

Ατομική Διπλωματική Εργασία

Διαδικτυακή Πλατφόρμα για Έξυπνο Σπίτι

Παναγιώτης Αχιλλέως

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2017

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Διαδικτυακή Πλατφόρμα για Έξυπνο Σπίτι

Αχιλλέως Παναγιώτης

Επιβλέπων Καθηγητής

Βάσος Βασιλείου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Βάσο Βασιλείου για την υποστήριξη, την καθοδήγηση και τις συμβουλές που μου προσέφερε καθ' όλη τη διάρκεια αυτής της Διπλωματικής Εργασίας.

Περίληψη

Με την ραγδαία ανάπτυξη της τεχνολογίας σε διάφορους τομείς, έχουμε καταφέρει να δημιουργήσουμε συσκευές που να εξυπηρετούν διάφορες καθημερινές ανάγκες του ανθρώπου, καθώς επίσης και να αλληλεπιδρούμε με υπολογιστικά συστήματα (Uber Phones, laptops, tablets κ.α.) . Αυτή το άγραφο έχει στόχο να σας δείξει τα βήματα τα οποία ακολουθήσαμε για να δημιουργήσουμε μια Διαδικτυακή πλατφόρμα η οποία να αλληλεπιδρά με τις Έξυπνες συσκευές τις οποίες έχουμε στο σπίτι . Αναφέρουμε τις τεχνολογίες που χρησιμοποιούμε , τις έρευνες τις οποίες υλοποιήσαμε για να μπορέσουμε να αντεπεξέλθουμε των περιστάσεων αλλά και την αρχιτεκτονική δομή της πλατφόρμας . Γίνεται επίσης εκτεταμένη αναφορά και επεξήγηση του κώδικα του online Simulator που χρησιμοποιήσαμε για να διεξάγουμε τα πειράματα και τους ελέγχους μας.

Τέλος αναφέρονται τα συμπεράσματα που προκύπτουν από αυτή την διπλωματική εργασία όπως επίσης και μελλοντικές εργασίες οι οποίες μπορούν να γίνουν πάνω στη συγκεκριμένη Πλατφόρμα .

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή.....

1.1 Εισαγωγή

1.2 Στόχος Διπλωματικής Εργασίας

1.3 Δομή Διπλωματικής Εργασίας

Κεφάλαιο 2 Περιγραφή Προβλήματος.....

2.1 Εισαγωγή

2.2 Περιγραφή Προβλήματος

2.3 Υπάρχοντα Συστήματα-Μειονεκτήματα

2.4 Μελλοντικό Βελτιωμένο Σύστημα

2.5 Έρευνα Αγοράς

Κεφάλαιο 3 Ανάλυση Απαιτήσεων - Προδιαγραφών.....

3.1 Εισαγωγή

3.2 Ανάλυση Απαιτήσεων - Προδιαγραφών

3.3 Platform Flowchart

Κεφάλαιο 4 Απαιτούμενη Γνώση και Τεχνολογίες.....

4.1 Εισαγωγή

4.2 PHP

4.3 JavaScript

4.4 SQL

4.5 HTML/CSS

4.6 APACHE Groovy

4.7 Άλλες τεχνολογίες που μελετήθηκαν

Κεφάλαιο 5 Αρχιτεκτονική του Συστήματος.....

5.1 Εισαγωγή

5.2 Βάση Δεδομένων

5.3 Μοντελοποίηση Συστήματος

5.4 Υλοποίηση Συστήματος

ΚΕΦΑΛΑΙΟ 6 Έλεγχος – Αξιολόγηση του Συστήματος.....

6.1 Εισαγωγή

6.2 Έλεγχος

6.3 Αξιολόγηση – Αποτελέσματα

ΚΕΦΑΛΑΙΟ 7 Μελλοντική εργασία.....

7.1 Μελλοντική εργασία

Βι β λ ι ο γ ρ α φ ί α

Κεφάλαιο 1

Εισαγωγή

- ο 1.1 Εισαγωγή
- ο 1.2 Στόχος Διπλωματικής Εργασίας
- ο 1.3 Δομή Διπλωματικής Εργασίας

1.1 Εισαγωγή

Όπως έχουμε αναφέρει και στην περίληψη αυτού του έγγραφου εγώ μαζί με το επιβλέποντα καθηγητή μου Κύριο Βάσο Βασιλείου θέλαμε να δημιουργήσουμε μια διαδικτυακή πλατφόρμα η οποία να ενώνει όλες τις Έξυπνες συσκευές που έχει ο κάθε χρήστης στο σπίτι του.

1.2 Στόχος Διπλωματικής Εργασίας

Στόχος είναι η δημιουργία μιας διαδικτυακής πλατφόρμας με σκοπό

1. Την δημιουργία προσωπικού λογαριασμού για κάθε χρήστη
2. Την προσθήκη κάθε συσκευής που υπάρχει στο σπίτι
3. Την αποθήκευση χρησικών ιστοσελίδων για το χρήστη
4. Την δημιουργία συνδυασμών των συσκευών από τον χρήστη

1.3 Δομή Διπλωματικής Εργασίας

Η Δομή της διπλωματικής εργασίας είναι αντίστοιχη του συγκεκριμένου έγγραφου.

Αρχικά καθορίσαμε τους στόχους τους οποίους θέλαμε να επιτύχουμε μέχρι το τέλος του δεύτερου εξαμήνου.

Σαν δεύτερο κομμάτι μελετήσαμε αρκετά από τα προϋπάρχοντα συστήματα και καταλήξαμε στο τι πρέπει να κάνει το δικό μας μελλοντικό σύστημα για να βελτιώσει τις αδυναμίες και προσθέσαμε λειτουργίες που δεν υπάρχουν ακόμα αλλά τις θεωρήσαμε απαραίτητες.

Στο τρίτο μέρος διεξαγάγαμε μια ανάλυση απαιτήσεων-προδιαγραφών και προσπαθήσαμε να κάνουμε τα πρώτα σχέδια πάνω στο μελλοντικό μας σύστημα.

Όπως θα δείτε στο κεφάλαιο 4 το επόμενο βήμα ήταν να μελετήσουμε όλες τις τεχνολογίες/προγραμματιστικές γλώσσες που χρειαστήκαμε .

Ακολουθώντας αρχίσαμε το σχεδιασμό του συστήματος και τελικά καταλήξαμε στην υλοποίηση του .

Τέλος διεξαγάγαμε δοκιμές, πειράματα με τη βοήθεια κάποιου online simulator για να δούμε ότι το σύστημα μας δουλεύει όπως το είχαμε σχεδιάσει και ότι όντως ανταποκρίνεται στις ανάγκες των χρηστών που είχαμε θέσει αρχικά σαν στόχο.

Κεφάλαιο 2

Περιγραφή Προβλήματος

- ο 2.1 Εισαγωγή
- ο 2.2 Περιγραφή Προβλήματος
- ο 2.3 Υπάρχοντα Συστήματα -Μειονεκτήματα
- ο 2.4 Μελλοντικό Βελτιωμένο σύστημα
- ο 2.5 Έρευνα Αγοράς

2.1 Εισαγωγή

Για να είμαστε σίγουροι για το τι μελετάμε, χρειάστηκε να κάνουμε μια εκτεταμένη έρευνα αγοράς πάνω στα προϊόντα Έξυπνου σπιτιού ανά το παγκόσμιο. Σε περίπτωση που μπορούσε το πανεπιστήμιο να μας παρέχει χρήματα για να αγοράσουμε κάποιες συσκευές, θα θέλαμε να αγοράσουμε τις καλύτερες σε ποιότητα . Αυτό μας βοήθησε στο να καθορίσουμε τα προϋπάρχον προβλήματα στα τρέχον συστήματα και ακολούθως να θέσουμε τους δικούς μας στόχους έτσι ώστε να τα λύσουμε στην δική μας πλατφόρμα .

2.2 Περιγραφή Προβλήματος - 2.3 Υπάρχοντα Συστήματα -Μειονεκτήματα

Οι περισσότερες εταιρίες που πωλούν έξυπνες συσκευές για το σπίτι παρέχουν τις δίκες τους ιστοσελίδες και εφαρμογές για τη λειτουργία τους αλλά στις περισσότερες περιπτώσεις απαιτείται η αγορά HUB της συγκεκριμένης μάρκας το οποίο είναι συμβατό μόνο με συγκεκριμένες μάρκες και μοντέλα συσκευών. Άρα ο πελάτης περιορίζεται στο να αγοράζει μόνο τα προϊόντα τους ή των συνεργατών τους .

Επίσης για τον συνδυασμό των συσκευών υπάρχουν αρκετά συστήματα ,ένα από τα καλύτερα είναι το IFTTT(IF THIS THEN THAT) ,το οποίο προσωπικά πιστεύω ότι είναι σχετικά δύσκολο στο χειρισμό από ένα απλό χρήστη ο οποίος δεν έχει καθόλου επαφή με προγραμματισμό.

2.4 Μελλοντικό Βελτιωμένο σύστημα

Με το νέο σύστημα:

Ο χρήστης - ιδιοκτήτης μπαίνει στο σύστημα και δημιουργώντας ένα λογαριασμό μπορεί να εισάγει τις δίκες του συσκευές και να τις ελέγχει χρησιμοποιώντας μόνο τις λειτουργίες της πλατφόρμας.

Ο χρήστης μπορεί να αποθηκεύσει όσες ιστοσελίδες χρειάζεται για την ερευνά αγοράς συσκευών, οδηγίες χρήσης ή οτιδήποτε άλλο.

Ο χρήστης μπορεί να συνδυάσει εύκολα τις συσκευές του με τη χρήση των σεναρίων (scripts) μέσα σε ένα εύκολο στο χειρισμό και όμορφα διαμορφωμένο περιβάλλον .

2.5 Έρευνα Αγοράς και κάποια από τα αποτελέσματα της

Google home

- Γενικές πληροφορίες 1)Κόστος 129 Δολάρια
2)Κυκλοφορεί από τις 4 του Νοέμβρη 2016
3)Αποτελείτε από smart speakerμε ενσωματωμένο AI βοηθό
- Σε βασικές γραμμές είναι σχεδόν το ίδιο με το amazonEcho (κυκλοφόρησε στην Αμερική πριν 2 χρόνια ,και άρχισε να πωλείτε στο Ηνωμένο Βασίλειο Σεπτέμβρη του 16)
- Βασικές λειτουργίες
 - 1) Smarthome hub: Παρέχει πρόσβαση σε οτιδήποτε Smarthome συσκευές υπάρχουν εγκατεστημένες στο σπίτι
 - 2) Αναπαραγωγή μουσικής και βίντεο μέσω τηλεόρασης η εξωτερικώνspeakers οπουδήποτε στο σπίτι
 - 3) AskGoogle: Εδώ εμφανίζετε η τεχνολογία Τεχνικής Νοημοσύνης αφού θα κάνεις ερωτήσεις και θα παίρνεις απαντήσεις από το Google
- Όταν κυκλοφορήσουν θα είναι συμβατά με :
 - 1) Philips Hue (Professional wireless lighting)

Διάφορα προϊόντα φωτισμού τα οποία καθοδηγούνε από εξωτερικές εφαρμογές

2) Nest ,έχουν προς το παρόν 4 προϊόντα (κάμερες εξωτερικού και εσωτερικού χώρου, θερμοστάτη και ανιχνευτή καπνού)

3) Smart things

4) IFTTT(IF this, then that)

5) GoogleChromecast (Παρέχουν υπηρεσίες αναπαραγωγής τηλεοπτικών προγραμμάτων και μουσικής)

Για μουσική :Spotify, Pandora and TuneIn

Αναπαραγωγή Βίντεο: Netflix and YouTube.

Smart Things (Samsung)

- Γενικές πληροφορίες

Ο Τρόπος που λειτουργούν τα SmartThings είναι ο εξής :

To SmartThings application

SmartThings Hub

Οι Συσκευές σου



- Βασικές λειτουργίες

Προσφέρονται διάφορα προϊόντα τα οποία ανήκουν σε 3 κατηγορίες :

1)Φωτισμός και ενέργεια

2)Έλεγχος και Ασφάλεια

3)Ευκολία και Ψυχαγωγία

- Σύνδεσμος για εγγραφή για πρόσβαση στην πλατφόρμα υλοποίησης

<http://developer.smarthings.com/>

- Κάποια από τα Προϊόντα

- 1) SmartThings Hub. Το πιο βασικό και αναγκαίο για τις υπηρεσίες του smart things αφού είναι το κέντρο επικοινωνίας με τις διάφορες συσκευές με το cloud της Samsung το οποίο επικοινωνεί και αυτό με τη σειρά του με το app το οποίο χειρίζονται οι χρήστες .
- 2) Samsung SmartThings Multipurpose Sensor :Μπορεί να τοποθετηθεί παντού(παράθυρα, πόρτες, ντουλάπια κτλ.). Ελέγχει αν είναι ανοιχτό η κλειστό (ανάλογα που έχει τοποθετηθεί), επίσης ελέγχει θερμοκρασία και δόνηση. Μπορεί να ενεργοποιήσει ανάλογα με την κατάσταση άλλες συσκευές όπως για παράδειγμα να ανάψει ή να σβήσει τα φώτα.
- 3) Samsung SmartThings Motion Sensor: Ελέγχει για οποιαδήποτε κίνηση στην τοποθεσία στην οποία έχει τοποθετηθεί όπως και τη θερμοκρασία. Μπορεί να ενεργοποιήσει ανάλογα με την κατάσταση άλλες συσκευές όπως για παράδειγμα να ανάψει ή να σβήσει τα φώτα.
- 4) Samsung SmartThings Water Leak Sensor: Ελέγχει την υγρασία έτσι ώστε σε περιπτώσεις, για παράδειγμα, που σπάσει κάποιος σωλήνας θα σας ενημερώσει πριν γίνουν εκτεταμένες ζημιές. Ελέγχει επίσης τη θερμοκρασία .

HomeKit (Apple)

- Γενικές πληροφορίες
 - Το homekit είναι η εφαρμογή της apple η οποία ελέγχει και μπορεί να ενώνει τις λειτουργίες των συσκευών που έχουμε σπίτι.
 - Λειτουργεί μόνο σε προϊόντα της apple όπως iPhone, iPad, iPod και σε iOS 8.1 ή επόμενες εκδόσεις
 - Κάθε κατασκευαστής που θέλει να παρασκευάσει συσκευή συμβατή με το HomeKit πρέπει να προσθέσει υποστήριξη για HomeKit στη συσκευή αλλά και στις εφαρμογές της και ακολούθως πρέπει να λάβει μέρος στο πρόγραμμα πιστοποίησης της apple για HomeKit συσκευές .
-
- Κάποια από τα Προϊόντα
 - 1) Elgato sensors
 - 2) Ecobee thermostat
 - 3) Ihome Smartplug
 - 4) Insteon Hub

Σύγκριση Amazon Echo με Google Home

	Google home	Amazon Echo
Price	\$130	\$180
Responds to voice commands	Yes	Yes
Always listening	Yes	Yes
Wake word	"Okay Google"	Alexa, Echo, or Amazon
Music streaming options	Google Play Music, YouTube Music, Spotify, Pandora, iHeartRadio, TuneIn, others	Amazon Prime Music, Spotify, Pandora, iHeartRadio, TuneIn, others
Smart home partnerships	Nest, SmartThings, Philips Hue, IFTTT	Nest, Ecobee, SmartThings, Wink, Insteon, Belkin WeMo, Philips Hue, Lix, Big Ass Fans, IFTTT, other devices via "skills"
Customizable appearance	Yes	No
Output to stereo system	Yes, via Chromecast	No (yes with Amazon Dot)
Synced audio playback to multiple devices	Yes, to any Google Cast device	No
Personal assistant highlights	Search Google, get a personalized daily briefing, check traffic, add items to calendar, make a shopping list, make a to do list, check flight status, track a package	Add items to calendar, make a shopping list, make a to do list, check flight status, track a package
Other features	Cast to your TV with Chromecast, launch and control Netflix and YouTube via Chromecast, send photos to your TV via Chromecast	Order a pizza, play a game, arrange an Uber pickup. Echo has an ever-growing list of 900+ skills and counting

Το Amazon Dot είναι όπως βλέπετε πιο κάτω μια πιο μικρή έκδοση του Amazon Echo.



Amazon Dot σύγκριση με τον με Amazon echo

Υπέρ

- Πρόσβαση σε διάφορες λειτουργίες της Alexa η οποίες δεν υποστηρίζονται από το echo.
- Περιλαμβάνει Audio-out port και Bluetooth speaker support
- Στα 50 δολάρια είναι 150 δολάρια πιο φτηνό από το echo

Κατά

- Η εφαρμογή της Alexa είναι κάποιες φορές ελαττωματική
- Δυο dot συσκευές στο ίδιο δωμάτιο δεν συγχρονίζονται και προσπαθούν να λύσουν ταυτόχρονα το ίδιο πρόβλημα που δίνεται στην Alexa
- Κάποιες φορές δημιουργούνται παράξενοι ήχοι από το built-in speaker
- Δεν μπορεί να χρησιμοποιήσει το Bluetooth και το usb audio output ταυτόχρονα.

Σύγκριση των hubs τα οποία κυκλοφορούν στην αγορά.

										
	Almond+	Almond2015	SmartThings	Wink	ISY994i Z	Vera Edge	Iris	Connect	BR100NX	Zee S2
Manufacturer	Securifi	Securifi	Samsung	Flextronics	Universal Devices	Mi Casa Verde	Lowes	Staples/D-Link	Nexia	HomeSeer HomeTroller
Price	\$249.99	\$80	\$99	\$80	\$209.99	\$100	\$60	\$69.99	\$90	\$190
Our Score¹	C-	D+	A-	B	B+	B	D	C	D	B+
Z-Wave	✓	–	✓	✓	✓	✓	✓	✓	✓	✓
Zigbee	✓	✓	✓	✓	–	–	✓	✓	–	–
Bluetooth	–	–	✓ (future)	✓	–	–	✓ (future)	✓	–	–
Insteon / X10	–	–	–	–	\$60 PLM	–	–	–	–	\$80 PLM
WiFi 802.11	b/g/n/ac	b/g/n	–	b/g/n	–	a/b/g/n	–	b/g/n	–	\$10
Ethernet	4x1Gb/s	2x1Gb/s	✓	–	✓	✓	✓	1x100Mb/s	1x100Mb/s	✓
USB	2x USB 3.0	1x USB 2.0	2 (future)	–	1	1	2	1	–	4
Cloud Service	Free	Free	Free	Free	Free with app purch.	Free	\$9.99/mo for Rules	Free	\$9.99/mo	Free
	Securifi Almond +	Securifi Almond2015	Samsung SmartThings	Flex Wink	Univ. Dev. ISY994iZ	VeraEdge	Lowes Iris	Staple/DLink Connect	Nexia BR100NX	Home Troller Zee S2
Local Control	✓	✓	some triggers	lights & switches	✓	✓	–	✓	–	✓
iPhone/iOS App	✓	✓	✓	✓	\$20-\$60 (MobileLinc)	✓	✓	✓	✓	✓
Android App	✓	✓	✓	✓	\$20 (MobileLinc)	✓	✓	✓	✓	✓
Windows Phone App	–	–	✓	–	\$25 (HA Pro)	✓	–	✓	✓	–
Apple Watch App	–	–	✓	✓	\$\$ addition (MobileLinc)	–	–	–	–	–
Android Wear	–	–	–	✓	–	–	–	–	–	–
IFTTT Support	–	–	✓	✓	–	–	–	–	–	✓
	Securifi Almond +	Securifi Almond2015	Samsung SmartThings	Flex Wink	Univ. Dev. ISY994iZ	MiCasa Verde	Lowes Iris	Staple/DLink Connect	Nexia BR100NX	Home Seer Zee S2
Mobile Device Alerts	✓	✓	✓	✓	✓	✓	✓	✓	✓	via IFTTT
SMS Text Alert	–	–	✓	✓	✓	✓	✓	✓	✓	✓
Email Alert	–	–	via IFTTT	via IFTTT	✓	✓	✓	–	✓	✓
Sunrise/set rule	–	–	✓	✓	✓	✓	\$\$-Premium	–	✓	✓
Amazon Echo Compatible	✓	✓	✓	✓	–	–	–	–	–	✓
Built-In Display Panel/Control	✓	✓	–	–	–	–	–	–	–	–
Battery Backup ²	–	–	✓ 10 hours	–	–	–	✓	–	✓	–
API/SDK	✓	✓	✓	✓	✓	✓	–	–	✓	✓

Κεφάλαιο 3

Ανάλυση Απαιτήσεων – Προδιαγραφών

- ο 3.1 Εισαγωγή
- ο 3.2 Ανάλυση Απαιτήσεων - Προδιαγραφών
- ο 3.3 Platform Flowchart

3.1 Εισαγωγή

Η εύρεση των απαιτήσεων και των προδιαγραφών του συστήματος έγινε μετά από την έρευνα αγοράς και την εν βάθος μελέτη των τρέχων συστημάτων . Σε αυτό το κεφάλαιο θα παρουσιάσουμε τις απαιτήσεις αυτές.

3.2 Ανάλυση Απαιτήσεων – Προδιαγραφών

Μετά από μελέτη των απαιτήσεων και προδιαγραφών του συστήματος που συμφωνήσαμε να δημιουργήσουμε καταλήξαμε στο ότι Η διαδικτυακή μας πλατφόρμα πρέπει να περιέχει τις πιο κάτω λειτουργίες.

1. Ασφαλής είσοδος στο σύστημα

- ▶ Δημιουργία Λογαριασμού στην πλατφόρμα
- ▶ Είσοδος στην πλατφόρμα με τη χρήση username και password

2. Συσκευές

- ▶ Δημιουργία συσκευής
- ▶ Επεξεργασία συσκευής
- ▶ Διαγράφη συσκευής

3. Σελιδοδείκτες

- ▶ Δημιουργία σελιδοδείκτη
- ▶ Διαγράφη σελιδοδείκτη

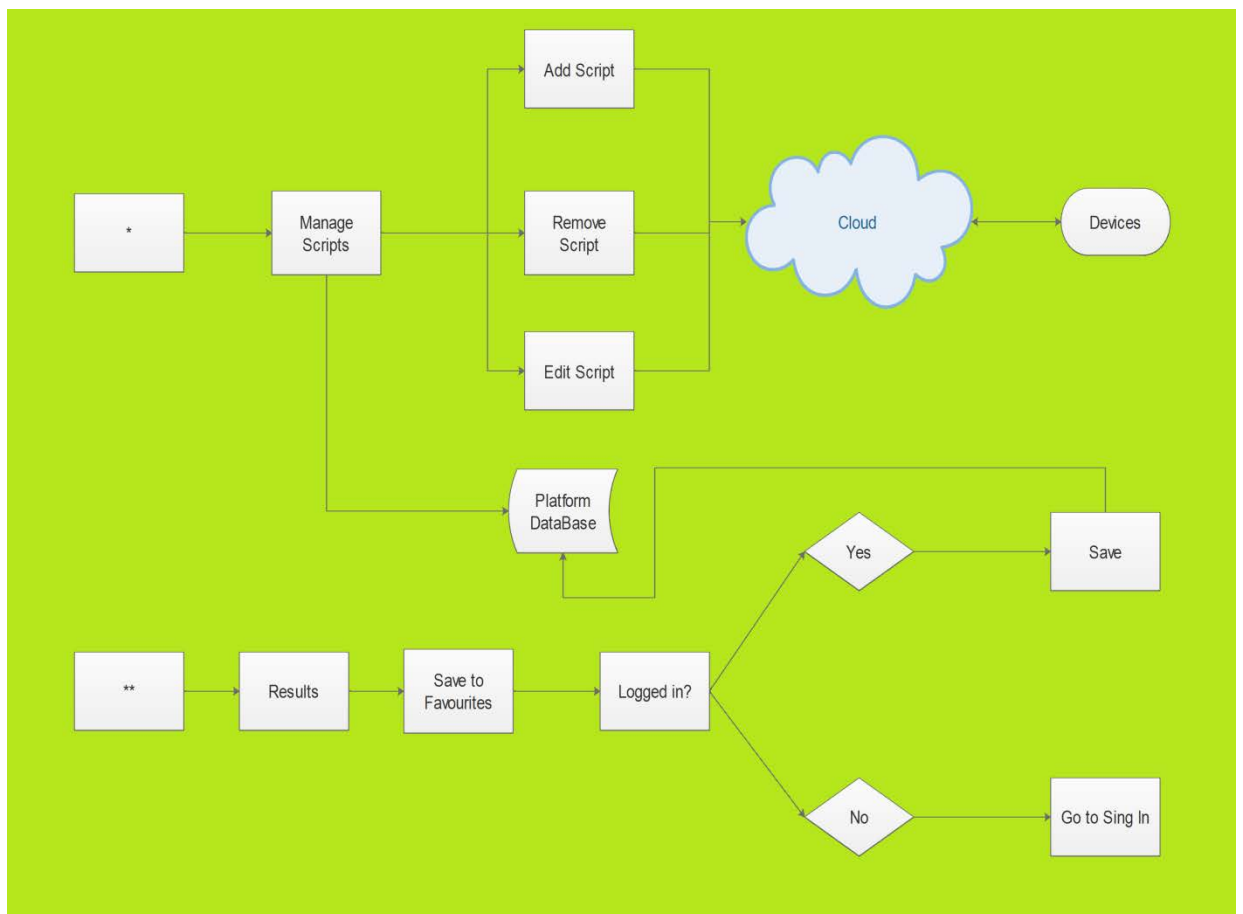
4. Σενάρια (Scripts)

- Δημιουργία Σεναρίου
- Επεξεργασία Σεναρίου
- Διαγραφή Σεναρίου

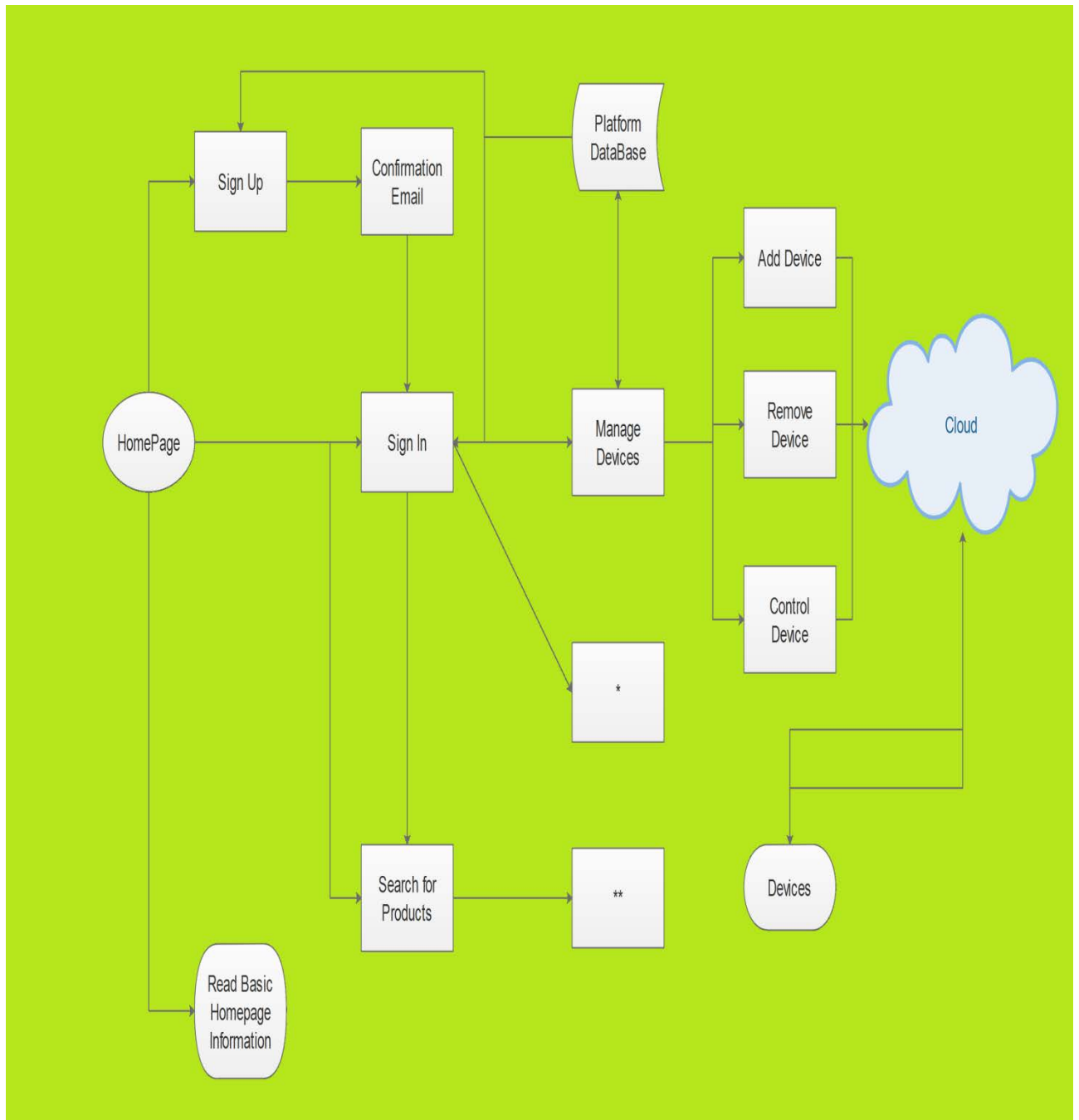
3.1 Platform Flowchart

Όπως βλέπουμε στα πιο κάτω flowcharts έχουμε παραλάβει όλες τις λειτουργίες που έχουμε αναφέρει στα προηγούμενα στάδια του κεφαλαίου.

flowchart 1



Flowchart 2



Κεφάλαιο 4

Απαιτούμενη Γνώση και Τεχνολογίες

- ο 4.1 Εισαγωγή
- ο 4.2 PHP
- ο 4.3 JavaScript
- ο 4.4 SQL
- ο 4.5 HTML/CSS
- ο 4.6 APACHE Groovy
- ο 4.7 Άλλες τεχνολογίες που μελετήθηκαν

4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα δούμε τεχνολογίες που μελέτησα και χρησιμοποίησα για την δημιουργία της πλατφόρμας. Οι τεχνολογίες αποτελούνται από γλώσσες προγραμματισμού, πρωτόκολλα διαδικτύου, άλλες ξεχωριστές πλατφόρμες οι οποίες παρέχουν παρόμοιες λειτουργίες με αυτές που θέλαμε να δημιουργήσουμε κτλ.

4.2 PHP

Η PHP είναι ακρώνυμο για την φράση **PHP: Hypertext Preprocessor**. Είναι μία γλώσσα που χρησιμοποιείται ευρέως και είναι η πιο γνωστή από τις άλλες γλώσσες παρόμοια σαν αυτή. Μπορεί ο οποιοσδήποτε να την κατεβάσει δωρεάν και να την χρησιμοποιήσει.

Είναι ανοιχτού κώδικα scripting language. Και για αυτό ένα πρόγραμμα γραμμένο σε PHP ονομάζεται smartapp. Αυτά τα scripts δεν μπορούν να τρέξουν με κάποιο εργαλείο όπως άλλες απλές γλώσσες προγραμματισμού offline αλλά πρέπει να εκτελεστούν σε server.

4.3 JavaScript

Η JavaScript είναι μια δυναμική αντικειμενοστραφής γλώσσα που χρησιμοποιείται κυρίως για ιστοσελίδες. Το συντακτικό της θυμίζει Java και C++ αλλά διαφέρει αρκετά από αυτές. Η διαφορά της με την HTML είναι ότι δημιουργεί δυναμικές ιστοσελίδες αντί στατικές. Μπορεί να γραφτεί σε HTML αρχείο αλλά και μόνη της.

4.4 SQL

Η SQL (Structured Query Language) είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων και η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα. Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα. Είναι στην ουσία η γλώσσα προγραμματισμού η οποία χειρίζεται τη Βάση Δεδομένων μας και τους πίνακες που την αποτελούν .

4.5 HTML/CSS

Μια ιστοσελίδα αποτελεί ένα έγγραφο γραμμένο σε HTML, στο οποίο περιλαμβάνονται η δομή της σελίδας, το κείμενο της και οι σύνδεσμοι προς άλλες σελίδες. Το έγγραφο αυτό ανακτάται από κάποιον εξυπηρετητή του διαδικτύου και μορφοποιείται από τον browser του κάθε χρήστη. Η μορφοποίηση του εγγράφου γίνεται με τη βοήθεια των πληροφοριών της σελίδας που ανακτά ο browser από τον εξυπηρετητή. Είναι μια γλώσσα σήμανσης υπερκειμένων που χρησιμοποιείται για τη δημιουργία δομημένων εγγράφων. Βασίζεται στην γλώσσα SGML, η οποία αποτελεί ένα μεγαλύτερο σύστημα επεξεργασίας εγγράφων. Είναι το μέσο δημιουργίας ιστοσελίδων. Επιτρέπει την ενσωμάτωση εικόνων, αντικειμένων και σεναρίων(scripts). Σε συνδυασμό με τα CSS βοηθά στον καθορισμό της εμφάνισης και της διάταξης του περιεχομένου μιας ιστοσελίδας.

4.6 APACHE GROOVY

Groovy είναι η γλώσσα προγραμματισμού η οποία χρησιμοποιήσαμε για να γράψουμε κώδικα στον online προσομοιωτή για να δοκιμάσουμε την Πλατφόρμα μας.

Apache Groovy είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού για την πλατφόρμα Java. Είναι μια δυναμική γλώσσα με χαρακτηριστικά παρόμοια με αυτά της Python, Ruby, Perl και Smalltalk. Μπορεί να χρησιμοποιηθεί ως γλώσσα δέσμης ενεργειών για την πλατφόρμα Java, να μεταγλωττιστεί δυναμικά με το bytecode Java Virtual Machine (JVM) και να διαλειτουργεί με άλλους κώδικες Java και βιβλιοθήκες. Ο Groovy χρησιμοποιεί μια συνταγή σγουρή-βραχίονα τύπου Java. Ο περισσότερος κώδικας Java είναι επίσης συντακτικά έγκυρος Groovy, αν και η σημασιολογία μπορεί να είναι διαφορετική. Το Groovy 1.0 κυκλοφόρησε στις 2 Ιανουαρίου 2007 και ο Groovy 2.0 τον Ιούλιο του 2012. Από την έκδοση 2, ο Groovy μπορεί επίσης να καταρτιστεί στατικά, προσφέροντας σύλληψη τύπου και απόδοση κοντά σε αυτή της Java. Το Groovy 2.4 ήταν η τελευταία σημαντική κυκλοφορία υπό τη χορηγία του Pivotal Software που έληξε τον Μάρτιο του 2015. [3] Από τότε ο Groovy άλλαξε τη δομή διακυβέρνησής του σε επιτροπή διαχείρισης έργων (PMC) στο Apache Software Foundation [4].



4.7 Άλλες τεχνολογίες που μελετήθηκαν

Z-Wave

Μια ιδιόκτητη προδιαγραφή ασύρματης επικοινωνίας που αναπτύχθηκε το 2004 για ασφαλή επικοινωνία συσκευών χαμηλής ισχύος και χαμηλής καθυστέρησης. Το πρωτόκολλο Z-Wave χρησιμοποιεί ένα δίκτυο πλέγματος που υποστηρίζει έως και 4 hops, επιτρέποντας στις συσκευές να αναμεταδίδουν την επικοινωνία μεταξύ τους και να επεκτείνουν την εμβέλεια του δικτύου, ελαχιστοποιώντας ταυτόχρονα την κατανάλωση ενέργειας. Τα δίκτυα Z-Wave περιορίζονται σε 232 συσκευές κατ'ανώτατο όριο.

Οι συσκευές Z-Wave λειτουργούν γύρω από την περιοχή συχνοτήτων 900 MHz, η οποία θεωρητικά επιτρέπει μεγαλύτερο εύρος λειτουργίας για μια δεδομένη κατανάλωση ισχύος με ανεμπόδιστους αισθητήρες σε σύγκριση με πρωτόκολλα υψηλότερων συχνοτήτων όπως το Zigbee. Στην πράξη, όμως, οι συσκευές Zigbee μπορούν να έχουν μεγαλύτερη εμβέλεια σε περιβάλλοντα με υψηλά εμπόδια στο σπίτι. Οι συσκευές Z-Wave γενικά έχουν λιγότερα ζητήματα συμβατότητας μεταξύ διαφόρων controllers, εξαιτίας πολλών παραγόντων, συμπεριλαμβανομένων και των περιορισμών αδειοδότησης της Sigma Design.

Τον Φεβρουάριο του 2016, η UL ενέκρινε το τελευταίο πρωτόκολλο Z-Wave mesh για τη συμμόρφωση με το σύστημα συναγερμού από συστήματα UL 1023. Τα εγκεκριμένα προϊόντα UL 1023 δεν αναμένεται να κυκλοφορήσουν μέχρι το φθινόπωρο του 2016, ωστόσο τα υπάρχοντα προϊόντα Z-Wave που χρησιμοποιούν το νεότερο chipset της σειράς 500 της σειράς Sigma ενδέχεται να είναι αναβαθμισμένα με firmware-upgradeable για συμβατότητα με UL.

Zigbee

Ένα ανοικτό πρότυπο προδιαγραφών ασύρματης επικοινωνίας, το οποίο καθιερώθηκε ως IEEE 802.15.4 το 2003 για ασφαλή επικοινωνία χαμηλής ισχύος και χαμηλής καθυστέρησης. Όπως συμβαίνει με το Z-Wave, το πρωτόκολλο Zigbee χρησιμοποιεί ένα δίκτυο mesh και υποστηρίζει ένα υψηλότερο όριο hop από το Z-Wave, το οποίο μπορεί να επεκτείνει σημαντικά το εύρος του δικτύου. Τα δίκτυα Zigbee μπορούν να υποστηρίξουν έναν πολύ μεγαλύτερο αριθμό συσκευών από το Z-Wave, με βιομηχανικά συστήματα που μερικές φορές έχουν χιλιάδες συνδεδεμένες συσκευές.

Οι συσκευές Zigbee λειτουργούν στην περιοχή συχνοτήτων 2,4 GHz και θεωρητικά έχουν μικρότερο εύρος λειτουργίας για μια δεδομένη κατανάλωση ενέργειας με ανεμπόδιστους αισθητήρες σε σύγκριση με το Z-Wave. Στην πράξη, ωστόσο, οι συσκευές Zigbee μπορούν να έχουν μεγαλύτερη εμβέλεια σε περιβάλλοντα που περιορίζονται σε μεγάλο βαθμό.

Δεδομένου ότι τα κανάλια Zigbee και WiFi λειτουργούν και στη ζώνη των 2,4 GHz, οι δρομολογητές Home WiFi και τα συστήματα αυτοματισμού Zigbee μπορεί μερικές φορές να αλληλεπιδρούν μεταξύ τους υπό ορισμένες συνθήκες, συχνά με το δίκτυο Zigbee να αντιμετωπίζει τα προβλήματα.

Bluetooth LE

Το Bluetooth Low Energy, που ονομάζεται επίσης Bluetooth LE, BLE και Bluetooth Smart, είναι μια ιδιόκτητη προδιαγραφή ασύρματης επικοινωνίας που αναπτύχθηκε το 2006 για ασφαλή επικοινωνία συσκευών χαμηλής κατανάλωσης και χαμηλής καθυστέρησης. Σε αντίθεση με τα πρωτόκολλα Z-Wave και Zigbee, οι συσκευές χαμηλής ενέργειας Bluetooth δεν υποστηρίζουν επί του παρόντος μια τοπολογία δικτύου πλέγματος. Ωστόσο, οι συσκευές Bluetooth extender είναι διαθέσιμες και η δυνατότητα πλέγματος Bluetooth βρίσκεται σε εξέλιξη. Το Bluetooth Low Energy λειτουργεί σε περιοχή συχνοτήτων 2,4 GHz και λόγω της πολύ χαμηλής κατανάλωσης ενέργειας έχει γενικά μια λειτουργική περιοχή μικρότερη από την Zigbee και Z-Wave. Το Bluetooth LE είναι ασύμβατο με το τυπικό Bluetooth, αν και υπάρχουν διαθέσιμα ολοκληρωμένα κυκλώματα που υποστηρίζουν και τα δύο πρότυπα.

Insteon

Μια ειδική προδιαγραφή επικοινωνίας που αναπτύχθηκε το 2005 το οποίο υποστηρίζει τόσο ασφαλή ασύρματη επικοινωνία όσο και επικοινωνία μέσω γραμμών μεταφοράς ηλεκτρικής ενέργειας. Όπως και με τα Z-Wave και Zigbee, το Insteon χρησιμοποιεί ένα mesh network, επιτρέποντας στις συσκευές να αναμεταδίδουν την επικοινωνία μεταξύ τους και να επεκτείνουν σημαντικά το εύρος του δικτύου. Όπως και με το Z-Wave, οι συσκευές Insteon λειτουργούν γύρω από την περιοχή συχνοτήτων 900 MHz. Το Insteon είναι επίσης συμβατό με το X10.

X10

Ένα παλαιότερο πρωτόκολλο επικοινωνίας συσκευών που αναπτύχθηκε στη δεκαετία του 1970 και υποστηρίζει την επικοινωνία μέσω γραμμών μεταφοράς ηλεκτρικής ενέργειας καθώς και ασύρματης επικοινωνίας. Οι συσκευές X10 γενικά κοστίζουν λιγότερο από τους σύγχρονους ομολόγους τους. Το πρωτόκολλο X10 είναι πιο αργό, λιγότερο ασφαλές και θεωρείται γενικά λιγότερο αξιόπιστο από τα νεότερα πρωτόκολλα συσκευών.

Local Control

Πολλοί ελεγκτές αυτοματισμού στο σπίτι απαιτούν σύνδεση στο διαδίκτυο και υπηρεσία cloud για λειτουργία. Ένας ελεγκτής που υποστηρίζει το Τοπικό Έλεγχο επιτρέπει την εκτέλεση των κανόνων αισθητήρων και τον χειροκίνητο έλεγχο των αισθητήρων σας, ακόμη και αν το cloud ή η υπηρεσία Διαδικτύου δεν είναι διαθέσιμη. Το τοπικό περιβάλλον χρήστη ελέγχου μπορεί να υλοποιηθεί με ένα τοπικό πρόγραμμα περιήγησης ιστού, μια εφαρμογή για φορητές συσκευές ή μια επιφάνεια εργασίας ή μια συσκευή panel UI.

IFTTT

Μια ισχυρή δυνατότητα όταν συνδυάζεται με έναν ελεγκτή αυτοματισμού σπιτιού που σας επιτρέπει να ενσωματώσετε τον αυξανόμενο αριθμό συσκευών και συστημάτων που έχουν δυνατότητα IFTTT χρησιμοποιώντας απλές δηλώσεις "If This Then That". Για παράδειγμα, εάν ο ελεγκτής σας υποστηρίζει IFTTT, μπορείτε να ενσωματώσετε μια υπηρεσία καιρού IFTTT για να προσθέσετε κανόνες (αναφερόμενες ως "recipes") που απενεργοποιούν το πότισμα κήπων εάν βρέχει. Παρόλο που σας συνιστούμε ιδιαίτερα να χρησιμοποιείτε ελεγκτές που είναι επίσης ικανοί να λειτουργούν χωρίς σύνδεση στο Διαδίκτυο, οι δυνατότητες IFTTT που βασίζονται στο cloud μπορεί να είναι σε θέση να βελτιώσουν σημαντικά το σύστημά σας για μη κρίσιμες λειτουργίες που δεν σχετίζονται με την ασφάλεια.

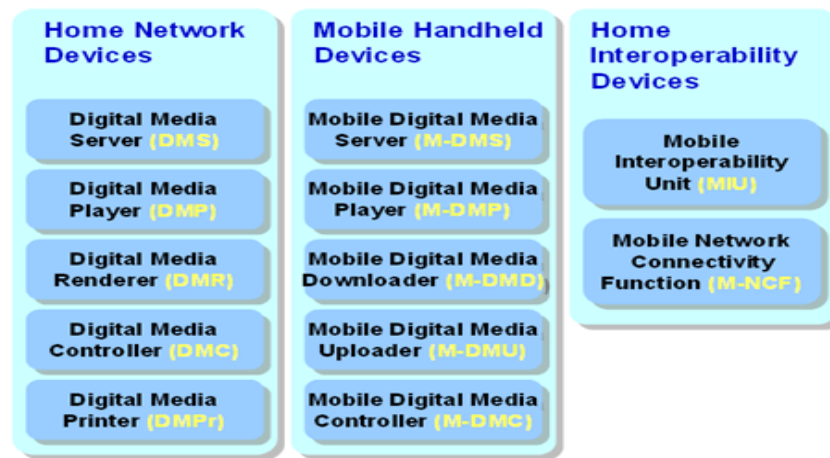
DLNA (Digital Living Network Alliance)

Το DLNA είναι ένα πρότυπο που χρησιμοποιείται από κατασκευαστές ηλεκτρονικών ειδών ευρείας κατανάλωσης, κινητών συσκευών και pc για να επιτρέπουν συσκευές ψυχαγωγίας μέσα στο σπίτι να μοιράζονται το περιεχόμενο μεταξύ τους σε ένα οικιακό δίκτυο.

Το μοντέλο των συσκευών που χρησιμοποιείται από το DLNA προέρχεται από το UPnP Forum fundamental device model. Το UPnP model περιλαμβάνεται από :

1. Οι συσκευές είναι οντότητες δικτύου που παρέχουν υπηρεσίες. Ένα παράδειγμα θα ήταν μια συσκευή παραγωγής Video(VCR).
2. Οι υπηρεσίες είναι η βασική μονάδα ελέγχου. Παρέχουν δράσεις και διατηρούν την κατάσταση. Ένα παράδειγμα θα ήταν η συσκευή VCR που παρέχει υπηρεσία μεταφοράς ταινιών, υπηρεσία συντονισμού και υπηρεσία χρονισμού.
3. Τα σημεία ελέγχου είναι οντότητες δικτύου που είναι σε θέση να ανακαλύψουν και να ελέγξουν άλλες συσκευές στο δίκτυο.

Για να καθορίσουν καλύτερα τα χαρακτηριστικά των συσκευών και των υπηρεσιών που προσφέρουν, οι Οδηγίες Διαλειτουργικότητας DLNA ορίζουν δώδεκα Κατηγορίες Συσκευών που οργανώνονται σε τρεις Κατηγορίες Συσκευών. Οι ακόλουθες κατηγορίες είναι οι εξής:



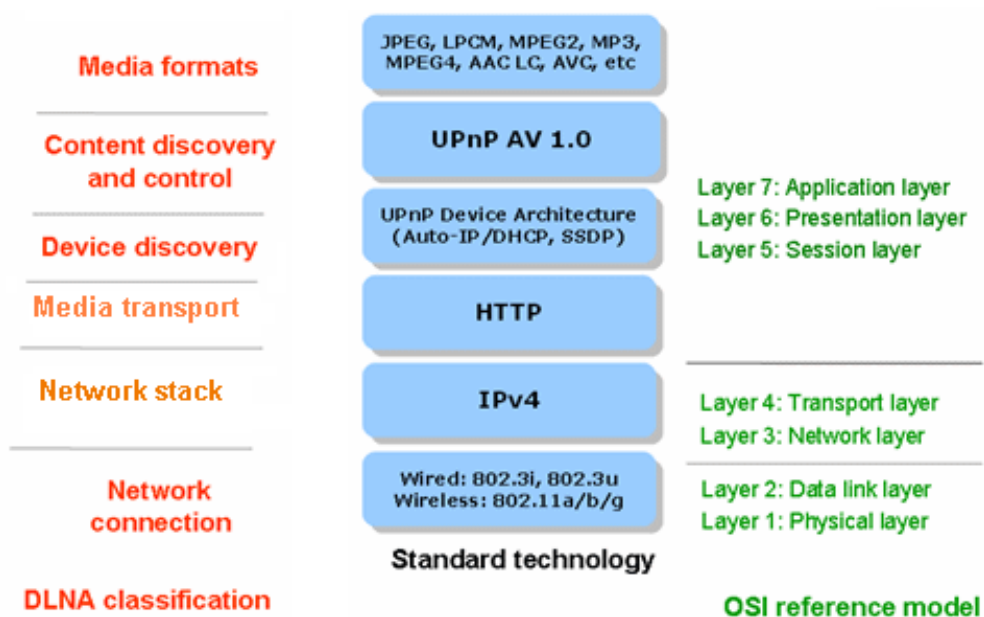
Media Formats που χρησιμοποιούνται από τις συσκευές

Images: JPGE ----> Optional: PNG, GIF, TIFF

Audio: LPCM ----> Optional: AC3, AAC, MP3, WMA9, ATRAC3plus

Video: MPEG2 ----> Optional: MPEG1, MPEG4, VC1, MPV1

Στο παρακάτω σχήμα παρατηρούμε την αρχιτεκτονική δομή του DLNA



Media Transport ορίζει πως τα περιεχόμενα κινούνται στο δίκτυο και είναι αναγκαστικά HTTP1.1

Σαν δεύτερη επιλογή μπορεί να χρησιμοποιηθεί και το RTP (The Real-time Transport Protocol) το οποίο είναι ένα πρωτόκολλο δικτύου το που μεταφέρει αρχεία ήχου και βίντεο πάνω σε IP δίκτυα.

Network Stack

Η βάση του UPnP Networking είναι το πρωτόκολλο TCP/IP v4.

Το UPnP κάνει εκτεταμένη χρήση των UDP και TCP πρωτοκόλλων.

Η ανακάλυψη γίνεται μέσω ενός HTTP Multicast μέσω UDP.

Οι υπηρεσίες καθορισμού, ελέγχου και συμβάντων παρέχονται μέσω HTTP πάνω στο TCP. Κάθε συσκευή πρέπει να εφαρμόσει ένα πρόγραμμα-πελάτη DHCP και να αναζητήσει έναν διακομιστή DHCP κατά την πρώτη σύνδεση με το δίκτυο. Εάν εντοπιστεί ένας διακομιστής DHCP, η συσκευή πρέπει να χρησιμοποιήσει τη διεύθυνση IP που έχει εκχωρήσει ο διακομιστής. Εάν δεν εντοπιστεί διακομιστής DHCP, η συσκευή πρέπει να χρησιμοποιήσει το Auto-IP για να δημιουργήσει μια διεύθυνση IP τοπικής σύνδεσης.

Rest (Representational State Transfer)

Το Representational State Transfer (REST) είναι ένα "αρχιτεκτονικό στυλ" που ορίστηκε από τον Roy Fielding. Οι έννοιες του REST είναι ανεξάρτητες από τον ιστό, αλλά ο ιστός είναι κατάλληλος για το REST.

- Απλουστευμένα, μια "Restful αρχιτεκτονική" περιλαμβάνει: Resources (πράγματα) με μοναδικά αναγνωριστικά (URLs) που μπορούν να έρθουν σε πολλές μορφές (text, html, json) με τις οποίες εκτελούμε λειτουργίες με τη χρήση των Verbs (GET, PUT, DELETE, POST)

- Τα ποιοι κοινά verbs που χρησιμοποιούνται είναι:

GET:

Περνώ μια αναπαράσταση του πόρου χωρίς να τον αλλάξω

PUT:

Δημιουργώ ή αντικαθιστώ ένα πόρο παρέχοντας μια αναπαράσταση του

DELETE:

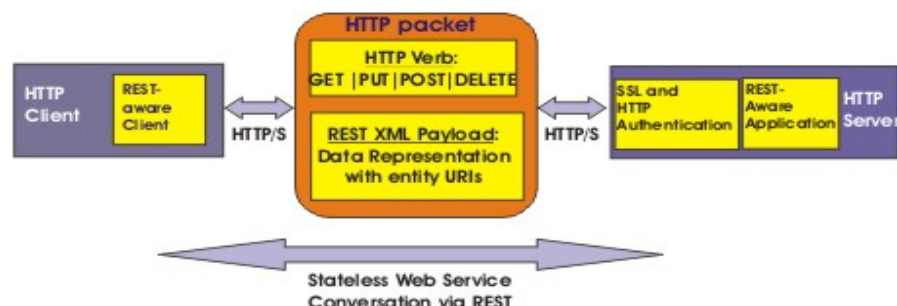
Διαγράφω ένα πόρο

POST:

Παρέχω ένα πόρο με επιπρόσθετες πληροφορίες δίνοντας μια αναπαράσταση αυτού και των πληροφοριών.

- Αναπαραστάσεις (πόρων)
Ως εξωτερικός χρήστης, δεν μπορείτε να χειρίζεστε τους πόρους άμεσα. Μπορείτε μόνο να χειρίζεστε τις αναπαραστάσεις αυτού του πόρου. Πολλοί άνθρωποι μπορούν να "πάρουν" μια αναπαράσταση ενός μόνο πόρου. Ο ίδιος πόρος μπορεί να εκπροσωπείται με πολλούς διαφορετικούς τρόπους. Το πρωτεύον πρωτόκολλο του Internet το (HTTP) είναι φτιαγμένο κατάλληλα για Restful αρχιτεκτονικές.

Στην ποιό κάτω εικόνα βλέπουμε τη χρήση του rest που κάνει το HTTP.

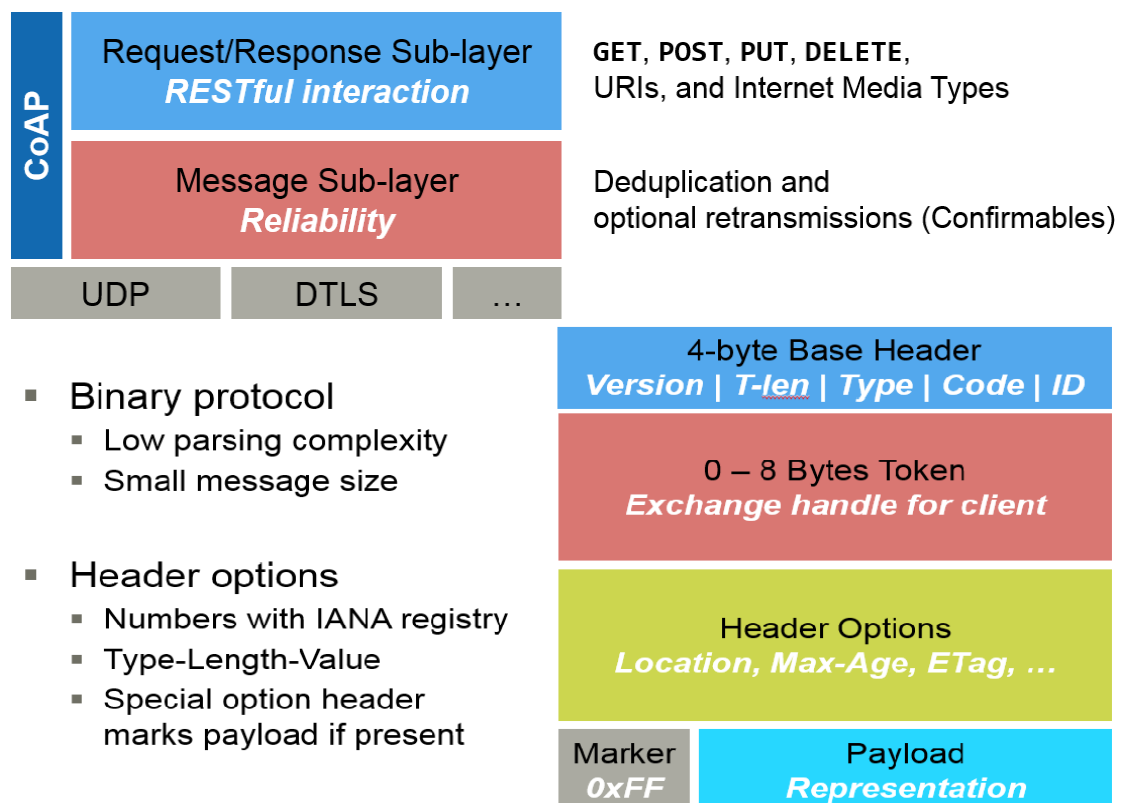


Constrained Application Protocol (CoAP)

Το Constrained Application Protocol (CoAP) είναι ένα εξειδικευμένο πρωτόκολλο μεταφοράς ιστού για χρήση με περιορισμένους κόμβους και περιορισμένα δίκτυα στο "Internet of Things".

Το πρωτόκολλο έχει σχεδιαστεί για εφαρμογές από μηχανή σε μηχανή (M2M), όπως το Smart energy και τους αυτοματισμούς κτιρίων

- Είναι RESTful πρωτόκολλο σχεδιασμένο από το μηδέν
- Transparent mapping to HTTP
- Επιπρόσθετες λειτουργίες για σενάρια M2M
- Το CoAP μπορεί να μεταφέρει διαφορετικούς τύπους φορτίων και μπορεί να προσδιορίσει τον τύπο του φορτίου που χρησιμοποιείται. Το CoAP ενσωματώνεται με XML, JSON, CBOR ή οποιαδήποτε μορφή δεδομένων της επιλογής σας.



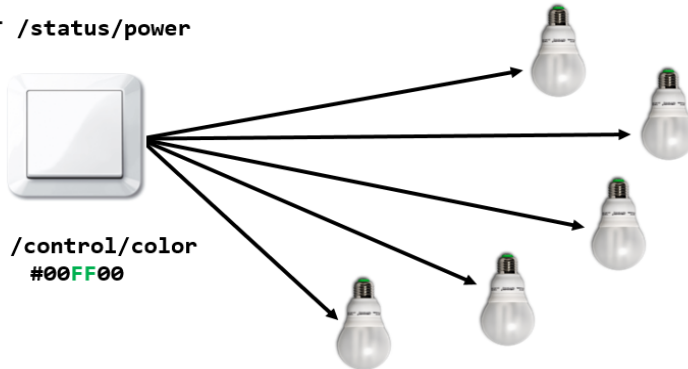
RESTful Group Communication

Enabled by IP multicast

all-lights.floor-d.example.com

GET /status/power

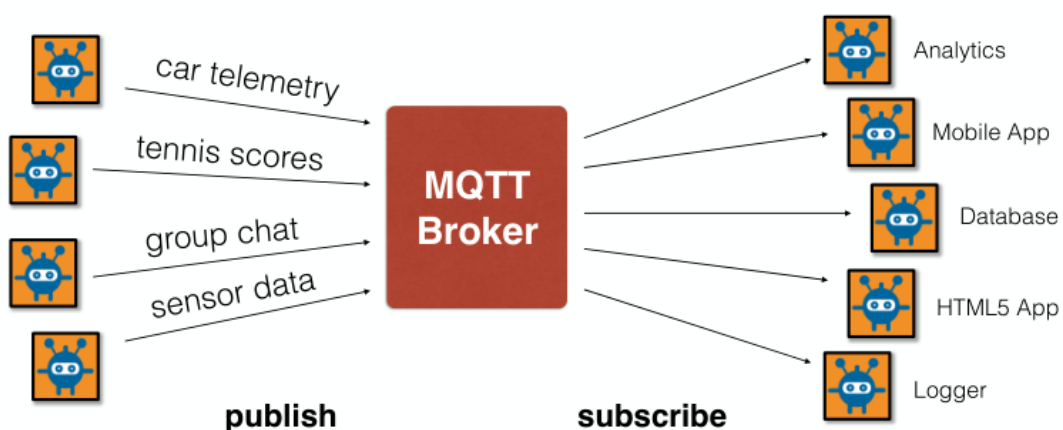
PUT /control/color
#00FF00



MQTT (MQ Telemetry Transport or Message Queuing Telemetry Transport)

- Το MQTT είναι ένα "ελαφρύ" πρωτόκολλο ανταλλαγής μηνυμάτων για δημοσίευση-εγγραφή, είναι σχεδιασμένο να χρησιμοποιείται πάνω από το πρωτόκολλο TCP / IP. Έχει σχεδιαστεί για συνδέσεις με απομακρυσμένες τοποθεσίες όπου απαιτείται "small code footprint" ή περιορισμένο εύρος ζώνης δικτύου. Το μοντέλο δημοσίευσης-εγγραφής μηνυμάτων απαιτεί έναν μεσίτη μηνυμάτων. Ο μεσίτης είναι υπεύθυνος για τη διανομή μηνυμάτων σε ενδιαφερόμενους πελάτες με βάση το θέμα ενός μηνύματος.

MQTT pub/sub decouples **senders** from **receivers**



Το MQTT ορίζει τις μεθόδους (verbs) για να υποδείξει την επιθυμητή ενέργεια που πρέπει να εκτελεστεί στον προσδιορισμένο πόρο. Αυτό που αντιπροσωπεύει αυτός ο πόρος, είτε τα προϋπάρχοντα δεδομένα ή τα δεδομένα που παράγονται δυναμικά, εξαρτώνται από την υλοποίηση του διακομιστή. Συχνά, ο πόρος αντιστοιχεί σε ένα αρχείο ή την έξοδο ενός εκτελέσιμου αρχείου που βρίσκεται στο διακομιστή.

Κάποιες από τις μεθόδους(Verbs)

Connect

Περιμένει να δημιουργηθεί μια σύνδεση με το διακομιστή.

Disconnect

Περιμένει το πρόγραμμα-πελάτη MQTT να ολοκληρώσει οποιαδήποτε εργασία πρέπει να κάνει και να αποσυνδεθεί η περίοδος σύνδεσης TCP / IP.

Subscribe

Περιμένει να ολοκληρωθεί η μέθοδος Subscribe ή Unsubscribe.

Unsubscribe

Ζητά από τον διακομιστή να καταργήσει την εγγραφή του πελάτη(unsubscribe) από ένα ή περισσότερα θέματα.

Publish

Επιστρέφει αμέσως στο νήμα εφαρμογής αφού περάσει το αίτημα στον πελάτη MQTT.

MQTT

simple to implement

Connect

Subscribe

Publish

Unsubscribe

Disconnect

```
client = new Messaging.Client(hostname, port, clientId)
client.onMessageArrived = messageArrived;
client.onConnectionLost = connectionLost;
client.connect({ onSuccess: connectionSuccess });

function connectionSuccess() {
    client.subscribe("planets/earth");
    var msg = new Messaging.Message("Hello world!");
    msg.destinationName = "planets/earth";
    client.publish(msg);
}

function messageArrived(msg) {
    console.log(msg.payloadString);
    client.unsubscribe("planets/earth");
    client.disconnect();
}
```

Κεφάλαιο 5

Αρχιτεκτονική του Συστήματος

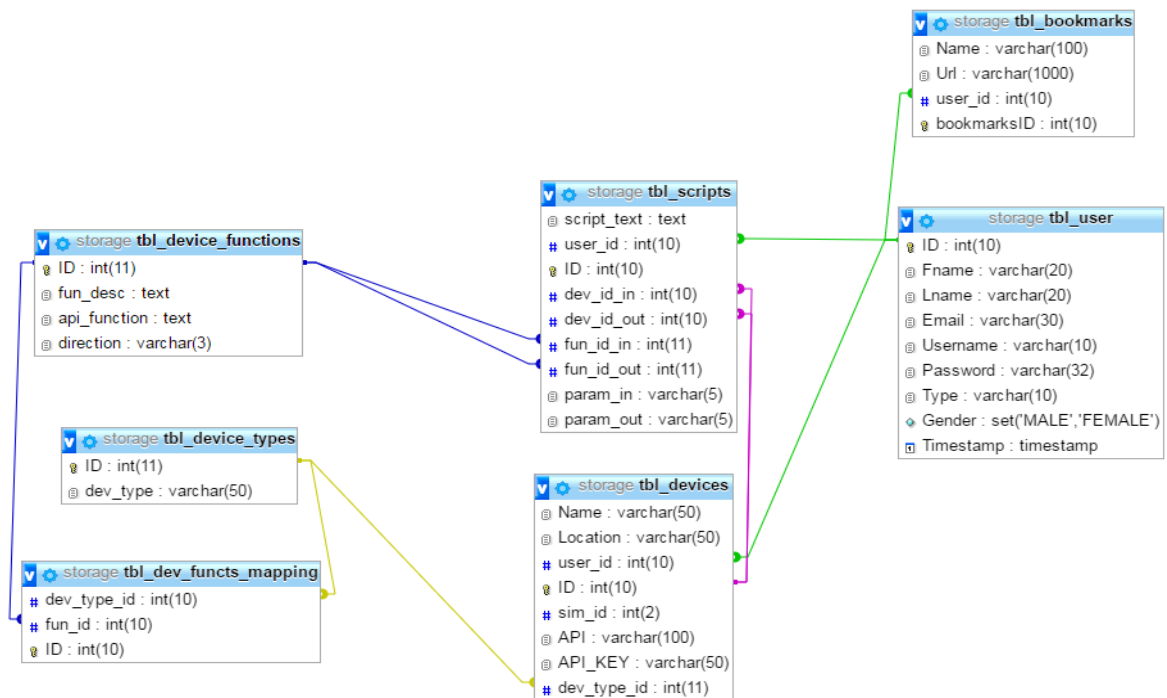
- ο 5.1 Εισαγωγή
- ο 5.2 Βάση Δεδομένων
- ο 5.3 Μοντελοποίηση Συστήματος
- ο 5.4 Υλοποίηση Συστήματος

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα δούμε τα διάφορα κομμάτια του συστήματος και πως λειτουργούν .

5.2 Βάση δεδομένων

Η Βάση Δεδομένων της πλατφόρμας μας είναι ένα από τα πιο σημαντικά κομμάτια . Μέσα στη Βάση αποθηκεύονται όλες οι πληροφορίες του χρήστη όπως τα στοιχεία του ,οι συσκευές του ,οι σελιδοδείκτες κτλ . Επίσης αποθηκεύουμε τα διάφορα CALLS τα οποία γίνονται σε κάθε συσκευή πάνω στον Προσομοιωτή. Στο πιο κάτω σχεδιάγραμμα βλέπουμε όλους τους πίνακες της Βάσης και όλα τα πρωτεύον και δευτερεύον κλειδιά που έχουν.



Πινάκες και τα περιεχόμενα τους

- **tbl_user**

Έχει πρωτεύον κλειδί το ID και περιέχει όλα τα στοιχεία του χρήστη όπως Όνομα, Επώνυμο, Φύλο κτλ.

- **tbl_bookmarks**

Έχει πρωτεύον κλειδί το bookmarkID και περιέχει όλους τους σελιδοδείκτες που αποθηκεύει ο κάθε χρήστης . Έχει σαν δευτερεύον κλειδί το ID του χρήστη για να ξέρουμε σε ποιοι χρήστη ανήκει ο κάθε σελιδοδείκτης.

- **tbl_scripts**

Έχει πρωτεύον κλειδί το ID το οποίο είναι ξεχωριστό για κάθε σενάριο . Έχει σαν δευτερεύον κλειδιά το ID του χρήστη για να ξέρουμε σε ποιο χρήστη ανήκει το κάθε σενάριο, το ID των συσκευών για να ξέρουμε τις συσκευές τις οποίες χρησιμοποιεί ένα σενάριο και το ID του tbl_device_functions για ξέρουμε τι functions μπορεί να εκτελέσει κάθε συσκευή .

- **tbl_devices**

Έχει πρωτεύον κλειδί το ID το οποίο είναι ξεχωριστό για κάθε device . Έχει σαν δευτερεύον κλειδιά το ID του χρήστη για να ξέρουμε σε ποιο χρήστη ανήκει η κάθε συσκευή και το ID του tbl_device_type για ξέρουμε σε τι κατηγορία/τύπο ανήκει η κάθε συσκευή.

- **tbl_device_funtions**

Έχει πρωτεύον κλειδί το ID το οποίο είναι ξεχωριστό για κάθε device functions . Σε αυτό το πίνακα αποθηκεύονται ΟΛΕΣ οι λειτουργίες που μπορούν να εκτελέσουν οι συσκευές μας.

- **tbl_device_types**

Έχει πρωτεύον κλειδί το ID το οποίο είναι ξεχωριστό για κάθε device type . Σε αυτό το πίνακα αποθηκεύονται όλοι οι τύποι συσκευών που χρησιμοποιούμε στον προσομοιωτή .

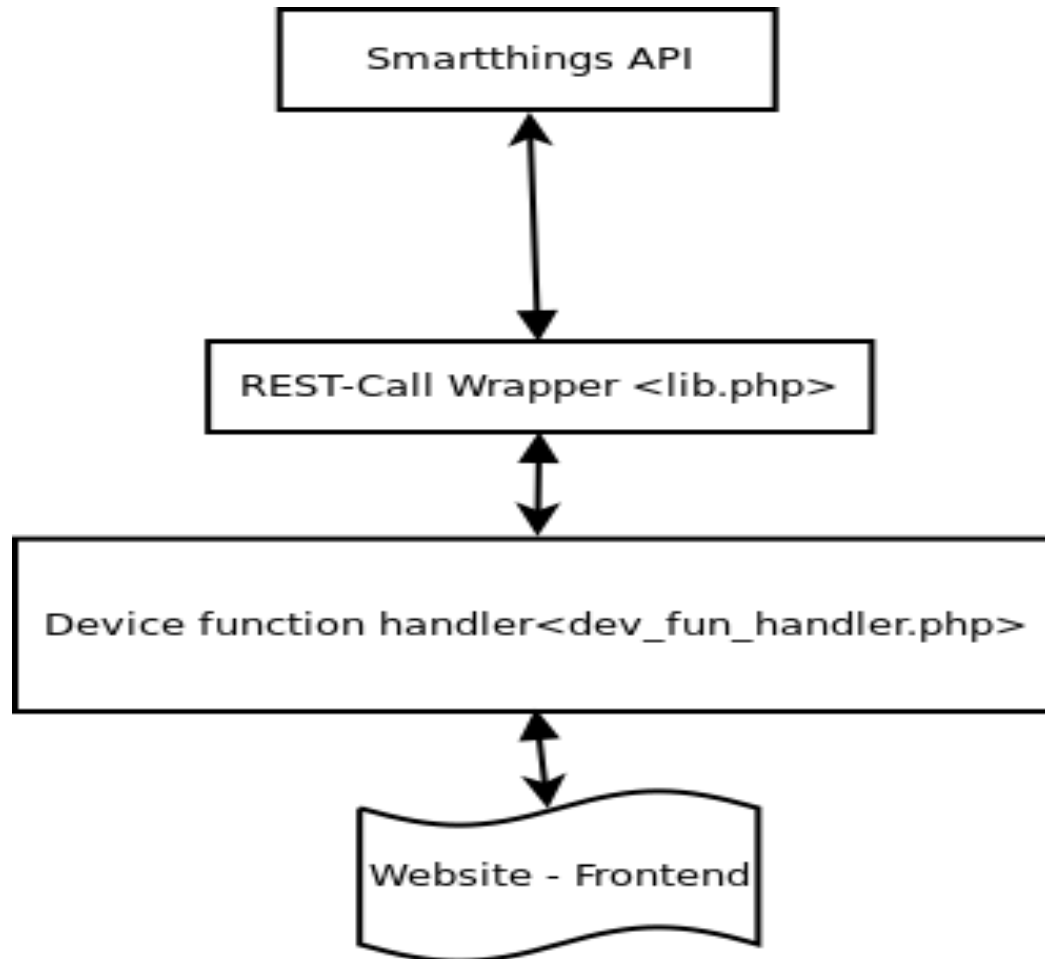
- **tbl_dev_function_mapping**

Έχει πρωτεύον κλειδί το ID το οποίο είναι ξεχωριστό για κάθε device mapping . Ο πίνακας αυτός χρησιμοποιείται για να συνδέει τις συσκευές μαζί με τις λειτουργίες τις οποίες μπορούν να εκτελέσουν .

5.3 Μοντελοποίηση του Συστήματος

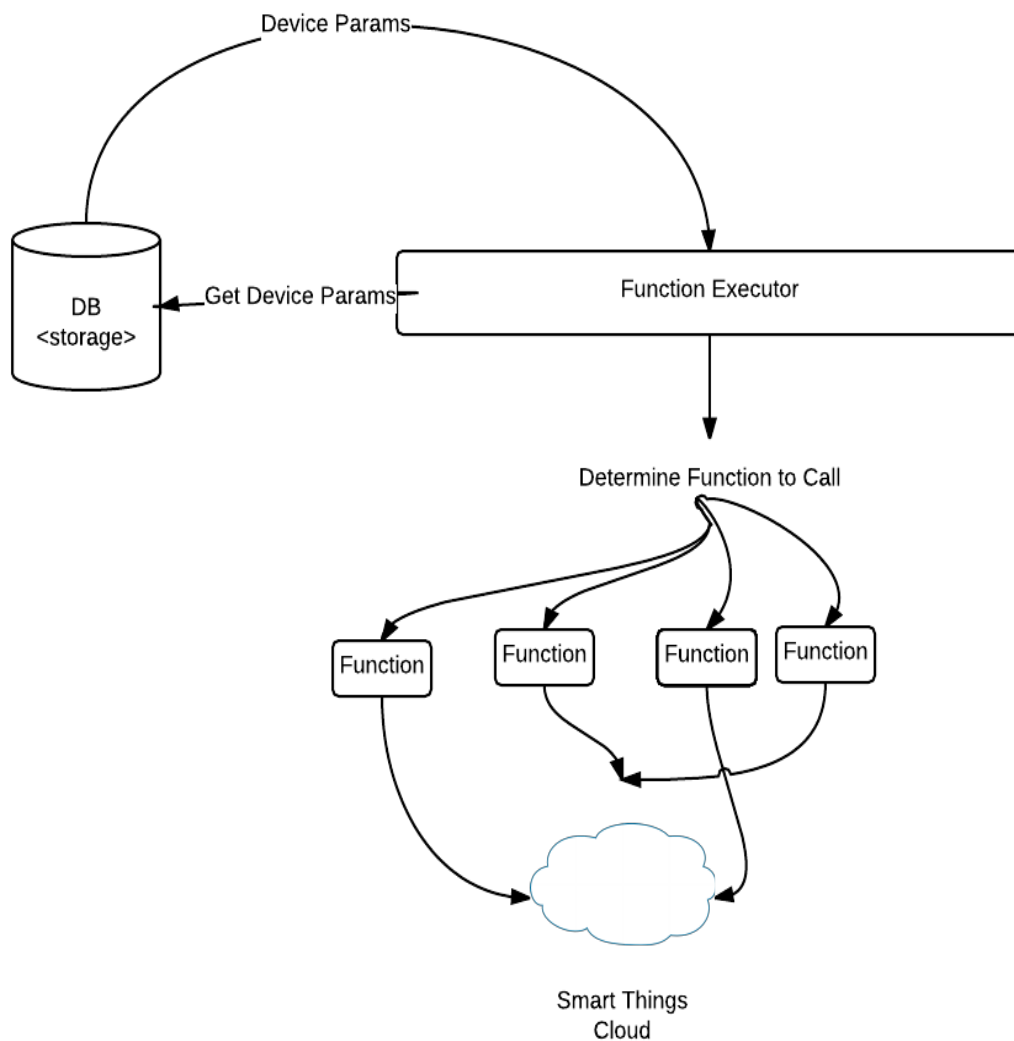
Με τη χρήση των πιο κάτω μοντέλων θα εξηγήσουμε πως λειτουργεί το σύστημα μας σε αρχιτεκτονικό επίπεδο .

Μοντέλο 1



- Η πλατφόρμα χρησιμοποιείται σαν διαπροσωπία των συσκευών.
- Εντολές από την ιστοσελίδα καλούν λειτουργίες από την κεντρική μονάδα επεξεργασίας των συσκευών.
- Την επαφή με το SmarThings Api εκτελεί η μονάδα Rest-call Wrapper (library)

Μοντέλο 2



- Ο διαχειριστής συσκευών προσομοιώνει τις διάφορες λειτουργίες των συσκευών .
- Τα διάφορα στοιχεία που είναι απαραίτητα για την εκτέλεση προέρχονται από τη βάση δεδομένων.

- **Ανάλογα με το τι λειτουργία έχει επιλέγει και με τον τύπο της συσκευή συμπεραίνεται από το διαχειριστή ποιιά λειτουργιά θα καλεστεί από το SmartThings Api**

5.3 Υλοποίηση Συστήματος

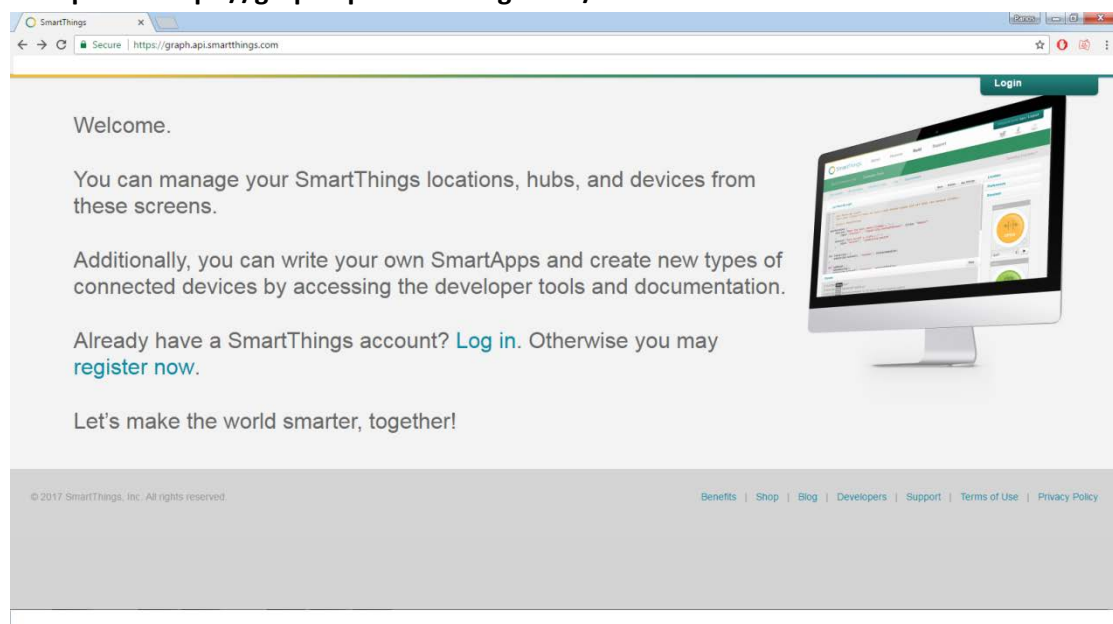
Η υλοποίηση του συστήματος διαμερίστηκε σε τρία βασικά κομμάτια :

1. Στη δημιουργία της ιστοσελίδας
2. Στη δημιουργία της Βάσης δεδομένων
3. Στη δημιουργία του Προσομοιωτή

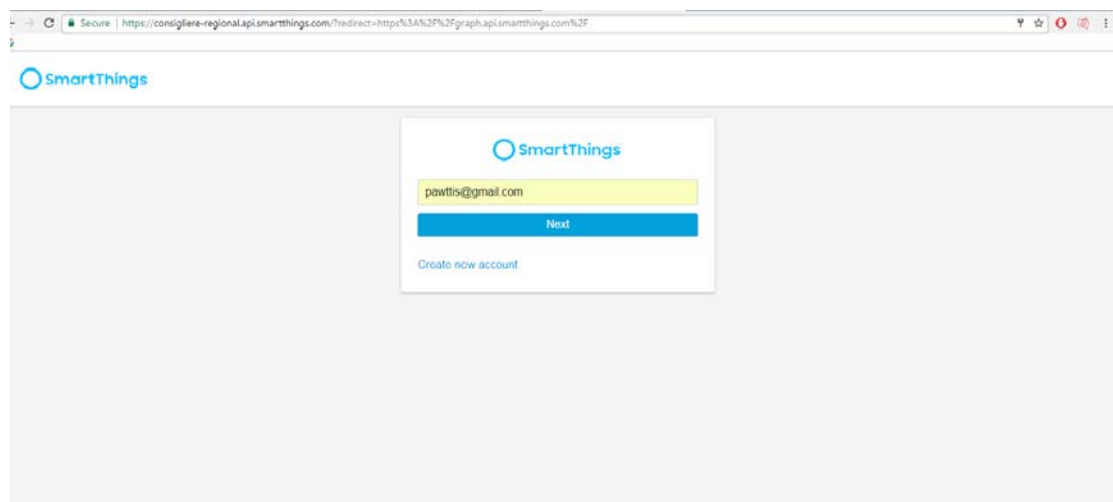
Για τα πρώτα δυο έχουμε πει αρκετά, για τον προσομοιωτή ο οποίος ήταν και ένα από τα πιο απαιτητικά κομμάτια, θα μιλήσουμε στο επόμενο κομμάτι αυτού του έγγραφου.

Παρουσίαση του site το οποίο χρησιμοποιούμε για τη δημιουργία του Simulator.

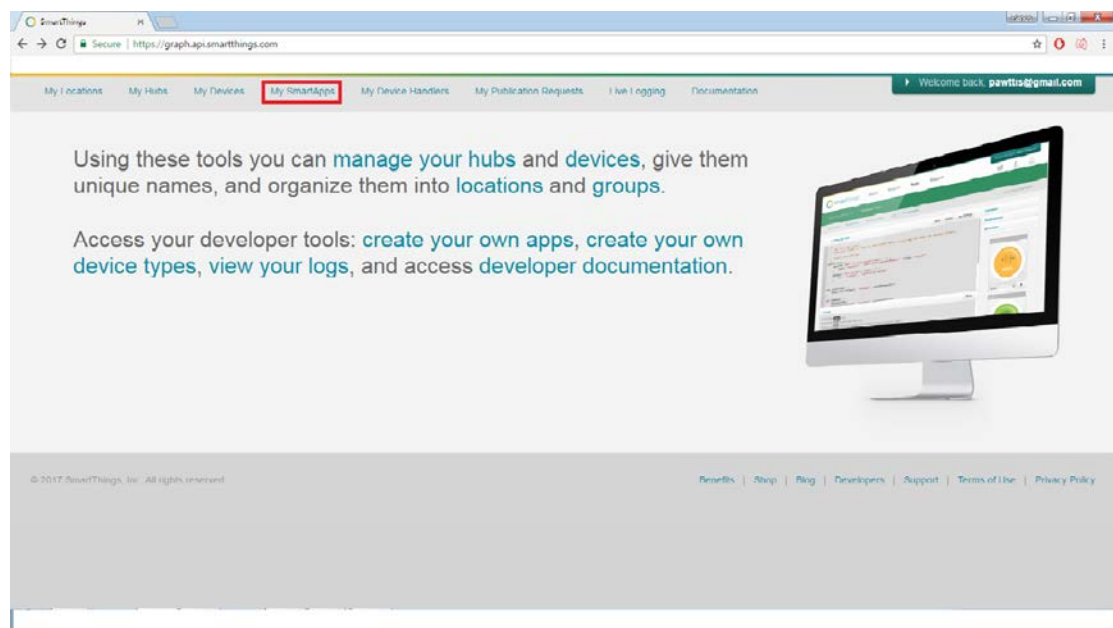
1. Πάμε στο <https://graph.api.smarthings.com/>



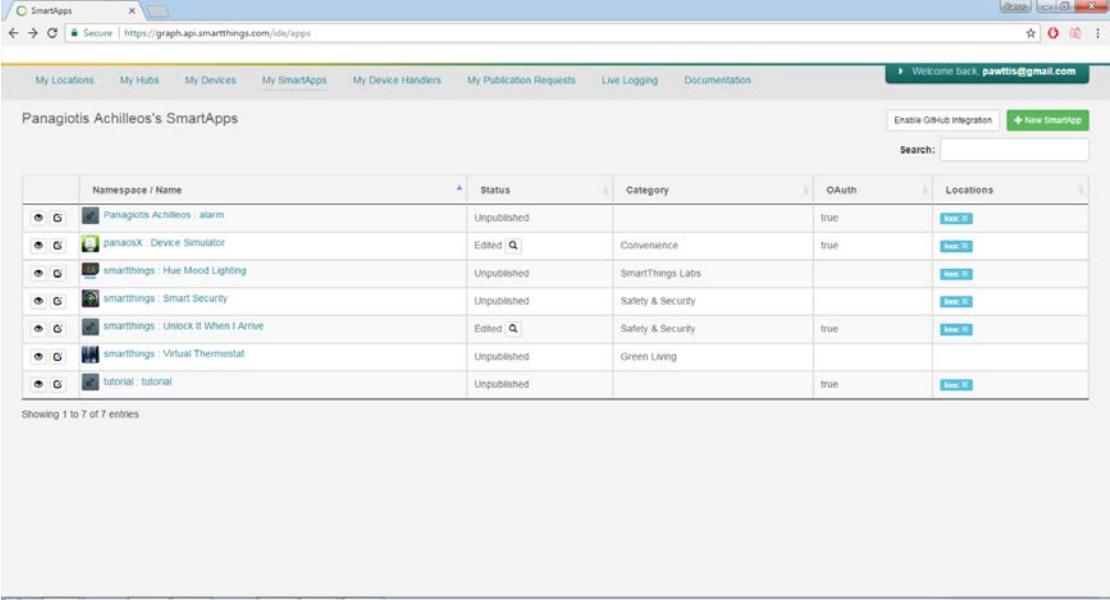
2.Κάνουμε Login στο site με το δικό μας λογαριασμό.



3.Οταν κάνουμε Login βλέπουμε το παρακάτω μενού. Επιλέγουμε το My Smart Apps



4. Στο My Smart Apps βλέπουμε όλες τις εφαρμογές που έχουμε δημιουργήσει στο λογαριασμό μας. Το Simulator που χρησιμοποιούμε είναι ένα από αυτές τις Έξυπνες Εφαρμογές. Εδώ μπορούμε να δημιουργήσουμε καινούργιες ή να διαγράψουμε κάποια από τις προϋπάρχουσες εφαρμογές μας.



The screenshot shows the 'SmartApps' interface in a web browser. The page title is 'Panagiotis Achilleos's SmartApps'. The navigation bar includes links for 'My Locations', 'My Hubs', 'My Devices', 'My SmartApps', 'My Device Handlers', 'My Publication Requests', 'Live Logging', and 'Documentation'. A user profile bar at the top right shows 'Welcome back, pawtts@gmail.com' and a '+ New SmartApp' button. Below the navigation bar, there is a search bar and a table of SmartApps. The table has columns for 'Namespace / Name', 'Status', 'Category', 'OAuth', and 'Locations'. The table lists seven SmartApps, including 'Panagiotis Achilleos : alarm', 'panaosX : Device Simulator', 'smarthings : Hue Mood Lighting', 'smarthings : Smart Security', 'smarthings : Unlock It When I Arrive', 'smarthings : Virtual Thermostat', and 'tutorial : tutorial'. Each row has a 'See It' button in the 'Locations' column. At the bottom of the table, it says 'Showing 1 to 7 of 7 entries'.

Namespace / Name	Status	Category	OAuth	Locations
Panagiotis Achilleos : alarm	Unpublished		true	See It
panaosX : Device Simulator	Edited Q	Convenience	true	See It
smarthings : Hue Mood Lighting	Unpublished	SmartThings Labs		See It
smarthings : Smart Security	Unpublished	Safety & Security		See It
smarthings : Unlock It When I Arrive	Edited Q	Safety & Security	true	See It
smarthings : Virtual Thermostat	Unpublished	Green Living		
tutorial : tutorial	Unpublished		true	See It

Showing 1 to 7 of 7 entries

Παρουσίαση και επεξήγηση Κώδικα Api Simulator

- Ο προσομοιωτής που χρησιμοποιούμε δηλώνετε μέσω ενός SmartApp που βλέπουμε στον πιο κάτω κώδικα , στο παρακάτω απόσπασμα κώδικα δηλώνουμε τις διάφορες ιδιότητες.

```
definition(  
  
    name: "Device Simulator",  
  
    namespace: "panaosX",  
  
    author: "Panagiotis Achilleos",  
  
    description: "serves as a wrapper for different functionalities to simulate devices",  
  
    category: "Convenience",  
  
    imageUrl: "https://s3.amazonaws.com/smartapp-icons/Meta/light_outlet.png",  
  
    iconX2Url: "https://s3.amazonaws.com/smartapp-  
icons/Meta/light_outlet@2x.png"  
  
)
```

- Οι συσκευές δηλώνονται για χρήση στον προσομοιωτή όπως βλέπουμε στο ποιοι παρακάτω κομμάτι . (Λίστα όλων υποστηριζόμενων συσκευών στον προσομοιωτή <http://docs.smarthings.com/en/latest/capabilities-reference.html>)

```
preferences {  
  
    section("Enable switches?") {  
  
        input "switches", "capability.switch", multiple: true  
  
    }  
  
    section("Do You want a thermostat?"){  
  
        input "thermostat", "capability.thermostat", multiple: false  
  
    }  
  
}
```



```

    section("Enable humidity sensors?"){
        input "humidity", "capability.relativeHumidityMeasurement", multiple: false
    }
    section("Enable alarm system?"){
        input "alarm", "capability.alarm", multiple: false
    }
    section("Enable colorful lamps?"){
        //input "colorLight", "capability.colorControl", multiple: true
    }
    section("Select the heater or air conditioner outlet(s)... "){
        input "outlets", "capability.switch", title: "Outlets", multiple: true
    }
    section("Set the desired temperature..."){
        input "setpoint", "decimal", title: "Set Temp"
    }
    section("Choose a temperature sensor... "){
        input "sensor", "capability.temperatureMeasurement", title:
"Sensor"
    }
    section("When I arrive..."){
        input "presence1", "capability.presenceSensor", title: "Who?",
multiple: true
    }
    section("Unlock the lock..."){
        input "lock1", "capability.lock", multiple: true
    }
}

```

- Στο επόμενο κομμάτι δηλώνονται οι διάφορες λειτουργίες των συσκευές σαν RESTFULL API calls. Στο πιο κάτω ισχύει ότι όταν έχουμε GET ζητούμε πληροφορίες από τη συσκευή και όταν έχουμε POST αλλάζουμε την κατάσταση της συσκευής.

```
mappings {  
    path("/alarm"){  
        action: [  
            GET: "getAlarmState"  
        ]  
    }  
  
    path("/switches") {  
        action: [  
            GET: "listSwitches"  
        ]  
    }  
  
    path("/switches/toggle") {  
        action: [  
            POST: "updateSwitches"  
        ]  
    }  
  
    path("/switches/toggle/:switchID"){  
        action: [  
            POST: "updateSwitches"  
        ]  
    }  
}
```

```
path("/thermostat/"){
  action: [
    GET: "getThermostatTemp"
  ]
}

path("/thermostat/toggle"){
  action: [
    POST: "toggleThermostat"
  ]
}

path("/thermostat/:desiredTemp"){
  action: [
    POST: "setThermostatTemp"
  ]
}

path("/humidity"){
  action: [
    GET: "getHumidity"
  ]
}

path("/door/lock") {
  action: [
    GET: "absence"
  ]
}

path("/door/unlock") {
  action: [
```

```
GET: "presence"
```

```
] }}
```

- Στην ακόλουθη λειτουργία επιστρέφει το επίπεδο υγρασίας εντός του χώρου όπου έχει τοποθετηθεί σε ποσοστό της 100 %.

```
def getHumidity() {  
    def resp = []  
  
    resp << [name: humidity.displayName, value: humidity.currentLevel]  
  
    return resp  
}
```

- Ο θερμοστάτης διαθέτει δύο καταστάσεις ,η πρώτη κατάσταση είναι να ελέγχει αν το δωμάτιο έχει ψηλότερη θερμοκρασία από τη προκαθορισμένη του χρήστη ,ενώ η δεύτερη κατάσταση ελέγχει αν το δωμάτιο έχει χαμηλότερη θερμοκρασία από την προκαθορισμένη του χρήστη .

```
def setThermostatMode() {  
    log.debug('Setting Thermostat Mode To' + params.mode)  
  
    if ( params.mode.contains('heat')){  
        thermostat.heat  
  
        return [[  
            message: 'Thermostat now on heating mode'  
        ]]  
    }else{  
        thermostat.cool  
  
        return [[  
            message: 'Thermostat now in cooling mode'  
        ]]  
    }  
}
```

- Ο Θερμοστάτης έχει δύο ειδών θερμοκρασίες που επιβλέπει .Η μια είναι το κατώτατο όριο θερμοκρασίας ου πρέπει να έχει ο χώρος και το ανώτατο όριο θερμοκρασίας που πρέπει να έχει ο χώρος .Τα ακόλουθα κομμάτια κώδικα θέτουν αυτά τα όρια.

```
def setCoolingPoint() {
    log.debug("Setting Thermostat Temp")
    log.debug(thermostat.currentTemperature)
    thermostat.setCoolingSetpoint(params.desiredTemp)
    log.debug(thermostat.currentTemperature)
    return [[
        message: 'cooling point has been set'
    ]]
}

def setHeatingPoint() {
    log.debug("Setting Heating Point")
    log.debug(thermostat.currentTemperature)
    thermostat.setCoolingSetpoint(params.desiredTemp)
    log.debug(thermostat.currentTemperature)
    return [[
        message: 'heating point has been set'
    ]]
}
```

- Πιο κάτω βλέπουμε την λειτουργία που συγχρονίζει το κουμπί του προσομοιωτή με το smartApp.

```
def toggleThermostat(){
    if (thermostat.currentMode == "off"){
        thermostat.auto
    }else{
```

```

        thermostat.off
    }
}

```

- **Η ακόλουθη λειτουργία επιστρέφει την τωρινή θερμοκρασία του θερμοστάτη.**

```

def getThermostatTemp() {
    log.debug("Getting Thermostat Temp")
    log.debug(thermostat.currentTemperature)
    def resp = []
    resp << [name: thermostat.displayName, value:
thermostat.currentTemperature]
    return resp
}

```

- **Η επόμενη λειτουργία επιστέφει μια λίστα των Διακοπών .Έχει δημιουργηθεί για δοκιμαστικούς σκοπούς.**

```

def listSwitches() {
    def resp = []
    switches.each {
        resp << [name: it.displayName, value: it.currentValue("switch")]
    }
    return resp
}

```

- **Παρακάτω βλέπουμε τη λειτουργία ενεργοποίησης του συναγερμού σε περίπτωση που χρειαστεί .**

```

def soundAlarm(){
    log.debug("sounding alarm")
    alarm.both()
    resp << [message: "Alarm is beeping"]
}

```

```
}
```

- Η επόμενη λειτουργία παίρνει την είσοδο του θερμοστάτη και ανάλογα αποφασίζει αν θα ανάψει το κλιματιστικό ή τη θέρμανση.

```
private evaluate()
{
    def resp = []

    def currentTemp=thermostat.currentTemperature

    def desiredTemp=params.desiredTemp

    log.debug "EVALUATE($currentTemp, $desiredTemp)"

    def threshold = 1.0

    if (mode == "cool") {

        // air conditioner

        if (currentTemp - desiredTemp >= threshold) {

            outlets.on()

            resp << [message: "Aic-conditioner On "]

        }

        else if (desiredTemp - currentTemp >= threshold) {

            outlets.off()

            resp << [message: "Aic-conditioner Off"]

        }

    }

    else {

        // heater

        if (desiredTemp - currentTemp >= threshold) {

            outlets.on()


```

```

        resp << [message: "Heater On"]

        }

        else if (currentTemp - desiredTemp >= threshold) {

            outlets.off()

            resp << [message: "Heater Off"]

            }

        }

    }

    return resp

}

```

- **Επιστέφεται η κατάσταση του συναγερμού ,δηλαδή αν είναι ανενεργός ή αν είναι σε λειτουργία.**

```

def getAlarmState(){

    if (alarm.getCurrentValue('alarm') != 'off'){

        return [

            [

                value: "ON"

            ]

        ]

    }else{

        return [[

            value: "OFF"

        ]] }}

```

- **Η πιο κάτω λειτουργία αλλάζει την κατάσταση των διακόπτων από ενεργά σε ανενεργά και αντίστροφα ανάλογα .**

```

def toggleSwitches() {

    def resp = []

```



```
// use the built-in request object to get the command parameter

def switchID = 0

if(params.switchID != null){

    switchID = params.switchID

}

def isOn = true

log.debug(switchID<4)

if(switchID>0 && switchID<4){

    log.debug('test')

    if (switches[switchID-1].currentValue("switch") == "on") {

        resp << [message: "turning off switch " + switchID]

        switches[switchID-1].off()

    }else{

        resp << [message: "turning on switch " + switchID]

        switches[switchID-1].on()

    }

}

}else{

    switches.each{

        if(it.currentValue('switch') == "on"){

            isOn = isOn && true

        }else{

            isOn = isOn && false

        }

    }

    if(isOn){

        switches.off()

        resp<<[ message: "Switches are now turned off"]

    }

}
```

```

        }else{

switches.on()

resp<<[ message: "All switches are now on"]

        }

    }

    return resp

}

```

- Το ακόλουθο κομμάτι περιγράφει τη λειτουργία της κλειδωνιάς της πόρτας. Όπου χρησιμοποιούμε την κατάσταση "presence " ο χρήστης υποθέτουμε ότι βρίσκεται μέσα στην οικία του. Το σύστημα ανοίγει όλες τις πόρτες οι οποίες έχουν ελεγχόμενη κλειδαριά . Όπου χρησιμοποιούμε την κατάσταση " absence " ο χρήστης υποθέτουμε ότι βρίσκεται εκτός της οικίας του. Το σύστημα κλειδώνει όλες τις πόρτες οι οποίες έχουν ελεγχόμενη κλειδαριά .

```

def presence()

{

    def resp = [Status: "presence"]

    def anyLocked = lock1.count{it.currentLock == "unlocked"} != lock1.size()

    if (anyLocked) {

        log.debug("Unlocking door")

        resp<<[message: "UNLocked door "]

        lock1.unlock()

    }

    return resp

}

def absence(){

```

```

log.debug "EVALUATE()"

def resp = [Status: "absence"]

def anyUnLocked = lock1.count{it.currentLock == "locked"} != lock1.size()

if (anyUnlocked) {

    resp<<[message: "Locked door "]

    lock1.lock()

}

return resp

}

```

- **Οι ακόλουθες λειτουργίες είναι απαραίτητη για να δουλέψει ο προσομοιωτής .Η κύρια τους λειτουργία είναι η εγκατάσταση των συσκευών και αναβάθμιση της διεπαφής του προσομοιωτή όταν ο προγραμματιστής ενημερώνει τον κώδικα του smartapp. Επίσης δηλώνει ποιες λειτουργίες αποτελούν χείριστες γεγονότων .**

```

def installed()

{

    subscribe(location, changedLocationMode)

    subscribe(app, appTouch)

    subscribe(presence1, "presence.present", presence)

}

def updated()

{

    unsubscribe()

    subscribe(location, changedLocationMode)

    subscribe(app, appTouch)

    subscribe(presence1, "presence.present", presence)

}

```

```
def changedLocationMode(evt) {  
    log.debug "changedLocationMode: $evt"  
    switches?.on()  
}
```

```
def appTouch(evt) {  
    log.debug "appTouch: $evt"  
    switches?.on()  
}
```

Κεφάλαιο 6

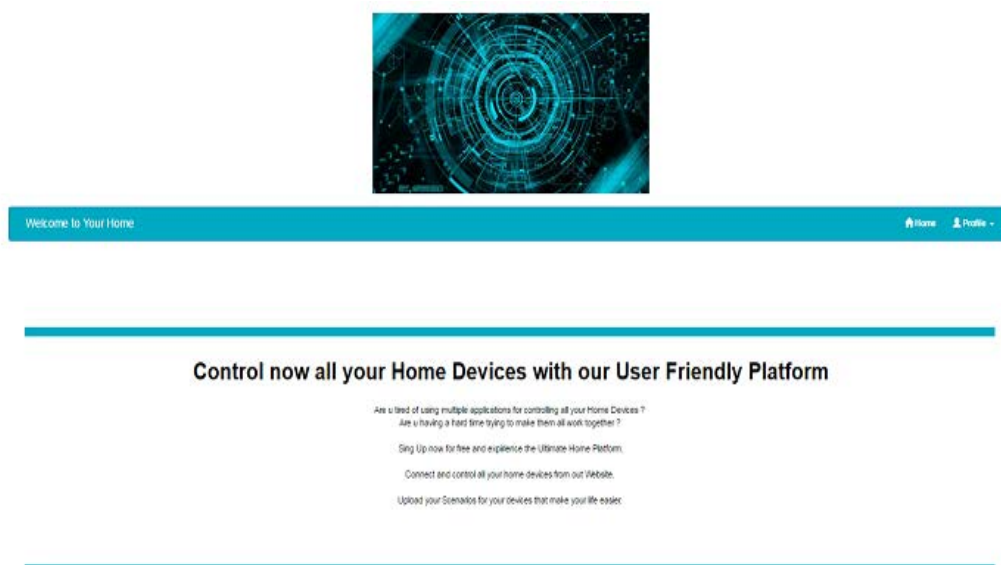
Έλεγχος – Αξιολόγηση του Συστήματος

- ο 6.1 Εισαγωγή
- ο 6.2 Έλεγχος
- ο 6.3 Αξιολόγηση – Αποτελέσματα

6.1 Εισαγωγή

Ανά το παγκόσμιο τεράστια χρηματικά ποσά ξοδεύονται για την ανάπτυξη πληροφοριακών συστημάτων. Είναι επομένως σημαντικό να αξιολογηθούν τα αποτελέσματα τους. Επίσης απαραίτητη είναι η αξιολόγηση τους για την καλύτερη κατανόηση της λειτουργίας των πληροφοριακών συστημάτων.

Στην πιο κάτω εικόνα βλέπουμε το Πρωτοσέλιδο της πλατφόρμας

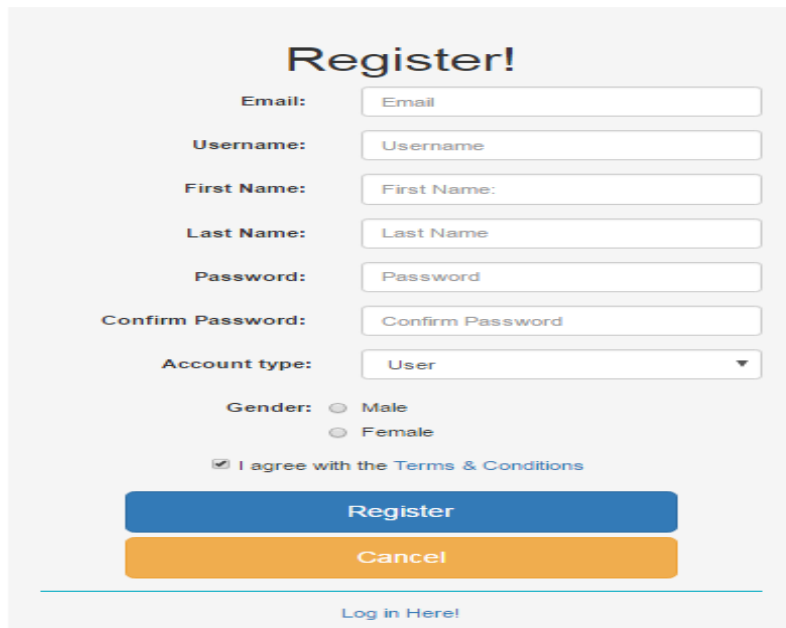


6.2 Έλεγχος

Διεξήγαμε ελέγχους διεξοδικά σε όλα τα κομμάτια του συστήματος. Στα αρχικά στάδια βρήκαμε αρκετά ελαττωματικά κομμάτια το οποία διορθώσαμε στην πορεία. Στο τέλος όμως καταλήξαμε με ένα σύστημα χωρίς λάθη και αποτυχίες.

Θα παρουσιάσουμε τον βασικό κύκλο ελέγχων που μπορεί να γίνει στο σύστημα μας.

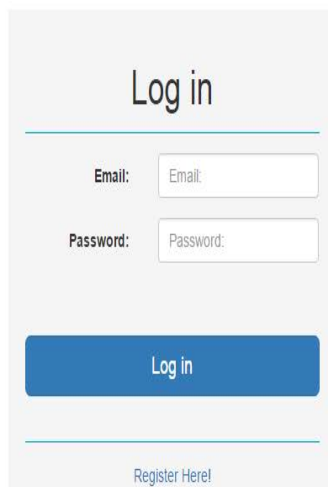
- Δημιουργία Καινούργιου Λογαριασμού-Sing Up



The image shows a registration form titled "Register!". It contains the following fields and elements:

- Email:** Input field with placeholder "Email".
- Username:** Input field with placeholder "Username".
- First Name:** Input field with placeholder "First Name".
- Last Name:** Input field with placeholder "Last Name".
- Password:** Input field with placeholder "Password".
- Confirm Password:** Input field with placeholder "Confirm Password".
- Account type:** Dropdown menu with "User" selected.
- Gender:** Radio buttons for "Male" and "Female".
- ☒ **I agree with the Terms & Conditions**
- Register** button (blue).
- Cancel** button (orange).
- [Log in Here!](#) link at the bottom.

- Σύνδεση λογαριασμού- Sing In



The image shows a login form titled "Log in". It contains the following fields and elements:

- Email:** Input field with placeholder "Email".
- Password:** Input field with placeholder "Password".
- Log in** button (blue).
- [Register Here!](#) link at the bottom.

- Δημιουργία, επεξεργασία και διαγραφή Σελιδοδείκτη

Welcome to Your Home

[Home](#) [Hello panais !](#)

My Bookmarks

google

https://madeby.google.com/home/

Delete Bookmark

amazon

https://www.amazon.com/mn/search?encoding=UTF8&field-keywords=echo

Delete Bookmark

torr

https://torrent2.eu/search?f=

Delete Bookmark

- Δημιουργία, επεξεργασία και διαγραφή Συσκευής

My Devices

Name:Thermostat Location:EsoSOu DeviceID:2	Type:thermostat	Add Device Delete Device	☒
Name:Light 1 Location:Living Room DeviceID:101	Type:switches	Add Device Delete Device	☒
Name:Humidity Sensor Location: DeviceID:102	Type:humidity	Add Device Delete Device	☒
Name:Alarm Location: DeviceID:105	Type:alarm	Add Device Delete Device	☒
Name:Door Location: DeviceID:106	Type:door	Add Device Delete Device	☒

[Add new device](#)

- Δημιουργία, επεξεργασία και διαγραφή Σεναρίου (Scripts)

Welcome to Your Home

HomeStudio panels 1

My Scripts

Script ScriptID: 40

Script :IF (Alarm IS (Valid values: ON/OFF) ON)Then Do (Light 1) Turn on

Edit Script

Play

Delete Script

Script ScriptID: 41

Script :IF (Light 1 IS (Valid values: ON/OFF) OFF)Then Do (Door) Lock

Edit Script

Play

Delete Script

Script ScriptID: 44

Script :IF (Thermostat Temperature: 62)Then Do (Thermostat) Activate AC/Heater At Temp: 80

Edit Script

Play

Delete Script

Script ScriptID: 47

Script :IF (Alarm IS (Valid values: ON/OFF) ON)Then Do (Door) Lock

Edit Script

Play

Delete Script

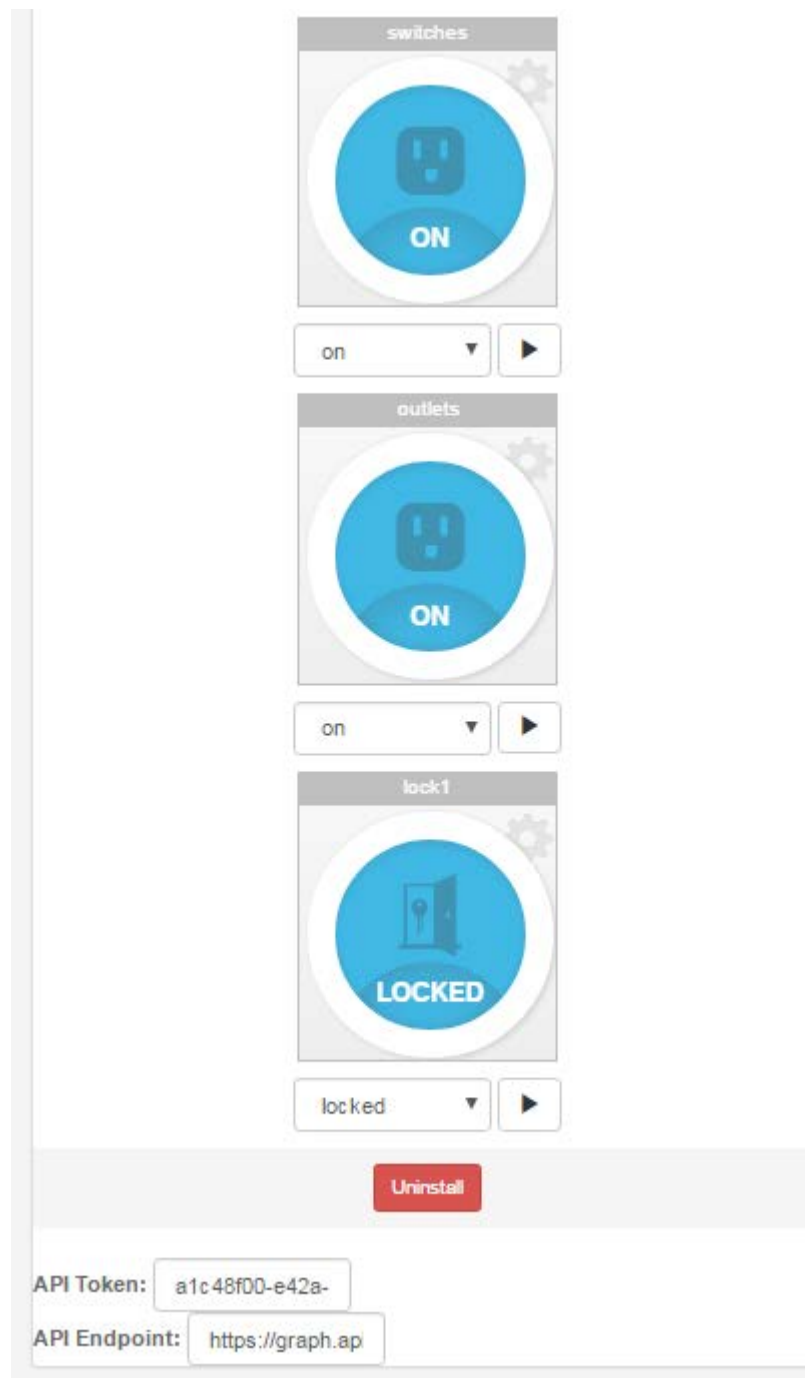
6.3 Αξιολόγηση-Αποτελέσματα

Τα αποτελέσματα της πλατφόρμας όταν την δοκιμάσαμε με τον προσομοιωτή ήταν πολύ θετικά .Δημιουργήσαμε συσκευές στην πλατφόρμα και στον προσομοιωτή και τις δουλέψαμε παράλληλα . Ο προσομοιωτής αλληλεπιδρούσε με το σύστημα μας και αντίθετα .Μπορούμε να αναβοσβήνουμε τις συσκευές αλλά και να χρησιμοποιούμε τα σενάρια για να ενεργοποιούμε και να απενεργοποιούμε τις εικονικές συσκευές .

Στην πιο κάτω εικόνα βλέπουμε 3 συσκευές και ένα δείκτη κατάστασης του προσομοιωτή, τον ανιχνευτή υγρασίας που είναι στο 60 % ,τον συναγερμό ο οποίος είναι ενεργοποιημένος ,το θερμοστάτη που δείχνει 62 βαθμούς Φαρενάιτ και τον δείκτη κατάστασης που δείχνει "present".



Στην επόμενη εικόνα βλέπουμε ακόμα 3 συσκευές, 1 switch το οποίο μας ανάβει τη λάμπα και είναι on, ένα outlet το οποίο τροφοδοτεί το κλιματιστικό το οποίο είναι επίσης on και την κλειδαριά της πόρτας που είναι κλειδωμένη .



ΚΕΦΑΛΑΙΟ 7

Μελλοντική εργασία

- ο 7.1 Μελλοντική εργασία

7.1 Μελλοντική εργασία

- Να δημιουργηθεί android application τον οποίο να χειρίζεται την πλατφόρμα και όλα της τα features πάνω σε android συσκευές
- Εισαγωγή μοντέλων αληθινών συσκευών και σύνδεση τους με την πλατφόρμα .
- Αναβάθμιση του προσομοιωτή έτσι ώστε να μπορούμε να προσθέτουμε επιπλέον συσκευές

Βιβλιογραφία

- **Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (by Robin Nixon)**
- **Groovy Recipes, Greasing the Wheels of Java (by Scott Davis)**
- **graph.api.smartthings.com**
- **<http://www.w3schools.com/>**
- **https://en.wikipedia.org/wiki/Main_Page**
- **<http://stackoverflow.com/>**
- **<http://docs.smartthings.com/en/latest/capabilities-reference.html#outlet>**