της βάσης μου δεν συμβαδίζει με το default πρότυπο του Spring που ζητάει δύο tables (users, roles). Οπότε παραμετροποιήθηκαν τα κλασσικά queries ώστε να επιστρέφουν τα ίδια αποτελέσματα όπως φαίνεται στο σχήμα 0.0. ΣΧΗΜΑ ΕΔΩ Και στις δύο περιπτώσεις ρυθμίστηκαν τα "antMatchers" ανάλογα με τους καταλόγους που πρέπει να έχει ο καθένας. Για παράδειγμα αν κάποιος ζητήσει τον κατάλογο "/user/dashboard" τότε θα μπορεί να συνδεθεί μόνο με τα credentials ενός χρήστη και όχι διαχειριστή. Στην περίπτωση που υπήρχε κοινός κατάλογος λειτουργεί με σειρά που έχει οριστεί στο annotation order. Τέλος κάθε κωδικός γίνεται encode με τη χρήση συνάρτησης bcrypt πριν αποθηκευτεί στην βάση και decode κατά την διαδικασία αυθεντικοποίησης. Παρακάτω στο Σχήμα 9 φαίνεται η αυθεντικοποίηση μέσω της βάσης δεδομένων καθώς και οι ρυθμίσεις καταλόγων για το ρόλο χρήστη.

```

@Configuration
@Order(2)
public static class App1ConfigurationAdapter extends WebSecurityConfigurerAdapter {
    public App1ConfigurationAdapter() {
        super();
    }
    @Override
    public void configure(WebSecurity web) throws Exception {
        web
            .ignoring()
            .antMatchers("/resources/**");
    }
    @Autowired
    DataSource dataSource;
    //AUTHENTICATION PROVIDER JDBC
    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.jdbcAuthentication().dataSource(dataSource)
            .usersByUsernameQuery("select username,password,'true' as enabled from users where username = ?")
            .authoritiesByUsernameQuery("select username, type as authority from users where username = ?").passwordEncoder(passwordEncoder());
    }
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        System.out.println("=====user=====");
        http
            .antMatcher("/user/**").authorizeRequests()
            .antMatchers("/user/login").permitAll()
            .antMatchers("/user/**")
            // .hasRole("USER")
            .hasAuthority("ROLE_USER") // Στην βάση είναι "USER"
            .and()
            .formLogin()
            .loginPage("/user/login")
            .permitAll()
            .and()
            .logout()
            .logoutRequestMatcher(new AntPathRequestMatcher("/user/logout"));
    }
}

```

Σχήμα

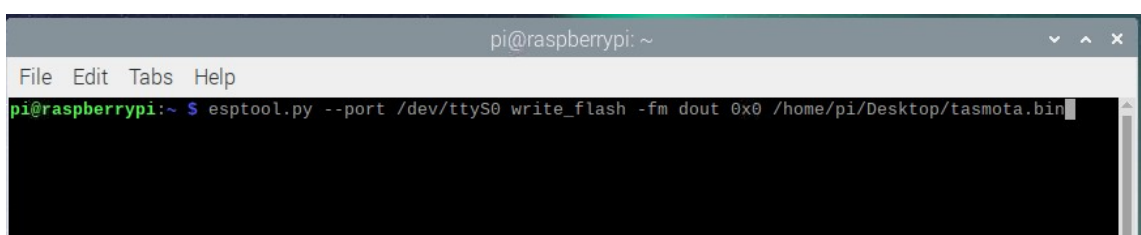
9 : Spring Web Security Config

4.2.3 Spring mqtt

Για την επικοινωνία του συστήματος έξυπνου σπιτιού και έξυπνων συσκευών μέσω mqtt χρησιμοποιήθηκε η βιβλιοθήκη maven paho mqtt ([Maven repository mqtt client link](#), n.d.). Δημιουργήθηκε μια μέθοδος subscribe στην οποία παίρνω από την βάση δεδομένων όλα τα topics των συσκευών και με μια επανάληψη φτιάχνω ένα mqtt client ώστε να κάνω subscribe για κάθε συσκευή. Κάθε αλλαγή κατάστασης έξυπνης συσκευής την αποθηκεύω στην βάση. Ομοίως δημιουργήθηκε μια μέθοδος unsubscribe.

4.2.4 Tasmota

Για την επίτευξη της επικοινωνίας μεταξύ των έξυπνων συσκευών και του συστήματος ήταν αναγκαία η εύρεση ενός κοινού προτύπου που θα υποστηριζόταν ανεξάρτητος εταιρείας κατασκευής. Έτσι με κάθε έξυπνο διακόπτη που φέρει ESP8266 chip μπορεί να αλλάξει το εργοστασιακό firmware σε firmware της tasmota ([Tasmota official page](#), n.d.). Συγκεκριμένα δημιουργήθηκε εγγραφή του καινούργιου firmware με την βοήθεια του raspberry pi και βασικών εντολών, αφού εγκαταστάθηκε το esptool, όπως φαίνονται στο Σχήμα 10.



Σχήμα 10 : Esp flash command

Έπειτα παραμετροποιήθηκαν οι ρυθμίσεις wifi, ip και τα credentials για τον mqtt server καθώς και το topic. Επειδή το firmware αυτό της tasmota είναι το ίδιο για όλες τις συσκευές παραμετροποιήθηκε και το αντίστοιχο template που καθορίζει την φυσική λειτουργία του διακόπτη. Αποτέλεσμα αυτού ήταν να ενημερώνεται ο server μου κάθε φορά που άλλαζε η κατάσταση του διακόπτη με mqtt μηνύματα. Τα μηνύματα ήταν της μορφής stat/hostname-topic/power. Για παράδειγμα σε έναν διακόπτη τοίχου δύο καναλιών αν πατηθεί ο πρώτος διακόπτης θα έρθει μήνυμα "ON" ή "OFF" στο αντίστοιχο topic. Τέλος με την χρήση http requests άνοιγα και έκλεινα τον διακόπτη από τον Spring controller με τα εξής url:

“http://<ip>/cm?cmnd=Power%20TOGGLE”

“http://<ip>/cm?cmnd=Power%20On”

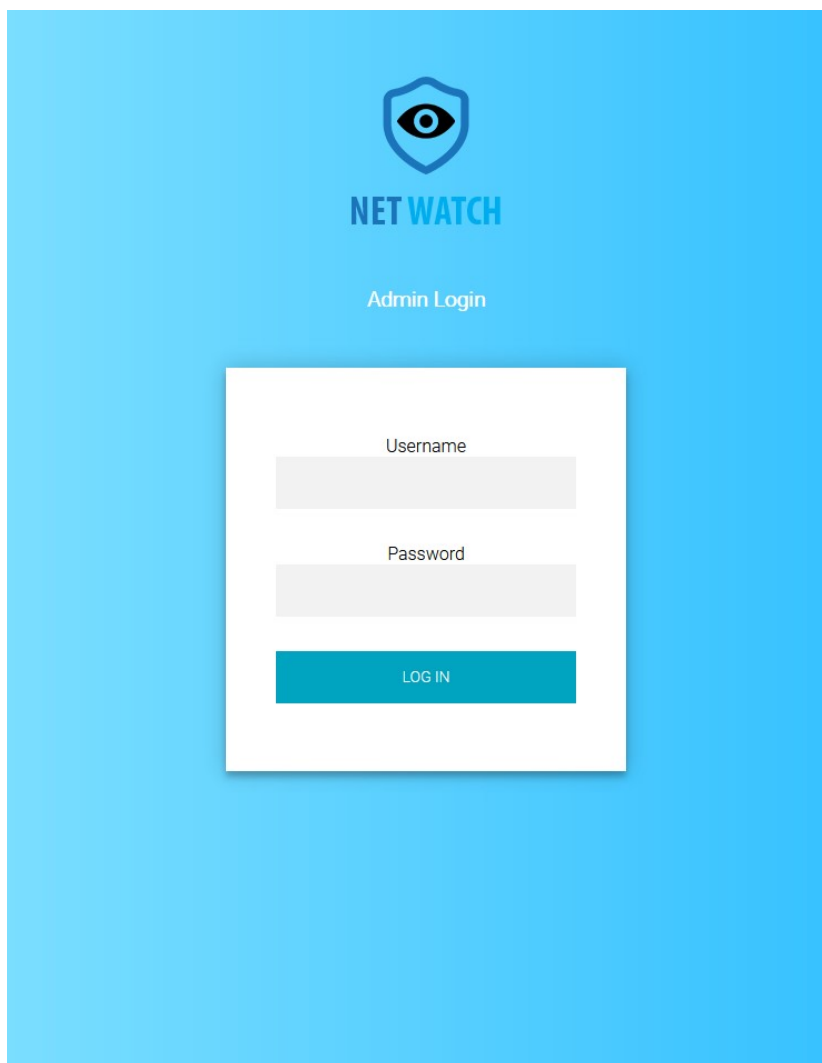
“http://<ip>/cm?cmnd=Power%20off”

Στην περίπτωση που ήθελα τους έξυπνους διακόπτες να έχουν κωδικό ώστε να μην είναι προσβάσιμοι χωρίς αυτό στο webui τους έβαζα δύο έξτρα παραμέτρους στο url. Οπότε γινόταν της μορφής “http://<ip>/cm?user=admin&password=tost123&cmnd=Power%20Toggle”.

4.2.5 User Interface

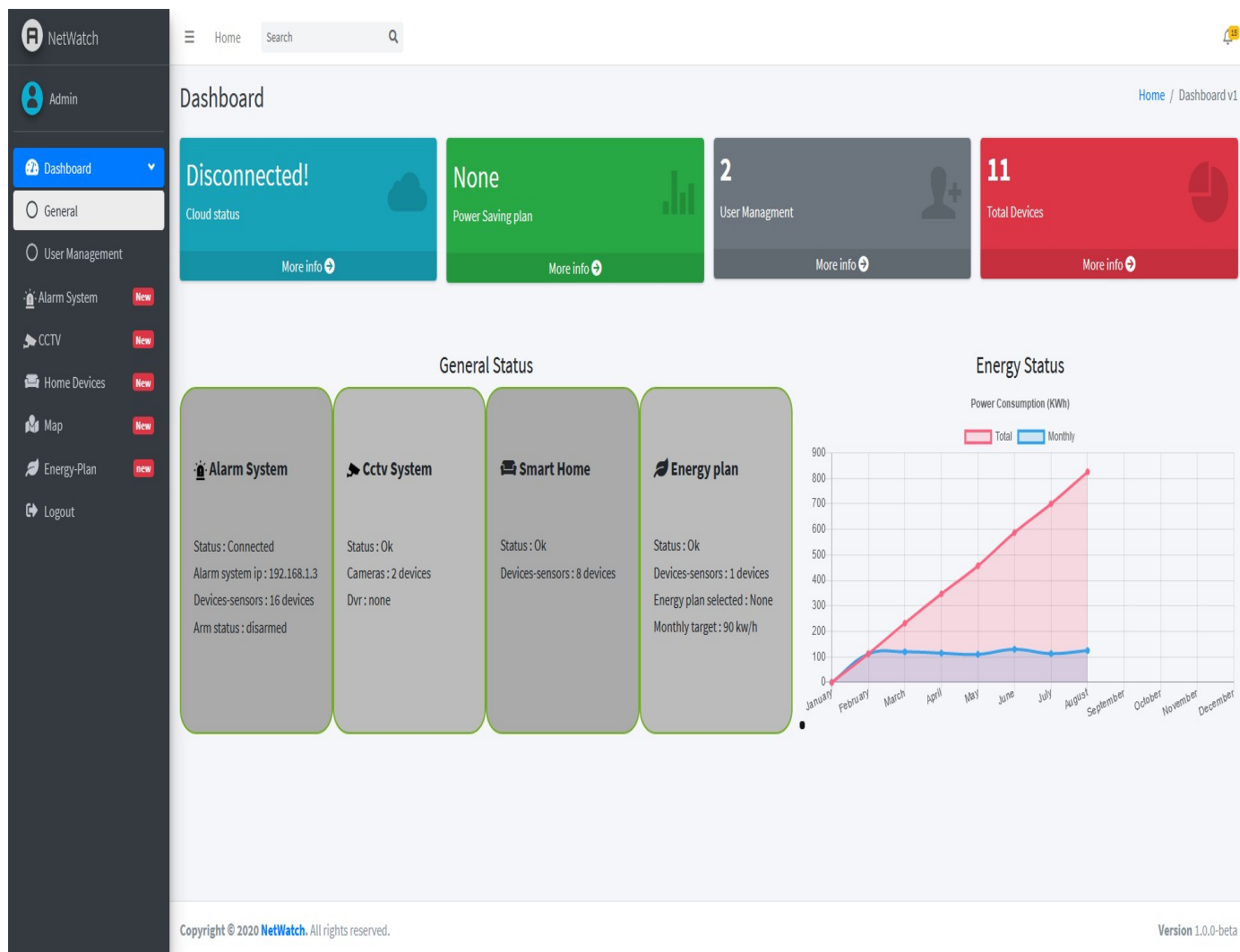
Για την δημιουργία φιλικού προς το χρήστη γραφικού περιβάλλοντος χρησιμοποιήθηκαν jsp σελίδες με html, bootstrap και συγκεκριμένα το adminlte template ([AdminLte bootstrap template official page](#), n.d.), css καθώς και jquery και javascript για την λειτουργικότητά τους. Για την δημιουργία των διαγραμμάτων χρησιμοποιήθηκε το Chart.js ([Chart js official page](#), n.d.). Στα παρακάτω σχήματα φαίνονται screenshots από το γραφικό περιβάλλον του διαχειριστή και του χρήστη.

Όλοι οι κατάλογοι που βρίσκονται κάτω από τον /admin/* προστατεύονται με στοιχεία ρόλου admin. Επόμενος πριν έχει πρόσβαση στο μενού θα πρέπει να κάνει σύνδεση πρώτα στο παρακάτω μενού. Στην περίπτωση που κάνει request σε έναν κατάλογο εντός /admin/* γίνεται redirect στο login page /admin/login.

The image shows a web page for 'NETWATCH' with a blue gradient background. At the top center is a logo consisting of a shield with an eye inside, and the text 'NETWATCH' below it. Underneath the logo is the text 'Admin Login'. In the center of the page is a white rectangular box containing a login form. The form has two input fields: 'Username' and 'Password', each with a light gray border. Below these fields is a blue button with the text 'LOG IN' in white capital letters.

Σχημα 11 : Admin login

Στο dashboard του admin υπάρχουν βασικές πληροφορίες για την κατάσταση των δύο συστημάτων όπως αριθμός χρηστών, αριθμός συσκευών, κατάσταση και ip του alarm system, κατάσταση ενεργειακού πλάνου και άλλα. Είναι ο πρώτος κατάλογος που γίνεται redirect ο admin σε περίπτωση επιτυχούς αυθεντικοποίησης.



Σχημα 12 : Admin Dashboard

Στον κατάλογο user management εμφανίζεται λίστα με τους ήδη υπάρχον χρήστες καθώς και μια φόρμα με πεδία username, password, usertype (admin ή user), fullname, mobile phone, email και alarm (optional πεδίο) ώστε να μπορεί να προσθέσει χρήστες.

The screenshot displays the 'User Management' section of the NetWatch application. On the left is a dark sidebar with the 'NetWatch' logo and a menu including 'Admin', 'Dashboard', 'General', 'User Management', 'Alarm System', 'CCTV', 'Home Devices', 'Map', 'Energy-Plan', and 'Logout'. The main area has a light blue header with 'Home' and a search bar. Below the header, the 'User Management' title is followed by a breadcrumb 'Home / User Management'. A modal window titled 'Users' is open, showing a table of existing users:

User #	User Name	User Type	Full name	Phone	Email	Alarm Access
0	billder7	admin	bill dervisiz	6983914843	billder@windowslive.com	yes
1	user1	USER	test	6999999999	mail	no

To the right of the table is a green 'Add new user' button. Below it is a form with the following fields: 'Username' (with a note 'Special symbols are not allowed.'), 'Password', 'User Type' (a dropdown menu currently set to 'User'), 'Fullname', 'Mobile Phone', 'Email' (with a note 'This is used in case you forget password!'), and 'Alarm' (with a note 'This is optional. In case you dont have one leave it empty!'). A blue 'Register' button is at the bottom of the form.

Σχημα 13 : Admin User Management

Στον κατάλογο alarm system εμφανίζονται σε λίστα όλες οι ζώνες (αισθητήρα/ες) του συστήματος συναγερμού. Σε αυτό μπορεί να προσθέσει ονόματα σε κάθε ζώνη. Αυτό επιτυγχάνεται με http requests στον controller. Η κατάσταση από τις ζώνες ενημερώνεται μια φορά το δευτερόλεπτο. Το σύστημα συναγερμού στέλνει amqr μηνύματα στο σύστημα smarthome το οποίο αποθηκεύει την κατάσταση των ζωνών στην βάση.

The screenshot displays the 'Alarm System Settings' page in the NetWatch admin interface. On the left is a dark sidebar with navigation links. The main area features a table of alarm sensors and a status panel on the right.

#	Device Name	Device Type	Sensor State	Connection	
#	Zone 1 eisodos	magnet	Closed	Online	Edit
#	Zone 2 radar-saloni	radar	Bypassed	Online	Edit
#	Zone 3 radar2	radar	Bypassed	Online	Edit
#	Zone 4 saloni-upnodomatio	magnet	Bypassed	Online	Edit
#	Zone 5 bill upnodomatio	magnet	Bypassed	Online	Edit
#	Zone 6 bill radar	radar	Bypassed	Online	Edit
#	Zone 7 jim upnodomatio	magnet	Bypassed	Online	Edit
#	Zone 8 name8	other	Unknown	Offline	Edit
#	Zone 9 name9	other	Unknown	Offline	Edit
#	Zone 10 name10	other	Unknown	Offline	Edit

Alarm status [Refresh](#)

Alarm ip: 192.168.1.3

Arm state: Off

Alarm state: Off

Bypassed Zones: 1,2,3,4,5,6,7

Total Users: 3

Last updated: 23-8-2020 19:51:21 23-8-2020 19:51:21

Σχημα 14 : Admin Alarm System

Στο μενού cctv settings εμφανίζεται λίστα με τις ήδη υπάρχων κάμερες αλλά και μια φόρμα με πεδία(device name, device type, channel 0-16, device ip,device vendor, username και password) για να μπορεί να προσθέσει και άλλες.

The screenshot displays the 'Cctv Settings' page in the NetWatch application. On the left, a dark sidebar contains navigation links: Admin, Dashboard, Alarm System, CCTV (highlighted), Home Devices, Map, Energy-Plan, and Logout. The main content area has a header with 'Home', 'Contact', and a search bar. Below the header, the 'Cctv Settings' section features a table titled 'Cctv Devices' with columns for #, Device Name/Type, Switch Ip, Channel, and Connection. A single device is listed: #0, camera1 Camera, 192.168.1.64, Channel 1, with a 'Connected' status. To the right of the table are 'Edit' and 'Delete' buttons. Adjacent to the table is a blue 'Add Device' form. This form includes input fields for Device Name, Device Type (set to Camera), Select Channel (set to 0), Device ip (192.168.1.x), Device Vendor (Hikvision), Username (admin), and Password (masked with asterisks). A blue 'Submit' button is at the bottom of the form. The footer of the page shows 'Copyright © 2020 NetWatch. All rights reserved.' and 'Version 1.0.0-beta'.

#	Device Name/Type	Switch Ip	Channel	Connection
0	camera1 Camera	192.168.1.64	1	Connected

Add Device

Device Name:

Select Device Type:

Select Channel:

Device ip:

Device Vendor:

Username:

Password:

Σχημα 15 : Admin Cctv

Στο μενού home devices βρίσκεται λίστα με όλες τις έξυπνες συσκευές καθώς βασικές πληροφορίες για αυτές όπως η ip και το connection το οποίο δείχνει αν έχουν επικοινωνία με το σύστημα. Επίσης βρίσκεται μια φόρμα με την οποία μπορεί να προσθέσει και άλλες συσκευές. Στο πεδίο device type μπορεί να επιλέξει τον τύπο της συσκευής (Light, Oven, Shutters, Garage door, Water heater, Washing machine, House heater, Air condition, other). Στο πεδίο select switch port επιλέγει πόσες θύρες έχει ο έξυπνος διακόπτης. Στα πεδία name, ip και topic δίνει όνομα συσκευής που επιθυμεί την διεύθυνση της συσκευής και το topic το οποίο έχει ορίσει στην συσκευή αντίστοιχα.

The screenshot displays the 'Smart Devices Settings' page in the NetWatch Admin interface. On the left is a sidebar with navigation options: Dashboard, Alarm System, CCTV, Home Devices (highlighted), Map, Energy-Plan, and Logout. The main content area is divided into two sections. The top section, titled 'Smart Devices', contains a table listing existing devices. The bottom section, titled 'Add Device', provides a form to register a new smart device.

#	Device Name/Type	Switch Ip	Switch Port	Connection	
0	KitchenLight Light	192.168.1.55	1	Connected	Edit Delete
1	Hall Light Light	192.168.1.56	1	Connected	Edit Delete
2	Entry Light Light	192.168.1.57	1	Connected	Edit Delete
3	Washing-machine1 Washing machine	192.168.1.58	1	Connected	Edit Delete
4	SonOfMqtt Light	192.168.1.36	0	Connected	Edit Delete
5	ShuttersLivingRoomUp Shutters	192.168.1.5	1	Connected	Edit Delete
6	ShuttersLivingRoomDown Shutters	192.168.1.5	2	Connected	Edit Delete

The 'Add Device' form includes the following fields:

- Select Device Type:** A dropdown menu currently showing 'Light'.
- Switch Port:** A dropdown menu with the instruction '*If device has only one switch port select 0 Select Switch Port' and a value of '0'.
- Device Name:** A text input field with the placeholder 'Name'.
- Switch ip:** A text input field with the placeholder '192.168.1.x'.
- Switch Vendor:** A dropdown menu currently showing 'Sonoff'.
- Switch Topic:** A text input field with the placeholder 'ex. switch1'.
- Submit:** A blue button to save the new device.

At the bottom of the page, there is a copyright notice: 'Copyright © 2020 NetWatch. All rights reserved.' and the version information: 'Version 1.0.0-beta'.

Σχημα 16 : Admin Smarthome

Στο μενού map settings μπορεί να προσθέσει συσκευές που υπάρχουν ήδη σε διάφορα δωμάτια της επιλογής του καθώς και να δει τις λίστες με τις συσκευές ανά δωμάτιο.

The screenshot displays the 'Map Settings' page in the NetWatch application. The left sidebar contains navigation links: Dashboard, Alarm System, CCTV, Home Devices, Map (highlighted), Energy-Plan, and Logout. The main content area is titled 'Map Settings' and shows a list of rooms: Bedroom 1, Bedroom 2, Bedroom 3, Hall, Living room, and Kitchen. Each room has a corresponding table of devices. The 'Living room' table is expanded, showing two devices: 'ShuttersLivingRoomUp' and 'ShuttersLivingRoomDown', both with IP address 192.168.1.5 and Channel 1 and 2 respectively. Each device has a 'Connected' status and 'Edit' and 'Delete' buttons. On the right, there is a 'Add Device to map' panel with a 'Select Device' dropdown, a 'Select Room' dropdown (set to 'Livingroom'), and an 'Add' button.

#	Device Name/Type	Switch Ip	Channel	Connection
1	ShuttersLivingRoomUp Shutters	192.168.1.5	1	Connected
2	ShuttersLivingRoomDown Shutters	192.168.1.5	2	Connected

Σχημα 17 : Admin Map

Στο μενού energy εμφανίζεται λίστα με τις συσκευές μέτρησης ενέργειας από την βάση δεδομένων. Σε αυτό το μενού ο admin μπορεί να προσθέσει ενεργειακές συσκευές συμπληρώνοντας την πρώτη φόρμα με πεδία device name, device ip και device topic. Πρέπει να σημειωθεί ότι ο admin μπορεί να προσθέσει ήδη υπάρχων έξυπνο διακόπτη που μετράει και κατανάλωση ρεύματος. Σε αυτή την περίπτωση η μόνη διαφορά τους θα είναι το topic. Στην δεύτερη φόρμα διαλέγει το ενεργειακό πλάνο που επιθυμεί και το μηνιαίο συνολικό όριο σε kw/h. Στην τελευταία φόρμα προσθέτει το topic στο οποίο έρχονται ενημερώσεις για την θερμοκρασία του σπιτιού.

The screenshot displays the 'Energy Devices Settings' page in the NetWatch Admin interface. The left sidebar contains navigation links: Dashboard, Alarm System, CCTV, Home Devices, Map, Energy Plan (highlighted), and Logout. The main content area features a table of 'Smart Devices' and three configuration panels on the right.

#	Device Name	Device Ip	Device topic	Connection
0	energymeter1	192.168.1.200	energy1	Connected Delete

Add Energy Meter Device

Device Name:

Device ip:

Device Topic:

Submit

Energy Plan Settings

Energy plan:

Monthly user target (in kw/h):

Submit

Temperature Settings

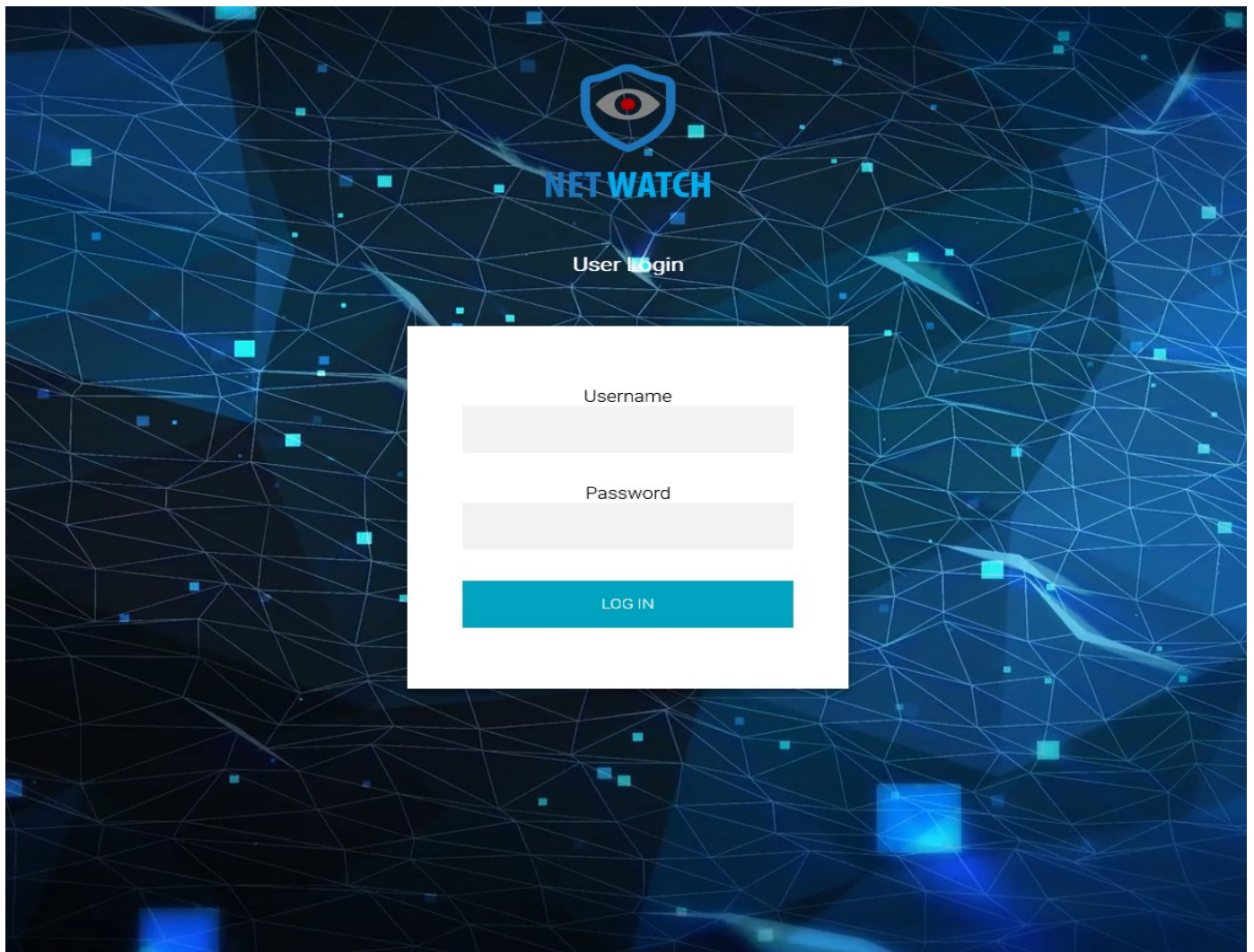
Set temp topic:

Submit

Copyright © 2020 NetWatch. All rights reserved. Version 1.0.0-beta

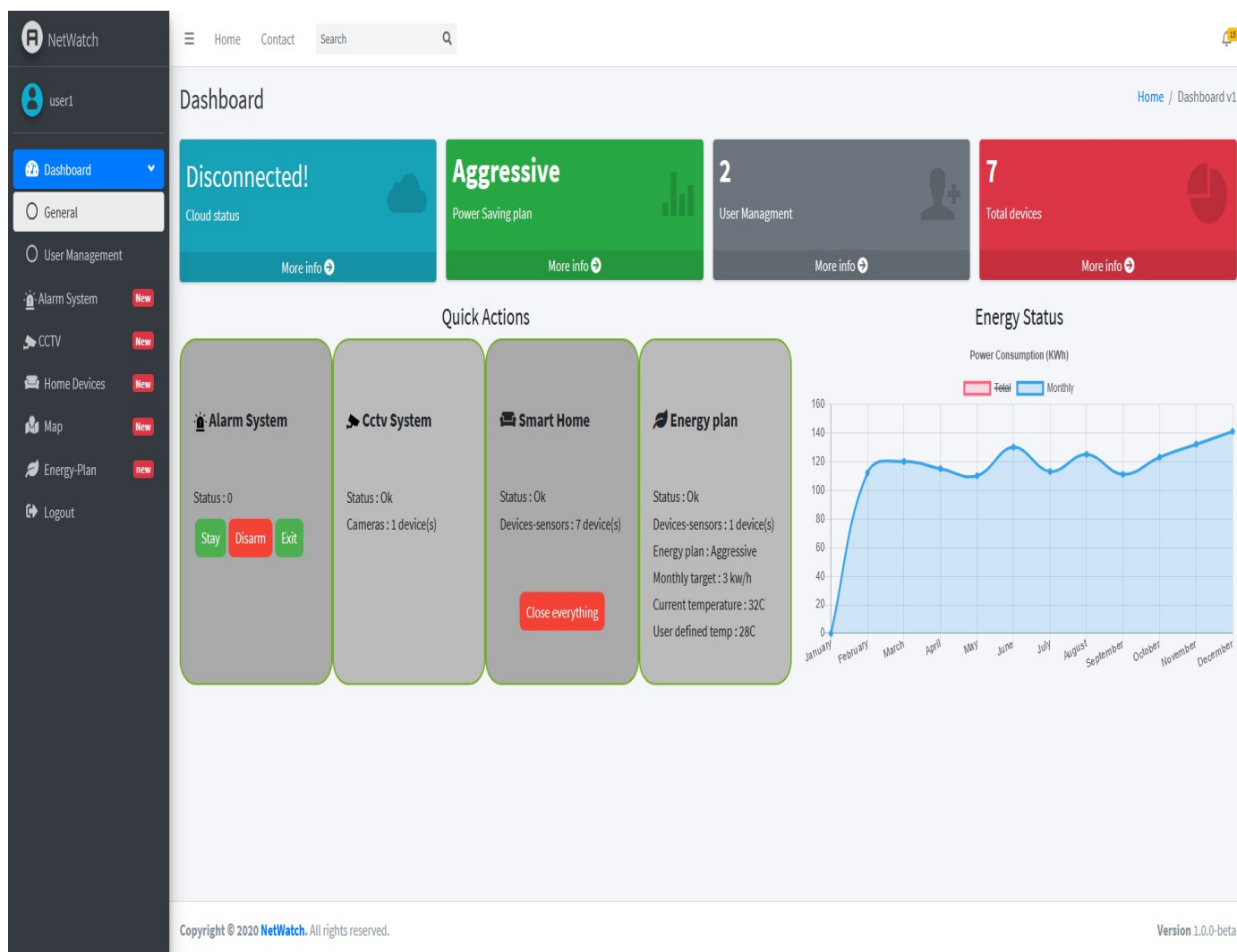
Σχημα 18 : Admin Energy

Όλοι οι κατάλογοι /user/* για να είναι προσβάσιμοι πρέπει να αυθεντικοποιηθεί χρήστης με ρόλο user. Σε κάθε άλλη περίπτωση γίνεται redirect στην σελίδα login του χρήστη. Σε αυτή χρησιμοποιήθηκε video background ώστε να διαφέρει από το login page του admin.



Σχήμα 19 : User Login

Στο μενού dashboard του χρήστη εμφανίζονται πάλι βασικές πληροφορίες όπως και στον admin με την διαφορά ότι σε αυτό μπορεί να ελέγξει τον συναγερμό (προϋπόθεση ότι έχει δώσει σωστό κωδικό alarm) και να κλείσει όλες τις συσκευές.



Σχημα 19 : User Dashboard

Στο μενού user management μπορεί να αλλάξει βασικές πληροφορίες του profile του, να προσθέσει alarm κωδικό καθώς και να αλλάξει κωδικό χρήστη.

NetWatch

Home Contact Search

user1

Dashboard

General

User Management

Alarm System New

CCTV New

Home Devices New

Map New

Energy-Plan New

Logout

User Management

Home / User Management

Change user information

Username

user1

Special symbols are not allowed.

User Type

USER

Fullname

first last name

Mobile Phone

69999999

Email

testemail@mail.com

This is used in case you forget password!

Alarm

•••••

This is optional. In case you don't have one leave it empty!

Update

User password change

Old password

Old Password

Password

New Password

Retype new password

Retype New Password

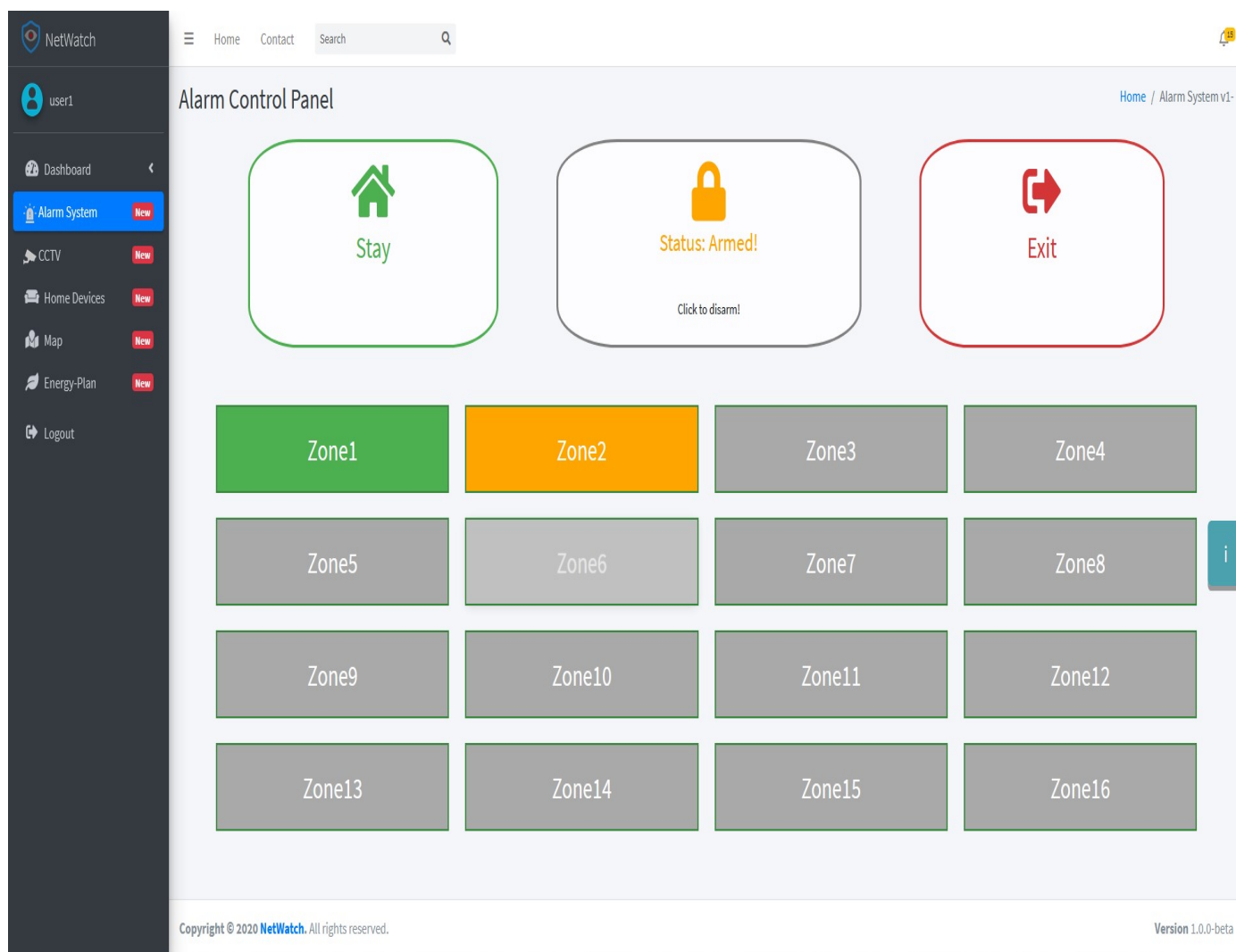
Update password

Copyright © 2020 NetWatch. All rights reserved.

Version 1.0.0-beta

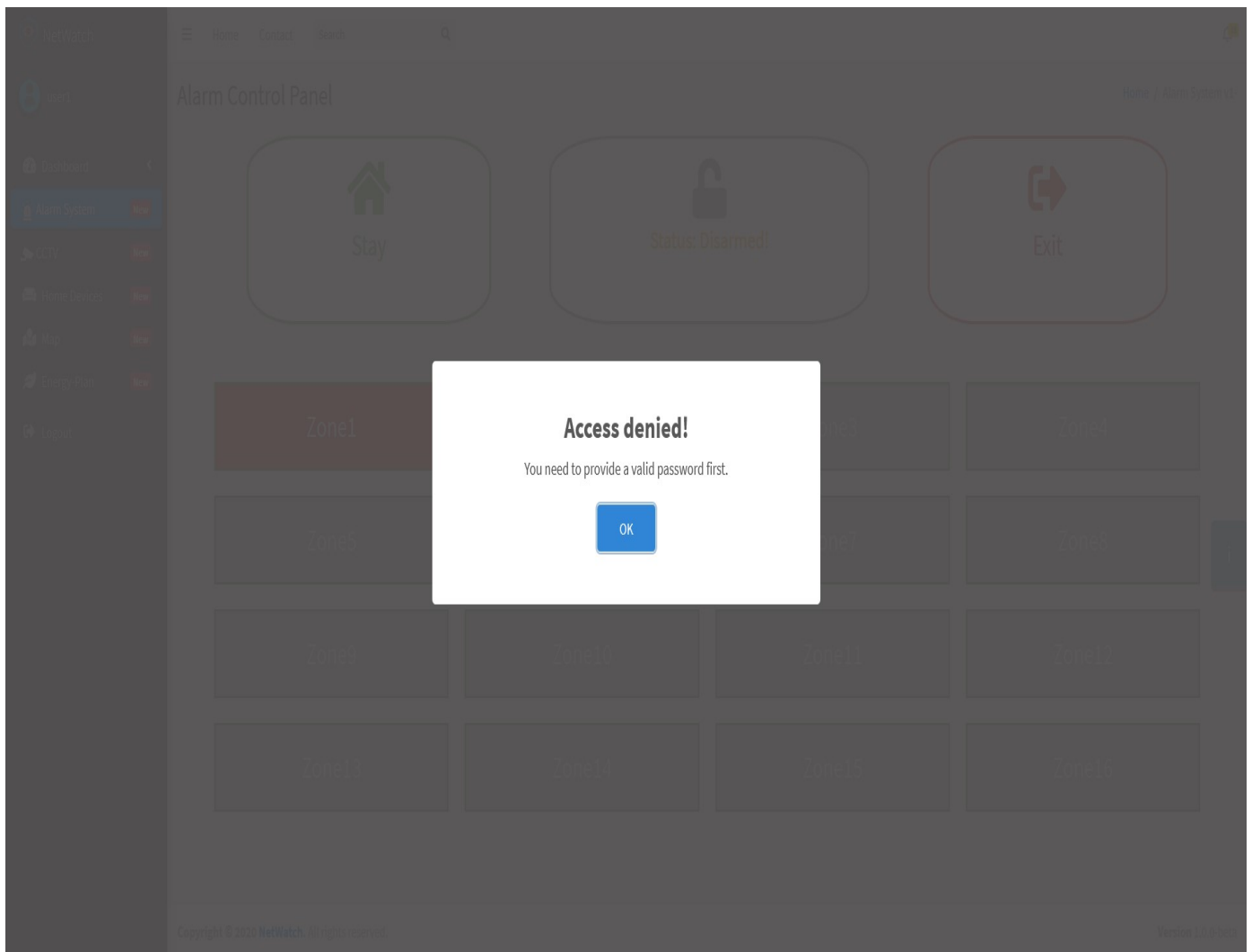
Σχήμα 20 : User User-Management

Στο μενού alarm system μπορεί να ελέγξει το σύστημα συναγερμού. Συγκεκριμένα πατώντας το κουμπί Stay ή exit οπλίζει τον συναγερμό και εμφανίζεται κατάλληλο μήνυμα αλλάζοντας επίσης το status σε armed. Για να τον αφοπλίσει αρκεί να πατήσει το μεσαίο κουμπί. Ακόμα εμφανίζεται λίστα με τις ζώνες που δείχνουν την κατάσταση τους ανάλογα με το χρώμα (πράσινο = ζώνη κλειστή, κόκκινο = ζώνη ανοιχτή, κίτρινο = ζώνη bypassed και γκρι = ζώνη ανενεργή - κενή.) Επίσης υπάρχει στα δεξιά ένα κουμπί info το οποίο περιέχει τις προαναφερθείσες πληροφορίες.



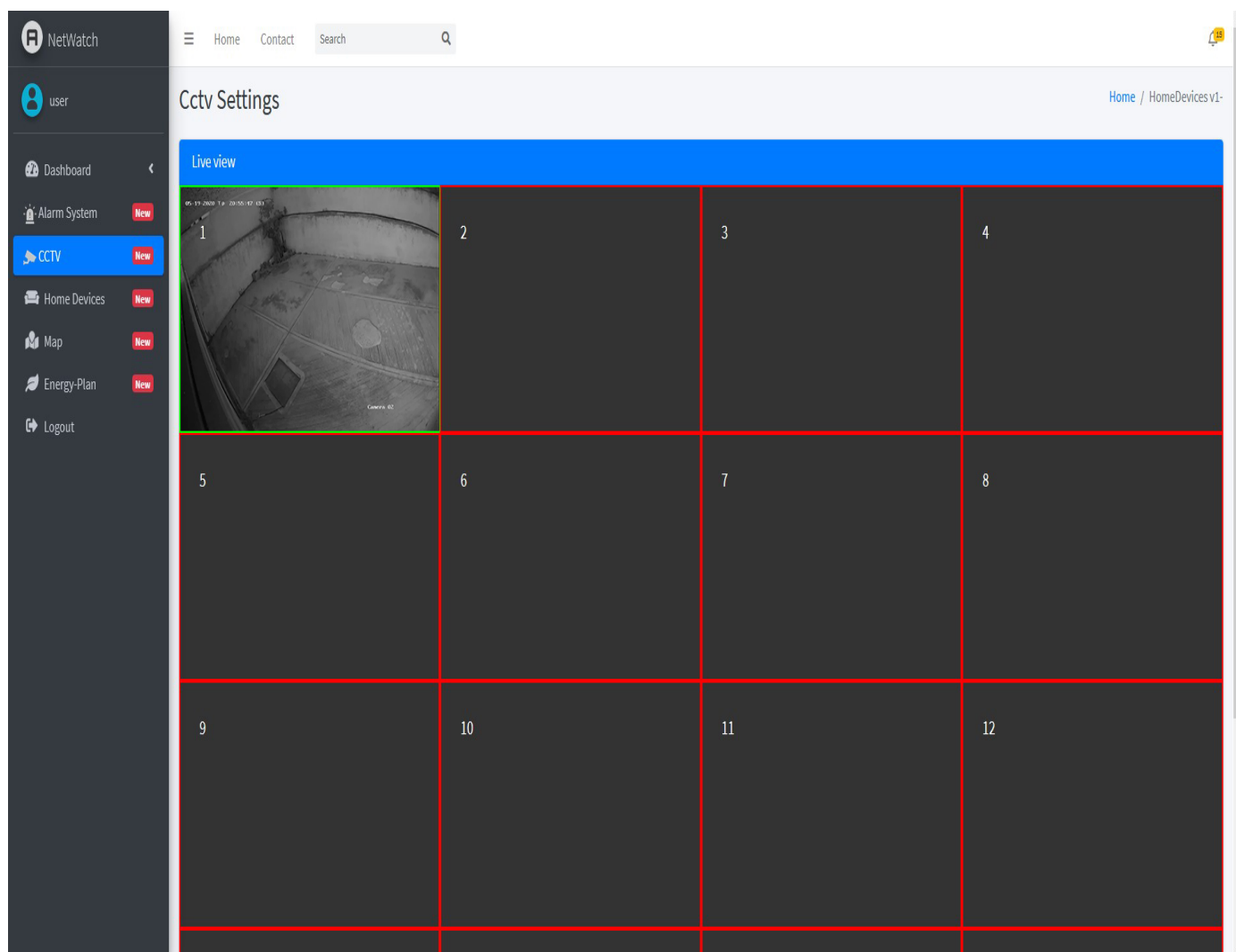
Σχημα 21 : User Alarm System

Στην περίπτωση που ο χρήστης δεν έχει δώσει κωδικό alarm ή έχει λανθασμένο κωδικό alarm τότε η περιορίζεται η πρόσβασή του στο μενού alarm system και εμφανίζεται κατάλληλο μήνυμα. Ύστερα γίνεται ανά κατεύθυνση στο μενού dashboard.



Σχημα 22 : User Alarm System Error

Στο μενού cctv ο χρήστης βλέπει όλες τις κάμερες που έχει προσθέσει ο admin(μέχρι 16 όσες και οι κάμερες που δέχεται ένα καταγραφικό 16 καναλιών). Για την προβολή εικόνας ο controller στέλνει στον client ένα link το οποίο και επιστρέφει live εικόνα jpg. Η κάθε εικόνα ανανεώνεται 2 φορές το δευτερόλεπτο με αποτέλεσμα ο χρήστης να βλέπει με 2 fps αντί του μέσου όρου που είναι 15 fps. Αυτός ο περιορισμός προκύπτει στο γεγονός ότι αρκετοί browsers δεν υποστηρίζουν rtsp ροές κατευθείαν.



Σχημα 23 : User Cctv

Στο μενού home devices εμφανίζεται λίστα με τα στοιχεία κάθε συσκευής όπου ο χρήστης μπορεί να ανοίξει και να κλείσει συσκευές. Το state (αν είναι κλειστή ή ανοιχτή) κάθε συσκευής ανανεώνεται μια φορά κάθε δευτερόλεπτο.

Smart Devices Settings

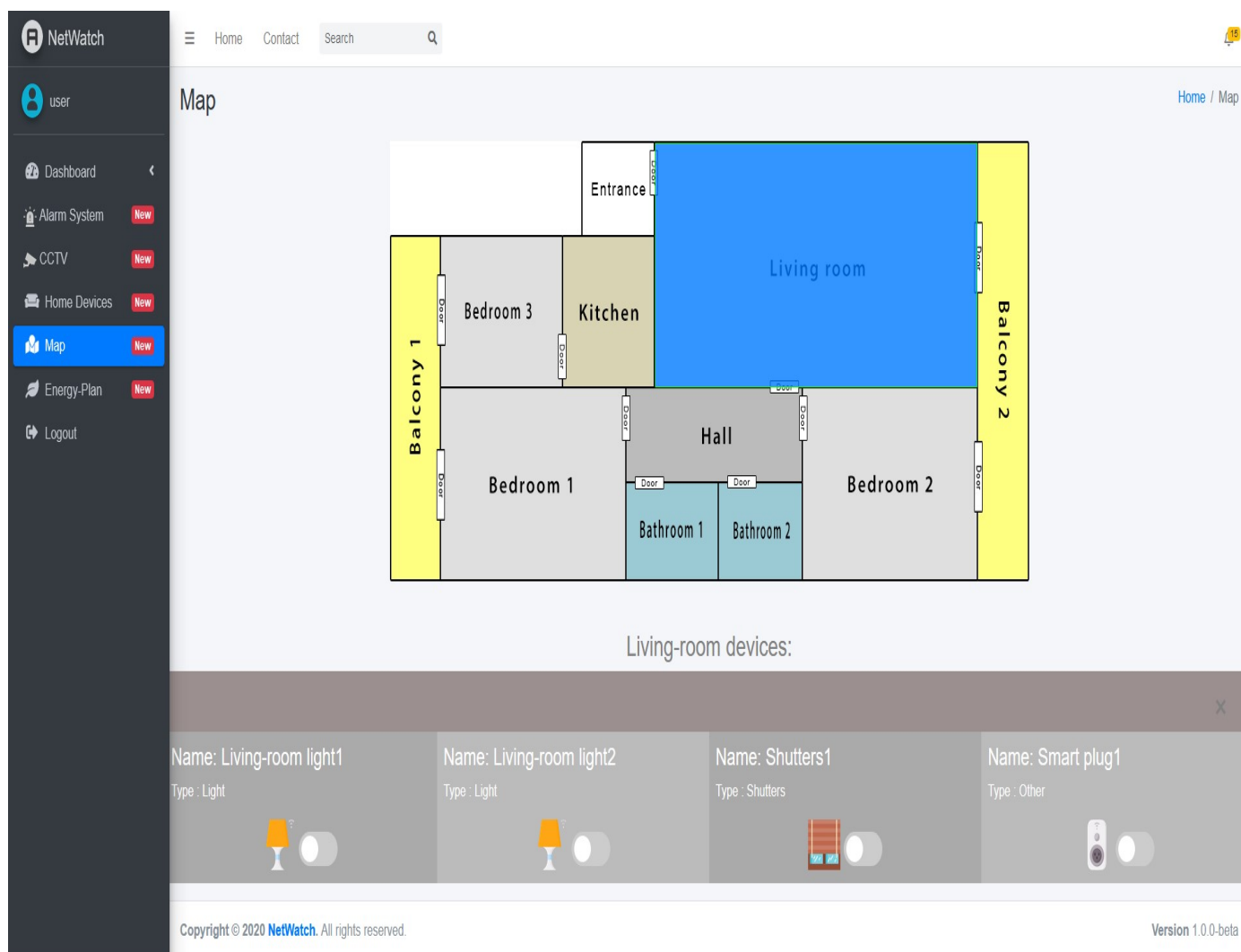
Smart Devices

#	Device Name/Type	Switch Ip	Switch Port	Connection	Action
0	KitchenLight Light	192.168.1.55	1	Connected	☒
1	Hall Light Light	192.168.1.56	1	Connected	☐
2	Entry Light Light	192.168.1.57	1	Connected	☐
3	Washing-machine1 Washing machine	192.168.1.58	1	Connected	☒
4	SonOffTestDev Light	192.168.1.36	0	Connected	☒
5	ShuttersLivingRoomUp Shutters	192.168.1.5	1	Connected	☐
6	ShuttersLivingRoomDown Shutters	192.168.1.5	2	Connected	☐

Copyright © 2020 NetWatch. All rights reserved. Version 1.0.0-beta

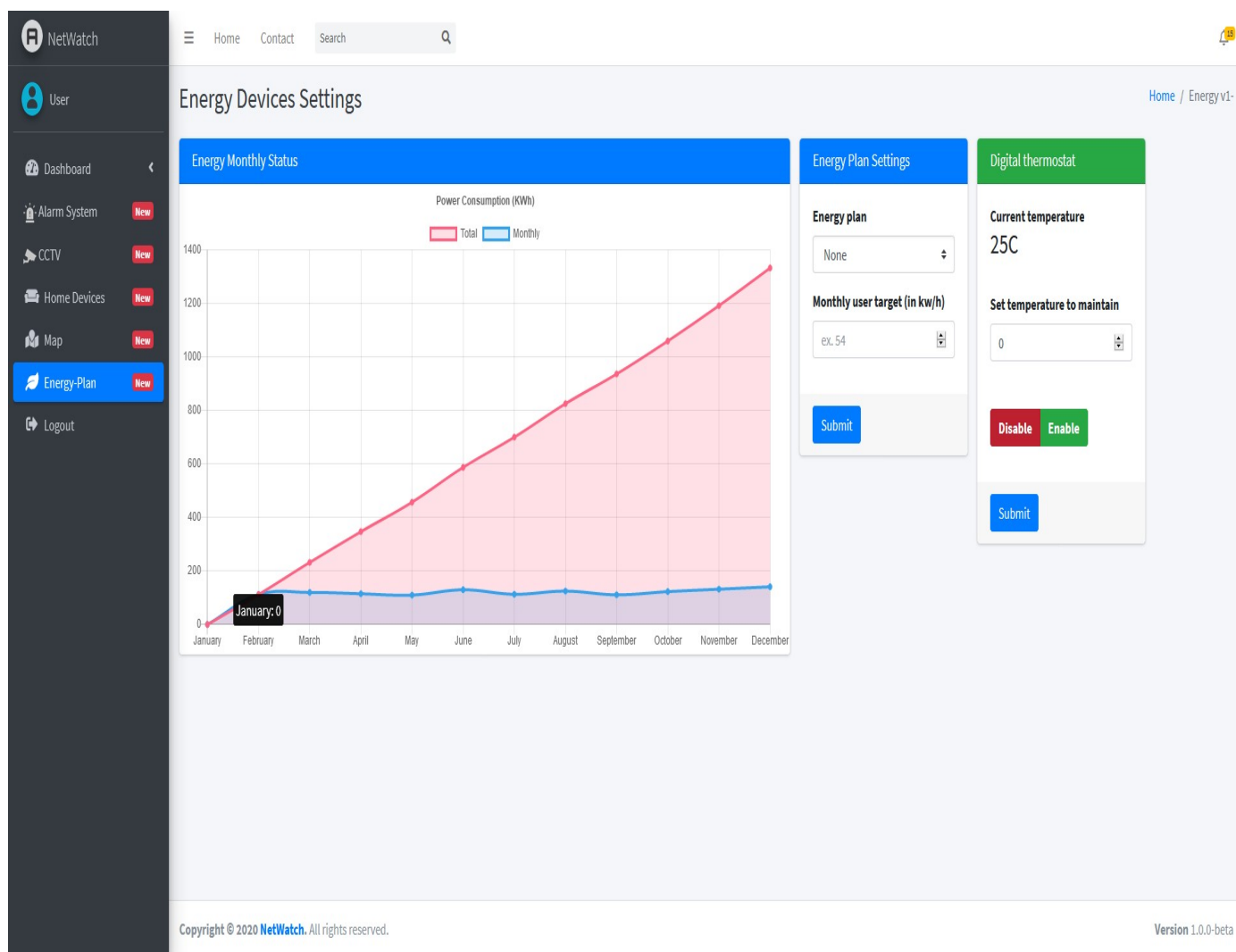
Σχημα 24 : User Smarthome

Στο μενού map ο χρήστης βλέπει τον χάρτη του σπιτιού. Από εκεί μπορεί να επιλέξει ένα δωμάτιο που επιθυμεί και θα εμφανιστεί στο κάτω μέρος της οθόνης λίστα με τις συσκευές καθώς και δυνατότητα ενεργοποίησης ή απενεργοποίησης αυτών. Στην περίπτωση που ένα δωμάτιο δεν έχει συσκευές εμφανίζεται κατάλληλο μήνυμα.



Σχημα 25 : User Map

Στο μενού energy-plan εμφανίζεται διάγραμμα που έχει ανά μήνα την κατανάλωση καθώς και την κατανάλωση συνολικά μαζί με τους προηγούμενους μήνες. Επίσης μπορεί να επιλέξει το ενεργειακό πλάνο που επιθυμεί όπως και ο admin. Αυτή η λειτουργία αφορά κυρίως φώτα τα οποία είναι ανοιχτά για μεγάλο χρονικό διάστημα. Ακόμα μπορεί να δει την θερμοκρασία του σπιτιού καθώς και να ορίσει μια θερμοκρασία που επιθυμεί ώστε το σύστημα να την διατηρήσει. Αυτή η λειτουργία προϋποθέτει να υπάρχει έστω μια συσκευή για κάθε τύπο (House heater, air condition).



Σχημα 26 : User Energy

5 Επίλογος

5.1 Συμπεράσματα

Στα πλαίσια της παρούσας πτυχιακής εργασίας θα γίνει επίδειξη Live Demo που να αναδεικνύει τη διαλειτουργικότητα των συσκευών και υπηρεσιών που αναπτύχθηκαν. Αυτό που παρατήρηθηκε καθόλη την διάρκεια της πτυχιακής μου ήταν ότι όσο εξελισσόταν και υλοποιούνταν ένα μέρος της εμφανίζονταν και άλλες ιδέες βελτίωσης και ανάπτυξης αυτής. Για παράδειγμα το γραφικό περιβάλλον άλλαζε συνεχώς εμφάνιση ώστε οι καινούργιες δυνατότητες και λειτουργίες να μην επηρεάζουν την ευχρηστία του συστήματος αλλά και να τον προδιαθέτουν τον χρήστη να τις χρησιμοποιήσει.

5.2 Μελλοντικές Επεκτάσεις

Με το πέρας αυτής της πτυχιακής θα ασχοληθώ με την ανάπτυξη και διαρκή υποστήριξη των δύο συστημάτων. Συγκεκριμένα όσον αφορά στο smarthome θα ήθελα να προσθέσω και άλλες λειτουργίες όπως video analytics (π.χ. Motion detection) σε υποστηριζόμενες κάμερες που θα βοηθούσε στον τομέα της ασφάλειας του κτιρίου. Ακόμα βασικός στόχος είναι η δημιουργία κάποιας Android εφαρμογής ώστε να έχει ο χρήστης ακόμα πιο εύκολη πρόσβαση και να μην χρειάζεται κάποιον browser. Επίσης θα ήθελα να υλοποιήσω κάποιον cloud server ώστε να έχει και απομακρυσμένη πρόσβαση. Χρήσιμη θα ήταν και η ύπαρξη δυνατότητας δυναμικής δημιουργίας σεναρίων με βάση το χρόνο και άλλων ενδείξεων. Όσον αφορά το alarm system θα ήθελα να υλοποιήσω την δυνατότητα σύνδεσης με κέντρο λήψης σημάτων ώστε να μπορεί να ειδοποιείται κατευθείαν η εταιρία σε περίπτωση διάρρηξης όπως γίνεται μέχρι τώρα στα κλασικά συστήματα.

Αναφορές

- Adminlte bootstrap template official page.* (n.d.). <https://adminlte.io/>.
- Chart js official page.* (n.d.). <https://www.chartjs.org/>.
- Denkovi usb relay board official page.* (n.d.). <https://denkovi.com/usb-relay-board-four-channels-for-home-automation-v2>.
- Mariadb official page.* (n.d.). <https://mariadb.org/about/>.
- Maven repository mqtt client link.* (n.d.). <https://mvnrepository.com/artifact/org.eclipse.paho/org.eclipse.paho.client.mqttv3>.
- Node-red mytimeout node.* (n.d.). <https://www.npmjs.com/package/node-red-contrib-mytimeout>.
- Node-red official page.* (n.d.). <https://nodered.org/about/>.
- Node-red wiki page.* (n.d.). <https://en.wikipedia.org/wiki/Node-RED>.
- Rabbitmq official java library.* (n.d.). <https://www.rabbitmq.com/java-client.html>.
- Rabbitmq official mqtt plugin.* (n.d.). <https://www.rabbitmq.com/mqtt.html>.
- Raspberry pi official page.* (n.d.). <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>.
- Smart home worldwide market users.* (n.d.). <https://www.statista.com/outlook/283/100/smart-home/worldwide#market-revenue>.
- Spring framework official page.* (n.d.). <https://docs.spring.io/spring/docs/4.3.x/spring-framework-reference/html/overview.html>.
- Tasmota official page.* (n.d.). <https://tasmota.github.io/docs/>.
- What is amqp.* (n.d.). <https://www.cloudamqp.com/blog/2019-11-21-what-is-amqp-and-why-is-it-used-in-rabbitmq.html>.
- What is rabbitmq.* (n.d.). <https://en.wikipedia.org/wiki/RabbitMQ>.
- What is sql.* (n.d.). <http://www.sqlcourse.com/intro.html>.
- Why mariadb over mysql.* (n.d.). <https://www.guru99.com/mariadb-vs-mysql.html>.