



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΚΑΤΕΥΘΥΝΣΗ: ΠΡΟΗΓΜΕΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ



**ΧΡΗΣΗ ΣΥΣΚΕΥΩΝ ΒΛΥΕΤΟΟΤΗ ΧΑΜΗΛΗΣ ΕΝΕΡΓΕΙΑΣ ΓΙΑ ΤΗΝ ΚΑΤΑΓΡΑΦΗ
ΚΑΙ ΑΝΑΛΥΣΗ ΚΑΘΗΜΕΡΙΝΩΝ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ ΤΟΥ ΧΡΗΣΤΗ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

Ροδόπουλου Διαμαντή

(Α.Μ 1625)

Επιβλέπων: Δημοσθένης Κυριαζής - Επίκουρος Καθηγητής

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος Πανεπιστημίου Πειραιώς Τμήματος Ψηφιακών Συστημάτων - Κατεύθυνση Προηγμένα Πληροφοριακά Συστήματα.

Κύριος σκοπός της εργασίας είναι η παρακολούθηση των καθημερινών δραστηριοτήτων του χρήστη στο χώρο όπου εκείνος διαμένει με τη χρήση συσκευών Bluetooth χαμηλής ενέργειας. Τέτοιες συσκευές είναι τα beacons, οι οποίες έχουν ενσωματωθεί και παρατηρούνται σε πολλούς τομείς, όπως στις μεταφορές, το εμπόριο, την υγειονομική περίθαλψη. Μεταδίδουν το σήμα τους κάθε συγκεκριμένο χρονικό διάστημα, το οποίο αναγνωρίζεται από άλλες "έξυπνες" κινητές συσκευές και έτσι μπορούμε να γνωρίζουμε ποιο beacon είναι κοντά μας. Επίσης χρησιμοποιήθηκε η συσκευή Puck της οικογένειας Espruino με ενσωματωμένους σένσορες για τη επικοινωνία με άλλες συσκευές στο δίκτυο Internet of Things και τη συλλογή δεδομένων θερμοκρασίας, φωτεινότητας όπως και κίνησης.

Η ανάπτυξη της κινητής εφαρμογής έγινε με τη χρήση του περιβάλλοντος Android Studio v3.1.4 το οποίο παρέχει όλα τα εργαλεία για την ανάπτυξη μιας τέτοια εφαρμογής και τις διεπαφές χρήστη αυτής, η πλατφόρμα Cloud Firestore της Firebase ως βάση δεδομένων. Επιπλέον, χρησιμοποιήθηκε το περιβάλλον Github για version control και continuous integration, όπως και για τη φιλοξενία της διαδικτυακής εφαρμογής.

Όπως αναφέρθηκε και παραπάνω, σχεδιάστηκε και υλοποιήθηκε και μία διαδικτυακή εφαρμογή, με σκοπό την εμφάνιση και μελέτη των δεδομένων που συλλέγονται από τις Bluetooth Low Energy/BLE συσκευές - beacons και Puck.js, τα οποία προκύπτουν από τις καθημερινές δραστηριότητες του χρήστη στο χώρο όπου διαμένει.

Μέσω της εφαρμογής τόσο της κινητής (mobile), όσο και της διαδικτυακής υπάρχει η δυνατότητα της "ζωντανής" παρακολούθησης της δραστηριότητας του χρήστη στις τοποθεσίες όπου έχουν ενσωματωθεί οι συσκευές beacons, ώστε να γνωρίζουμε ανα πάσα στιγμή που βρίσκεται. Η συσκευή Puck χρησιμοποιήθηκε για τη λήψη δεδομένων θερμοκρασίας και φωτεινότητας, καθώς και κίνησης του χρήστη καθ' όλη τη διάρκεια της ημέρας, όπως και για την ενημέρωση της κατάστασης της υγείας του, καθώς με το πάτημα του κουμπιού του Puck μπορεί να ενημερώσει για το αν νιώθει καλά ή αν βρίσκεται σε κατάσταση έκτακτης ανάγκης. Ο χρήστης μπορεί να καταχωρήσει όσες ηλεκτρονικές διευθύνσεις αγαπημένων του προσώπων επιθυμεί, έτσι ώστε πατώντας το κουμπί για να δηλώσει έκτακτη ανάγκη ένα ηλεκτρονικό μήνυμα θα σταλεί στις διευθύνσεις αυτές και τα πρόσωπα θα ενημερωθούν άμεσα για την κατάσταση του.

Λέξεις κλειδιά: Beacons, Puck.js, BLE, Internet of Things

ABSTRACT

This diploma thesis was conducted within the framework of the postgraduate program of the University of Piraeus Department of Digital Systems - Direction of Advanced Information Systems.

The main purpose of the work is to monitor the daily activities of the user at the place where he lives by using low power Bluetooth devices. Such devices are the beacons which have been integrated and observed in many areas, such as transport, commerce, health care and many more. They transmit their signal every specific time, which is recognized by other "smart" mobile devices and so we can know which beacon is near us. Also used was the Espruino Puck device with integrated sensors for communicating with other devices in the Internet of Things network and collecting temperature, brightness and motion data.

The development of the mobile application was made using the Android Studio v3.1.4 environment, which provides all the tools for developing such an application and the user interfaces of and Firebase's Cloud Firestore platform as a database. Additionally, the Github environment was used for version control and continuous integration, as well as for hosting the web application.

As we mentioned above, an online application was designed and implemented to display and study the data collected by the Bluetooth Low Energy / BLE devices - beacons and Puck.js, which result from the daily activities of the user in the area where stays.

Through the implementation of both mobile and online, there is the possibility of "live" monitoring of the user's activity on the locations where the beacons are incorporated so that we know wherever he is. The Puck was used to capture temperature and brightness data as well as user movement throughout the day as well as to update its health status as pressing the Puck button can tell if he feels good or is in an emergency. The user can register as many e-mail addresses as desired by the person so that by pressing the button to declare an emergency an e-mail will be sent to those addresses and the persons will be immediately informed of his situation.

Keywords: Beacons, Puck.js, BLE, Internet of Things

ΕΥΧΑΡΙΣΤΙΕΣ

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κ. Κυριαζή Δημοσθένη για την εμπιστοσύνη που μου έδειξε, αναθέτοντας μου τη διπλωματική αυτή εργασία, την καθοδήγησή του, τη βοήθεια και τις πολύτιμες συμβουλές του. Επίσης θα ήθελα να ευχαριστήσω θερμά τον μηχανικό λογισμικού και ερευνητή κ. Ανδρέα Μενύχτα για την πολύτιμη βοήθειά του σε τεχνικά ζητήματα και την καθοδήγηση του. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και την κοπέλα μου Μαρία για την υποστήριξη τους.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Κεφάλαιο 1. Εισαγωγή	1
1.1 Σκοπός της Διπλωματικής Εργασίας	2
1.2 Οργάνωση Κειμένου	2
Κεφάλαιο 2. Διαδίκτυο Πραγμάτων και Έξυπνες Συσκευές	13
2.1 Πρωτόκολλο Bluetooth Low Energy (BLE)	18
2.2 Συσκευές Beacons	26
2.3 Τρόπος λειτουργίας συσκευών Beacons	23
2.4 Ζώνες απόστασης μεταξύ beacons και άλλων συσκευών	24
2.5 Τεχνικά χαρακτηριστικά συσκευής Beacon	27
2.6 Ρυθμίσεις συσκευών Beacons	28
2.7 Εγκατάσταση συσκευής Beacon - Δημιουργία περιοχής χώρου/Region	32
Κεφάλαιο 3. Συσκευή Espruino Puck.js	34
3.1 Τεχνικά χαρακτηριστικά συσκευής Puck.js	35
3.2 Επικοινωνία συσκευής Puck	35
3.3 Περιβάλλον ανάπτυξης Χρήστη - Espruino Web IDE	36
3.4 Σύνδεση συσκευής Puck μέσω του Web IDE	38
3.5 Προεγκατεστημένες Μέθοδοι - Λειτουργίες Puck.js	39
3.5.1 Φωτάκια - LEDs	40
3.5.2 Κουμπί - Button	40
3.5.3 Λειτουργία Bluetooth	40
3.5.4 Μαγνητόμετρο - Magnetometer	40
3.5.5 NFC - Near Field Communications	41
3.5.6 Σένσορας φωτεινότητας - Light sensor	42
3.5.7 Σένσορας Θερμοκρασίας	42
3.5.8 Επίπεδο Μπαταρίας - Battery level	42
3.6 Μετάδοση/Διαφήμιση δεδομένων με τη συσκευή Puck	43
3.7 Υπηρεσίες και Χαρακτηριστικά	46
3.8 Υπηρεσίες συσκευής Puck.js	47

3.9 GATT Profile - Γενικό προφίλ χαρακτηριστικών	47
3.9.1 Τοπολογία συνδεδεμένου δικτύου - Connected Network Topology	48
3.9.2 Ενέργειες GATT	49
3.10 Προγραμματισμός συσκευής Puck	51
3.11 Κατανάλωση Μπαταρίας - Power Consumption	54
Κεφάλαιο 4. Firebase - Cloud Firestore	56
4.1 Μοντέλο δεδομένων Cloud Firestore Data	57
4.1.1 Εγγραφα - Documents	58
4.1.2 Συλλογές - Collections	59
4.2 Βασικές Δυνατότητες Cloud Firestore	61
4.3 Επιλογή τρόπου σύνδεσης του χρήστη στην Εφαρμογή	62
Κεφάλαιο 5. Προδιαγραφές και Αρχιτεκτονική Συστήματος	64
5.1. Προδιαγραφές και Αρχιτεκτονική Συστήματος	64
Κεφάλαιο 6. Υλοποίηση Συστήματος	66
6.1 Αρχική Οθόνη - Welcome Screen	66
6.2 Διεπαφή Σύνδεσης Χρήστη στην Εφαρμογή - LoginActivity	67
6.3 Διεπαφή Εγγραφής Νέου Χρήστη - RegisterActivity	69
6.4 Διεπαφή Καταχώρησης Προσωπικών Στοιχείων Χρήστη	71
6.5 Διεπαφή Ανίχνευσης συσκευών Beacons	73
6.6 Διεπαφή Καταχώρησης Περιοχής/Beacon Region - NewRegionActivity	76
6.7 Διεπαφή Ανίχνευσης συσκευής Puck	79
6.8 Διεπαφή Σύνδεσης με τη συσκευή Puck	79
6.9 Καρτέλα Λίστας Περιοχών Χρήστη - HomeFragment	81
6.10 Διεπαφή Χρήστη Καταχώρησης Σημειώσεων Region - NotesActivity	83
6.11 Καρτέλα Λίστας Εγγράφων Συλλογής PuckData	85
6.12 Καρτέλα Λίστας Εγγράφων Συλλογής RegionEvents	91
6.13 Διεπαφή Απεικόνισης Γραφήματων-TimelineFragment	94
6.14 Καρτέλα Χρήστη Λίστας Ηλεκτρονικών Διευθύνσεων	96
6.15 Διεπαφή Χρήστη Στατιστικών Περιοχής - RegionStatisticsActivity	99
6.16 Διαδικτυακή Εφαρμογή	100

6.17 Σελίδα Δεδομένων Συσκευής Puck.js	102
6.18 Σελίδα Δεδομένων Συσκευών Beacons	102
6.19 Σελίδα Παρακολούθησης Ζωντανής Περιοχής	105
Κεφάλαιο 7. Επίλογος	107
7.1 Σύνοψη	107
7.2 Μελλοντικές Επεκτάσεις	109
Πηγές - Βιβλιογραφία	110

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

Εικόνα 1: Συσκευές Beacons	13
Εικόνα 2: Συσκευές Beacons στο Διαδίκτυο των Πραγμάτων	17
Εικόνα 3: Χώροι στους οποίους χρησιμοποιούνται συσκευές beacons	18
Εικόνα 4: Πρωτόκολλο iBeacon	18
Εικόνα 5: Πρωτόκολλο Eddystone	18
Εικόνα 6: Ζώνες απόστασης μεταξύ Beacon και έξυπνης συσκευής	25
Εικόνα 7: Επωνυμίες Bluetooth Smart Ready/Smart	28
Εικόνα 8: Παράδειγμα τιμών συσκευής beacon (UUID, Major, Minor)	31
Εικόνα 9: Κώδικας δημιουργίας περιοχής/region	32
Εικόνα 10: Πλακέτα συσκευής Puck	34
Εικόνα 11: Μικροελεγκτής Nordic Semiconductor nRF52832	34
Εικόνα 12: Περιβάλλον ανάπτυξης Χρήστη - Espruino Web IDE	36
Εικόνα 13: Graphical Designer περιβάλλοντος προγραμματισμού Espruino	37
Εικόνα 14: Εύρεση διαθέσιμων συσκευών	38
Εικόνα 15: Connected to Port	39
Εικόνα 16: Connected Topology(Central - Peripheral)	49
Εικόνα 17: Send Request/Response between Central and Peripheral	50
Εικόνα 18: Profile/Service/Characteristics	50
Εικόνα 19: Κομμάτι κώδικα Puck(1ο μέρος)	52
Εικόνα 20: Κομμάτι κώδικα Puck(2ο μέρος)	53
Εικόνα 21: Αναμμένο μπλε LED - Διαφήμιση των δεδομένων Puck	54
Εικόνα 22: Cloud Firestore	56
Εικόνα 23: Κανόνες Βασής Δεδομένων Cloud Firestore	61
Εικόνα 24: Καρτέλα Authentication Cloud Firestore	61
Εικόνα 25: Λίστα λογαριασμών χρηστών	62
Εικόνα 26: Πάροχοι Σύνδεσης στην Εφαρμογή	62
Εικόνα 27: Περιβάλλον Cloud Firestore	63

Εικόνα 28: Συλλογή PuckData	63
Εικόνα 29: Έγγραφο της συλλογής PuckData	63
Εικόνα 30: Αρχική Διεπαφή Χρήστη	66
Εικόνα 31: Διεπαφή Σύνδεσης Χρήστη στην Εφαρμογή - LoginActivity	67
Εικόνα 32: Μέθοδος signInWithEmailAndPassword της Cloud Firestore	68
Εικόνα 33: Οθόνη Εγγραφής Χρήστη/Δημιουργίας Λογαριασμού	69
Εικόνα 34: Μέθοδος Εγγραφής Νέου Χρήστη της Cloud Firestore	70
Εικόνα 35: Διεπαφή Καταχώρησης Προσωπικών Στοιχείων Χρήστη	71
Εικόνα 36: Έγγραφο με το ID χρήστη και στοιχεία αυτού	72
Εικόνα 37: Διεπαφή Ανίχνευσης συσκευών Beacons	73
Εικόνα 38: Οθόνη Ανίχνευσης συσκευών Beacons	74
Εικόνα 39: Επιλογή Unregister Beacon	75
Εικόνα 40: Διεπαφή Καταχώρησης Περιοχής/Beacon Region	76
Εικόνα 41: Παράθυρο ειδοποίησης για ανίχνευση beacons	77
Εικόνα 42: Διεπαφή Ανίχνευση συσκευής Puck	79
Εικόνα 43: Διεπαφή σύνδεσης με τη συσκευή Puck	79
Εικόνα 44: Γραμμή Εργαλείων - Μενού Διεπαφής Σύνδεσης με Puck	80
Εικόνα 45: Connection State & Device Address	80
Εικόνα 46: Καρτέλα Λίστας Περιοχών Χρήστη - HomeFragment	81
Εικόνα 47: Παράθυρο Διαγραφής Περιοχής Region	82
Εικόνα 48: Διεπαφή χρήστη καταχώρησης σημειώσεων περιοχής/region	83
Εικόνα 49: Παράθυρο Επεξεργασίας Σημείωσης	84
Εικόνα 50: Παράθυρο Διαγραφής Σημείωσης	84
Εικόνα 51: Καρτέλα Λίστας Εγγράφων Συλλογής PuckData	85
Εικόνα 50: Ειδοποίηση Έκτακτης Ανάγκης	87
Εικόνα 51: E-mail έκτακτης ανάγκης	88
Εικόνα 52: Ειδοποίηση Χαμηλής Μπαταρίας	88
Εικόνα 53: Καρτέλα Δεδομένων Κατάστασης Έκτακτης ανάγκης	89
Εικόνα 54: Καρτέλα Δεδομένων Καλής κατάστασης χρήστη	89
Εικόνα 55: Παράθυρο Συνδέσεων Puck-Κινητής Συσκευής Χρήστη	90
Εικόνα 56: Καρτέλα Λίστας Εγγράφων Συλλογής RegionEvents	91

Εικόνα 57: Καρτέλα δεδομένων εισόδου στην περιοχή/region	93
Εικόνα 58: Καρτέλα δεδομένων εξόδου από την περιοχή/region	93
Εικόνα 59: Διεπαφή Χρήστη - Γραφήματα Θερμοκρασίας και Φωτεινότητας Περιοχής.	94
Εικόνα 60: Παράθυρο τιμής θερμοκρασίας data point	95
Εικόνα 61: Παράθυρο τιμής φωτεινότητας data point	95
Εικόνα 62: Καρτέλα Χρήστη Λίστας Ηλεκτρονικών Διευθύνσεων	96
Εικόνα 63: Παράθυρο επεξεργασίας email	97
Εικόνα 64: Παράθυρο διαγραφής email	97
Εικόνα 65: Διεπαφή Χρήστη Καταχώρησης E-mail	98
Εικόνα 66: Διεπαφή Αριθμητικών Μεγεθών Περιοχής	99
Εικόνα 67: Web App Login Page	100
Εικόνα 68: Μήνυμα Απόρριψης Εισόδου/Λανθασμένος Κωδικός Πρόσβασης	101
Εικόνα 69: Μήνυμα Απόρριψης Εισόδου/Λανθασμένη Ηλεκτρονική Διεύθυνση	101
Εικόνα 70: Σελίδα Δεδομένων Συσκευής Puck.js	102
Εικόνα 71: Καρτέλα Δεδομένων Puck.js	103
Εικόνα 72: Ημερομηνία της εκάστοτε ημέρας	103
Εικόνα 73: Σελίδα Δεδομένων Δεδομένων Συσκευών Beacons	103
Εικόνα 74: Καρτέλα Δεδομένων Beacon	104
Εικόνα 75: Σελίδα Παρακολούθησης Ζωντανής Περιοχής	104
Εικόνα 76: Καρτέλα Στοιχείων χρήστη και τωρινή περιοχή	105
Εικόνα 77: Καρτέλα Στοιχείων περιοχής/χώρου	105
Εικόνα 78: Παρουσίαση κατάστασης ζωντανής τοποθεσίας	106
Εικόνα 79: Καρτέλα στοιχείων χρήστη - Live Region ζωντανής τοποθεσίας	106
Εικόνα 80: Καρτέλα Live Region	106

Κεφάλαιο 1. Εισαγωγή

1.1 Σκοπός της Διπλωματικής Εργασίας

Οι τεχνολογικές εξελίξεις των τελευταίων ετών έχουν οδηγήσει σε μία εκρηκτική αύξηση του αριθμού των έξυπνων συσκευών και των έξυπνων εφαρμογών, φέρνοντας όλο και πιο κοντά τον φυσικό χώρο με τον χώρο του λογισμικού, σε έναν σύστημα που ονομάζεται Διαδίκτυο Πραγμάτων.

Σκοπός της διπλωματικής εργασίας είναι η καταγραφή και ανάλυση των δεδομένων που λαμβάνουμε από τις έξυπνες συσκευές, με τελικό αποτέλεσμα την αναγνώριση πιθανών δραστηριοτήτων του χρήστη καθημερινά ή αναγνώριση και έγκαιρη ενημέρωση για τυχόν έκτακτες καταστάσεις στην υγεία του χρήστη.

Συγκεκριμένα, αντικείμενο της εργασίας είναι η ανάπτυξη ενός τεχνικού υπόβαθρου, με τη μορφή μιας πρότυπης εφαρμογής για κινητές συσκευές (mobile), στην οποία χρησιμοποιούνται συσκευές Bluetooth χαμηλής ενέργειας (BLE) τις οποίες ο χρήστης θα πρέπει να ενσωματώσει σε περιοχές εντός του χώρου στον οποίο διαμένει. Με τη βοήθεια των συσκευών μπορούμε να λαμβάνουμε ακριβή δεδομένα σχετικά με την τοποθεσία του χρήστη την εκάστοτε ημερομηνία και ώρα. Οι συσκευές beacons και η συσκευή Puck.js επικοινωνούν μεταξύ τους με αποτέλεσμα να λαμβάνουμε δεδομένα κάθε ορισμένο χρονικό διάστημα ή όποτε το θελήσουμε με το πάτημα ενός κουμπιού το οποίο είναι ενσωματωμένο στη συσκευή Puck.

Μέσω της κινητής εφαρμογής, ο χρήστης μπορεί να επεξεργαστεί πληροφορίες του προσωπικού του λογαριασμού ανά πάσα στιγμή. Βασική λειτουργία της εφαρμογής είναι η ανίχνευση συσκευών beacons και η καταχώρηση των πληροφοριών τους κατά τη δημιουργία των περιοχών/regions που επιθυμεί ο χρήστης. Μπορεί να διαγράψει κάποια από αυτές και να δηλώσει μία εκ νέου τοποθετώντας τη σε κάποιον άλλο χώρο εντός της κατοικίας του.

Η συσκευή Puck χρησιμοποιήθηκε για τη λήψη δεδομένων θερμοκρασίας και φωτεινότητας, καθώς και κίνησης του χρήστη καθ' όλη τη διάρκεια της ημέρας, όπως και για την ενημέρωση της κατάστασης της υγείας του, καθώς με το πάτημα του κουμπιού του Puck μπορεί να ενημερώσει για το αν νιώθει καλά ή αν βρίσκεται σε κατάσταση έκτακτης ανάγκης. Η συσκευή διαθέτει τρία φωτάκια LED τα οποία θα αναβοσβήσουν όταν ο χρήστης θα θέλει να πατήσει το κουμπί της, ώστε να του γίνει γνωστό πως η κατάσταση του δηλώθηκε επιτυχώς.

Ο χρήστης μπορεί να καταχωρήσει όσες ηλεκτρονικές διευθύνσεις αγαπημένων του προσώπων επιθυμεί, έτσι ώστε πατώντας το κουμπί για να δηλώσει έκτακτη ανάγκη ένα ηλεκτρονικό μήνυμα θα σταλεί στις διευθύνσεις αυτές και τα πρόσωπα θα ενημερωθούν άμεσα για την κατάσταση του.

Το αντικείμενο της παρούσας διπλωματικής εργασίας αφορά σε μια πρωτότυπη εφαρμογή η οποία θα μπορούσε να επεκταθεί μελλοντικά προσθέτοντας σε αυτήν μια πληθώρα νέων λειτουργιών σε συνεργασία με την ενσωμάτωση και άλλων έξυπνων συσκευών.

1.2 Οργάνωση Κειμένου

- Στο Κεφάλαιο 2 θα γίνει η παρουσίαση του υπόβαθρου της διπλωματικής. Συγκεκριμένα, θα αναφερθούν λεπτομερώς οι τεχνολογίες που χρησιμοποιήθηκαν καθώς και η τεχνολογία των συσκευών beacons.
- Στο Κεφάλαιο 3 θα γίνει ανάλυση της συσκευής Bluetooth Low Energy Puck.js η οποία έπαιξε σημαντικό ρόλο στην επίτευξη των στόχων της παρούσας διπλωματικής εργασίας.
- Στο κεφάλαιο 4 παρουσιάζεται η πλατφόρμα της Cloud Firestore και επεξηγούνται διάφορες λειτουργίες που παρέχονται καθώς και οι συλλογές και τα έγγραφα δεδομένων της βάσης δεδομένων.
- Στο Κεφάλαιο 5 γίνεται η ανάλυση των προδιαγραφών της εφαρμογής.
- Στο Κεφάλαιο 6 γίνεται η περιγραφή της υλοποίησης τόσο της Android εφαρμογής όσο και της διαδικτυακής εφαρμογής, και εξηγείται λεπτομερώς πως επιτελείται, από τεχνικής άποψης, η κάθε λειτουργία της εφαρμογής.
- Στο Κεφάλαιο 7, τέλος, γίνεται μια σύντομη σύνοψη της εφαρμογής, καθώς και μια αναφορά στις πιθανές μελλοντικές επεκτάσεις της εφαρμογής και τις πιθανές προσθήκες άλλων λειτουργιών.

λοσοφία του IoT είναι η σύνδεση όλων των ηλεκτρονικών συσκευών μεταξύ τους (τοπικό δίκτυο) ή με δυνατότητα σύνδεσης στο διαδίκτυο παγκόσμιο ιστό.[1]

Η έννοια "Things" (πράγματα) δεν είναι αυστηρά συνδεδεμένη με ορισμένα προϊόντα. Αναφέρεται σε μία ευρεία ποικιλία συσκευών εντελώς διαφορετικά μεταξύ τους, όπως για παράδειγμα αυτοκίνητα με ενσωματωμένους αισθητήρες, κάμερες, κλιματιστικά, φώτα, συστήματα ασφαλείας, smartwatches ακόμα και αυτοκίνητα των οποίων οι περίπλοκοι αισθητήρες εντοπίζουν αντικείμενα στην πορεία τους. Είναι μερικά από τα πολλά προϊόντα τεχνολογίας. Βασικό χαρακτηριστικό όλων είναι η σύνδεση μεταξύ τους με απώτερο σκοπό την δυνατότητα του χρήστη να τα ελέγχει από έναν υπολογιστή ή κινητό. Ο όρος Internet of Things αποδόθηκε την δεκαετία του 1990 από τον Kevin Ashton.

Το IoT αποτελεί μια τεχνολογία που θα αλλάξει τις ισορροπίες στο μέλλον, θα φέρει δραματικές αλλαγές στις αγορές, τις υπηρεσίες υγείας και στη βιομηχανία. Τα δεδομένα που θα συγκεντρώνονται θα μπορούν να χρησιμοποιηθούν για τη βελτίωση της απόδοσης, τον εντοπισμό και την πρόβλεψη των αναγκών των ανθρώπων και των οργανισμών πριν αυτές εκδηλωθούν. Χαρακτηριστική είναι η περίπτωση της πόλης του Όσλο, όπου με τέτοιου είδους έξυπνες λύσεις επιτεύχθηκε η μείωση του ενεργειακού κόστους κατά 62%!

Το Internet of Things αποτελεί κάτι περισσότερο από μια διευκόλυνση για τους καταναλωτές, δεδομένου ότι δημιουργεί νέες πηγές πληροφοριών, νέα επιχειρηματικά μοντέλα, νέες υπηρεσίες και νέα καινοτόμα προϊόντα σε πολλούς κλάδους. Ενδεικτικά κάποιοι κλάδοι που θα επηρεαστούν άμεσα είναι:

Υγειονομική περίθαλψη & υπηρεσίες υγείας

Πολλοί άνθρωποι παγκοσμίως ήδη χρησιμοποιούν smartwatches ή άλλες έξυπνες συσκευές για να παρακολουθούν την κατάσταση της υγείας τους. Εκτιμάται πως στο άμεσο μέλλον στα νοσοκομεία η παρακολούθηση των ασθενών θα γίνεται μέσω μόνιτορ τα οποία θα είναι συνδεδεμένα σε ένα δίκτυο IoT. Με αυτόν τον τρόπο θα βελτιωθούν οι υπηρεσίες υγείας και υγειονομικής περίθαλψης, θα να εξοικονομηθεί χρόνος και θα μειωθεί δραματικά το κόστος περίθαλψης των ασθενών.

Μεταφορές

Κάποτε τα αυτοκινούμενα αυτοκίνητα φάνταζαν σενάριο επιστημονικής φαντασίας, τα τελευταία χρόνια εταιρίες όπως η Tesla Motors, η BMW και η Volvo έχουν δημιουργήσει αυτοκίνητα τα οποία κινούνται αυτόνομα ή με την επίβλεψη του ανθρώπου. Αξιοποιώντας την τεχνολογία αυτή τα αυτοκίνητα είναι συνδεδεμένα στο διαδίκτυο συλλέγοντας πληροφορίες για τη διαδρομή, την κίνηση στους δρόμους, τις κλιματολογικές συν-

θήκες και τη κατάσταση του οδοστρώματος μετέπειτα αναλύουν και επεξεργάζονται τις πληροφορίες έτσι μπορούν να αποφασίσουν ποια είναι η κατάλληλη ταχύτητα και βέλτιστη διαδρομή που πρέπει να ακολουθηθεί. Το 2016 η εταιρία μεταφορών Uber πρόσθεσε στο αυτοκινητιστικό της στόλο τα πρώτα αυτοκινούμενα αυτοκίνητα τα οποία θα προσφέρουν υπηρεσίες μεταφορών οι οποίες θα κοστίζουν λιγότερο.

Λιανικό Εμπόριο

Τόσο οι καταναλωτές όσο και τα καταστήματα μπορούν να επωφεληθούν από τις λειτουργίες IoT στο εμπόριο. Οι επιχειρήσεις θα μπορούν να παρακολουθούν τα αποθέματα τους αυτοματοποιημένα και θα ειδοποιούνται σε πραγματικό χρόνο για την πορεία λειτουργιών ή επιχειρησιακών διαδικασιών. Θα αυξηθεί η ροή πληροφοριών που θα έχει στη διάθεση της η επιχείρηση με τις εφαρμογές mobile wallet να έρχονται και να προστίθονται με την σειρά τους στα εργαλεία της εταιρίας που βασίζονται στην τεχνολογία IoT. Επίσης, μέσω χρήσης αισθητήρων και των συστημάτων παρακολούθησης θα γνωρίζει ανά πάσα ώρα και στιγμή πόσοι καταναλωτές βρίσκονται στο κατάστημα αλλά και τι ενέργειες πραγματοποιούν. Βασισμένη στο IoT έχει αναπτυχθεί τεχνολογία η οποία ανιχνεύει και ταυτόχρονα αναλύει τις ανθρώπινες εκφράσεις. Έτσι η επιχείρηση μπορεί να εξάγει πληροφορίες σχετικά με τις προτιμήσεις των καταναλωτών βασισμένες στις εκφράσεις και τις συνήθειες τους. Εκμεταλλευόμενοι αυτή την τεχνολογία η διαφήμιση και η προώθηση των προϊόντων θα γίνεται πιο αποτελεσματικά και άμεσα. Από τη μεριά των καταναλωτών οι αγορές θα γίνουν γρηγορότερες αφού οι νέοι τρόποι πληρωμής που θα ενταχθούν θα κάνουν τις πληρωμές αυτοματοποιημένες. Επιπρόσθετα, μέσω των πληροφοριών που θα έχουν συλλέξει οι εταιρίες θα μπορούν να προσφέρουν εξατομικευμένα προϊόντα και υπηρεσίες αυξάνοντας κατακόρυφα την ικανοποίηση των πελατών τους.

Η μεγαλύτερη πρόκληση που θα κληθούν να αντιμετωπίσουν οι εταιρίες είναι η ασφάλεια. Με τη σκέψη πως ένα τεράστιο δίκτυο συνδεδεμένων συσκευών οι οποίες ανταλλάσσουν, επεξεργάζονται και συλλέγουν πληροφορίες. Όλος αυτός ο όγκος θα πρέπει να αρχειοθετηθεί και να ταξινομηθεί με ασφάλεια. Για παράδειγμα, έχουμε αισθητήρες για να παρακολουθούμε την υγεία ενός ασθενή, θα πρέπει να διασφαλίσουμε ότι αυτά τα δεδομένα θα μείνουν ασφαλή και δεν θα πέσουν στα χέρια λάθος ανθρώπων. Εξίσου σημαντική αποτελεί η διαδικασία εύρεσης αξιόπιστων και ενεργειακά αποδοτικών τρόπων αποθήκευσης και ανάλυσης των δεδομένων που θα παράγουν ταυτόχρονα δισεκατομμύρια συσκευές.

Και ενώ ακούγεται επαναστατικό και καινοτόμο ως concept, το IoT δεν είναι κάτι καινούριο στην πράξη, καθώς έχει βασιστεί στις εξελίξεις της τεχνολογίας μέσα στις τελευταίες δεκαετίες. Μία από αυτές τις εξελίξεις αφορά στο κομμάτι των ηλεκτρονικών υπολογιστών - στην εξέλιξή τους δηλαδή από τους κεντρικούς υπολογιστές στους ηλεκτρονικούς υπολογιστές (PC), από τα PC στα notebooks και από τα notebooks στις φορητές

συσκευές (όπως τα tablets, smartphones και τα wearables). Κατά τη διάρκεια αυτής της εξέλιξης, οι συσκευές έγιναν μικρότερες - πια υπάρχουν ακόμα και συσκευές που φοριούνται - καθώς όλα τα εξαρτήματα περιορίστηκαν σε μέγεθος, ενώ οι επιδόσεις τους βελτιώθηκαν - τόσο που αρκετές συσκευές σήμερα ξεπερνούν σε επιδόσεις τους ογκώδεις κεντρικούς υπολογιστές του παρελθόντος.

Παράλληλα και οι τιμές των προϊόντων και των εξαρτημάτων περιορίστηκαν και πλέον είναι οικονομικά προσιτά, κάτι που διευκολύνει την ευρεία υιοθέτηση τους από τους καταναλωτές. Σε τελική ανάλυση, όλες αυτές οι εξελίξεις είχαν ως αποτέλεσμα την κυκλοφορία προϊόντων που συνδέονται μεταξύ τους - συσκευών δηλαδή που αποτελούν το βασικό στοιχείο και την πόρτα που ανοίγει το δρόμο του IoT.

Εκτός από την εξέλιξη του hardware στους υπολογιστές, η πρόοδος σε θέματα συνδεσιμότητας είναι εξίσου σημαντική. Τεχνολογίες όπως η 3G και 4G LTE έχουν προσφέρει στους καταναλωτές τη δυνατότητα να είναι συνδεδεμένοι όπου και αν βρίσκονται. Παράλληλα, οι λειτουργίες σύνδεσης των συσκευών προσφέρουν τη δυνατότητα για άμεση σύνδεση δύο ή και περισσότερων συσκευών μεταξύ τους ή και έμμεση σύνδεση τους μέσω wi-fi ώστε οι συσκευές αυτές να επικοινωνούν μεταξύ τους.

Οι καταναλωτές ήδη διαθέτουν πολλές 'έξυπνες' συσκευές - smart devices που είναι εξοπλισμένες με τους απαραίτητους αισθητήρες και ισχύ επεξεργασίας. Παρόλα αυτά, οι συσκευές αυτές δεν αξιοποιούν τις δυνατότητές τους στο μέγιστο, ενώ η έλλειψη συνεργασίας και συμβατότητας μεταξύ των συσκευών διαφορετικών κατασκευαστών δημιουργεί επιπλέον εμπόδια.

Το Internet of Things είναι μια έννοια που αφορά τα αντικείμενα της καθημερινότητάς μας - από βιομηχανικές μηχανές μέχρι wearable συσκευές που χρησιμοποιούν ενσωματωμένους αισθητήρες για τη συλλογή δεδομένων και την ανάληψη κάποιας δράσης σε αυτά μέσα σε ένα δίκτυο. Κάπως έτσι λειτουργεί ένα κτίριο που χρησιμοποιεί αισθητήρες (sensors) για την αυτόματη ρύθμιση της θέρμανσης ή του φωτισμού. Άλλο παράδειγμα είναι ο ένας εξοπλισμός παραγωγής που προειδοποιεί το προσωπικό συντήρησης για μια επικείμενη βλάβη.

Στην παραδοσιακή ανάλυση, τα δεδομένα αποθηκεύονται και μετά αναλύονται. Ωστόσο, στην περίπτωση των δεδομένων συνεχούς ροής (streaming data) όπως αυτά του IoT, τα μοντέλα και οι αλγόριθμοι είναι αυτοί που αποθηκεύονται και τα δεδομένα περνούν μέσα από αυτά για ανάλυση. Αυτό το είδος της ανάλυσης καθιστά δυνατό τον εντοπισμό και την εξέταση μοτίβων καθώς τα δεδομένα δημιουργούνται, σε πραγματικό χρόνο.

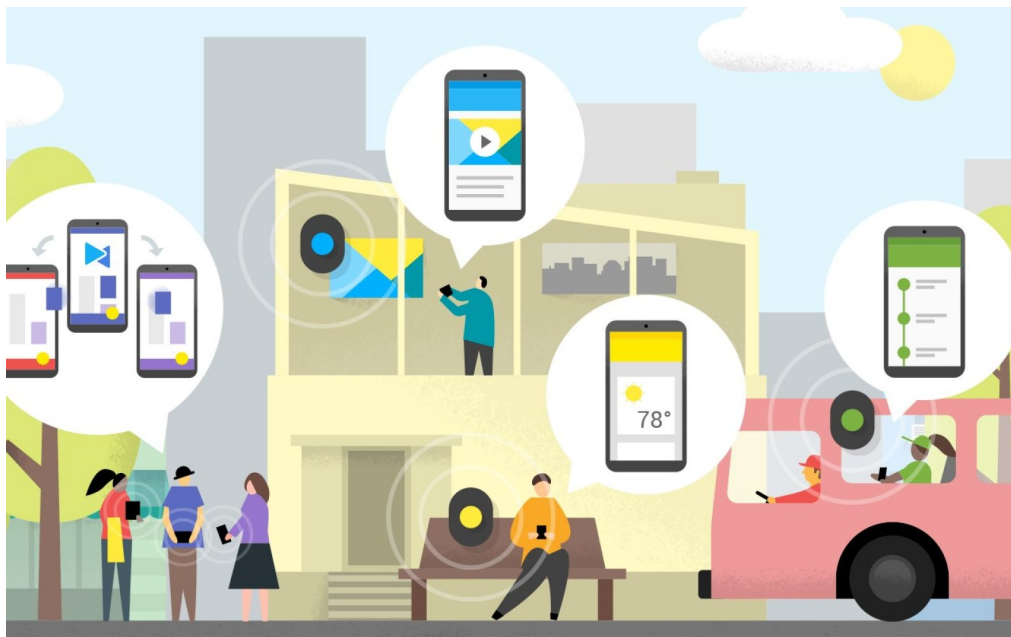
Έτσι, πριν αποθηκευτούν τα δεδομένα, στο cloud ή σε οποιοδήποτε άλλον χώρο αποθήκευσης, υπόκεινται σε επεξεργασία. Έπειτα, χρησιμοποιούνται μέθοδοι analytics

ώστε να αποκρυπτογραφηθούν τα δεδομένα, ενώ όλοι οι συσκευές σας θα συνεχίσουν να εκπέμπουν και να λαμβάνουν δεδομένα.

Με τεχνικές προχωρημένων μεθόδων ανάλυσης - advanced analytics, τα data stream analytics μπορούν να προχωρήσουν πέρα από την απλή παρακολούθηση των υπαρχουσών συνθηκών και την αξιολόγηση των κατώτατων ορίων στην πρόβλεψη μελλοντικών σεναρίων και στην εξέταση πολύπλοκων ερωτημάτων.

Για να εκτιμηθεί το μέλλον με τη χρήση αυτών των ροών δεδομένων (data streams), θα πρέπει να έχετε τεχνολογίες υψηλής απόδοσης που μπορούν να προσδιορίζουν μοτίβα στα δεδομένα σας τη στιγμή που αυτά δημιουργούνται. Μόλις ένα μοτίβο αναγνωρίζεται, μετρήσεις ενσωματωμένες στη ροή δεδομένων, οδηγούν στην αυτόματη προσαρμογή των συνδεδεμένων συστημάτων ή δημιουργούν ειδοποιήσεις για άμεσες δράσεις και λήψη καλύτερων αποφάσεων.

Στο επόμενο κεφάλαιο αναλύεται η τεχνολογία των συσκευών Beacons οι οποίες χρησιμοποιήθηκαν στα πλαίσια υλοποίησης της παρούσας διπλωματικής εργασίας.



Εικόνα 2: Συσκευές Beacons στο Διαδίκτυο των Πραγμάτων

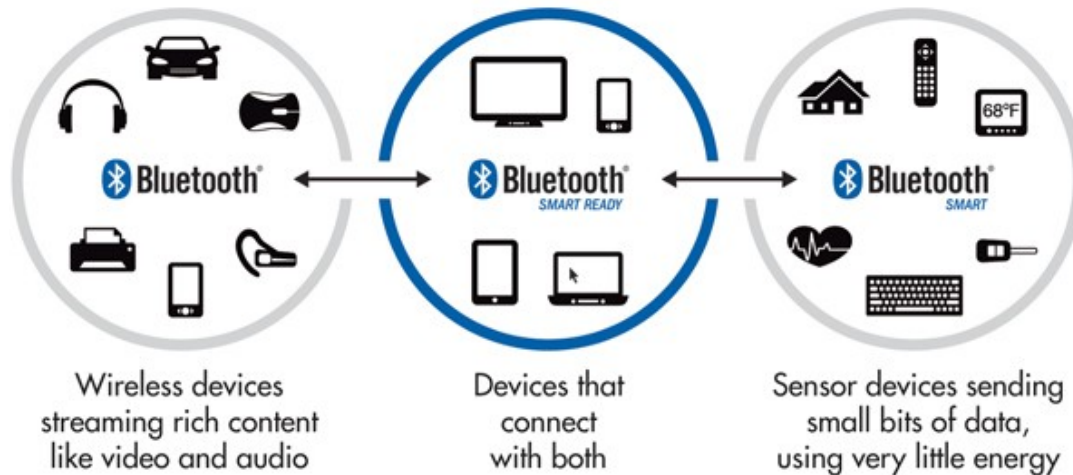
2.1 Πρωτόκολλο Bluetooth Low Energy (BLE)



Τα beacons λειτουργούν αποκλειστικά μέσω του πρωτοκόλλου Bluetooth Low Energy (BLE), μέρος του Bluetooth 4.0 και αποτελούν συσκευές που χαρακτηρίζονται με την επωνυμία Bluetooth Smart. Η πρώτη είσοδος του BLE στην τεχνολογία ήταν το 2006 υπό την ονομασία Wibree αλλά το 2010 ενσωματώθηκε στην οικογένεια του Bluetooth. Η εκδοχή του πρωτοκόλλου Bluetooth 4.0 υποστηρίζει τα ακόλουθα δύο συστήματα ασύρματης τεχνολογίας:

- ✓ **Basic Rate** (BR, συχνά αναφέρεται και ως BR/EDR από το Basic Rate/Enhanced Data Rate)
- ✓ **Bluetooth Low Energy** (BLE)

Το BLE δημιουργήθηκε με σκοπό τη μεταφορά μικρών πακέτων δεδομένων ανά χρονική περίοδο, ενώ καταναλώνει σημαντικά λιγότερη ηλεκτρική ενέργεια από ότι οι συσκευές που λειτουργούν βάσει του BR/EDR. Οι συσκευές που υποστηρίζουν πρωτόκολλο BR και BLE αναφέρονται ως διπλής λειτουργίας (dual-mode devices) και παρουσιάζονται με την επωνυμία Bluetooth Smart Ready. Συνηθέστερα σε ένα σύστημα λειτουργίας βασισμένο στο πρωτόκολλο Bluetooth ένα κινητό τηλέφωνο ή ένας φορητός ηλεκτρονικός υπολογιστής είναι διπλής λειτουργίας συσκευές (dual-mode devices). Οι συσκευές που αναφέρονται ως μοναδικής λειτουργίας (single-mode devices), δηλαδή λειτουργούν μόνο με το πρωτόκολλο BR ή το πρωτόκολλο BLE, παρουσιάζονται με την επωνυμία Bluetooth Smart. Αυτές οι συσκευές μοναδικής λειτουργίας χρησιμοποιούνται στις περιπτώσεις οι οποίες η χαμηλή ενεργειακή κατανάλωση κατέχει πρωταρχική σημασία, όπως συμβαίνει στα beacons.

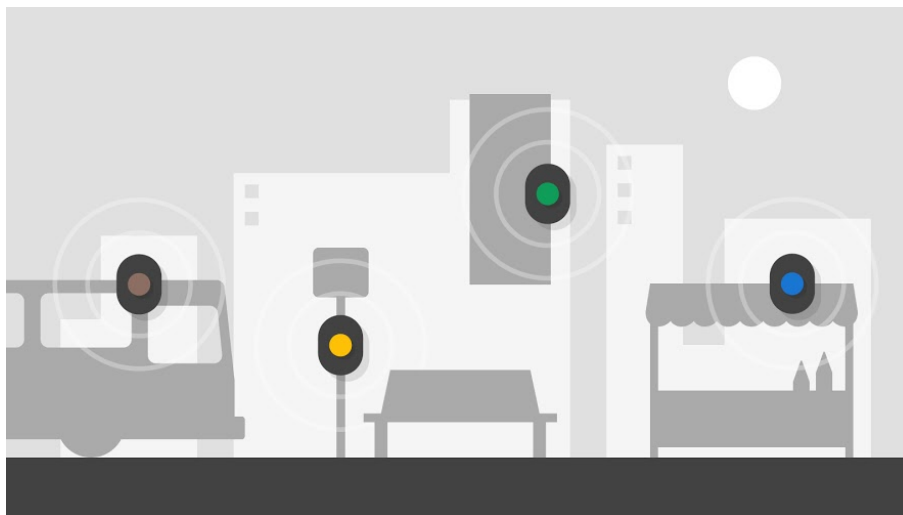


Εικόνα 8: Επωνυμίες Bluetooth Smart Ready/Smart

Το BLE ως μια σχετικά νέα ασύρματη τεχνολογία σχεδιάστηκε για να ικανοποιήσει κυρίως εφαρμογές που απαιτούσαν χαμηλή κατανάλωση όπως αυτές που σκοπό έχουν την παρακολούθηση και τον έλεγχο. Λειτουργεί στο φάσμα των 2.4 GHz ISM (Industrial, Scientific and Medical) ζώνης και διατίθεται ελεύθερα παγκοσμίως.

2.2 Συσκευές Beacons

Τα beacons αποτελούν πομπούς μετάδοσης σημάτων, εντός μικρής εμβέλειας, σε συμβατές, φορητές ηλεκτρονικές συσκευές. Τα σήματα μεταφράζονται σε ενημερωτικά και προωθητικά μηνύματα για τον χρήστη στην προσωπική του έξυπνη, φορητή συσκευή όσο βρίσκεται σε κοντινή απόσταση από τα beacons.



Εικόνα 3: Συσκευές Beacons

Η τεχνολογία Beacon εμφανίστηκε το έτος 2013, κατά το *Apple Worldwide Developers Conference (WWDC)* [2], όπου η εταιρία Apple παρουσίασε το πρωτόκολλο που ανέπτυξε με την ονομασία τεχνολογία iBeacon. Η τεχνολογία δεν αποτελεί μονοπώλιο της Apple, μιας και πλέον είναι συμβατή σε όλες τις συσκευές που υποστηρίζουν Android λογισμικό. Η ονομασία που επικρατεί σήμερα, είναι αυτή της τεχνολογίας Beacon.

Τα beacons αποτελούν μικρές συσκευές, διαφόρων σχημάτων και διαστάσεων, από μέγεθος usb stick έως μεγαλύτερο και πιο ορατό. Λειτουργούν ως πομποί αποστολής ραδιοκυμάτων σε κοντινές τους, φορητές, ηλεκτρονικές, έξυπνες συσκευές (smart devices) όπως κινητά τηλέφωνα (iphones, smartphones), υπολογιστές και ταμπλέτες (tablets). Τα σήματα μεταδίδονται χρησιμοποιώντας το πρότυπο Bluetooth Low Energy (BLE), όπου προσφέρει χαμηλή κατανάλωση ενέργειας, ιδανικό για μεταφορά μικρής ποσότητας δεδομένων. Τόσο η ισχύς του εκπεμπόμενου σήματος, όσο και το χρονικό διάστημα μετάδοσης μεταξύ των σημάτων μπορεί να ρυθμιστεί ώστε να επιτευχθεί η επιθυμητή κάλυψη.

Είναι σαν ένας φάρος: μεταδίδει επανειλημμένα ένα μόνο σήμα που μπορούν να δουν άλλες συσκευές. Αντί να εκπέμπει ορατό φως, μεταδίδει ένα ραδιοφωνικό σήμα UUID που αποτελείται από ένα συνδυασμό γραμμάτων και αριθμών που μεταδίδονται σε κανονικό διάστημα περίπου ένα δέκατο (1/10) του δευτερολέπτου. Μια συσκευή εξοπλισμένη με Bluetooth, όπως ένα smartphone, μπορεί να "βλέπει" ένα beacon μόλις βρίσκεται σε εμβέλεια.

Τα Eddystone και iBeacon είναι πρωτόκολλα επικοινωνίας. Όπως αναφέρθηκε προηγουμένως, τα beacons στέλνουν σήματα Bluetooth σε συγκεκριμένα χρονικά διαστήματα και τα πρωτόκολλα επικοινωνίας περιγράφουν τη μορφή που συνθέτει το σήμα.



Εικόνες 4 & 5: Πρωτόκολλο iBeacon



Πρωτόκολλο Eddystone

Οι φάροι Bluetooth διαδραματίζουν εξαιρετικά σημαντικό ρόλο τόσο στο μέλλον του Διαδικτύου Πραγμάτων (IoT) όσο και στην εγγύτητα. Τα beacons έχουν αυξανόμενη σημασία στην εγγύτητα για πολλούς λόγους, όπως:

- **Υψηλά ποσοστά διείσδυσης:** πάνω από το 90% των έξυπνων συσκευών smartphones είναι εξοπλισμένα με Bluetooth. Το γεγονός αυτό καθιστά εφικτή και εύκολη την επικοινωνία μεταξύ beacons και συσκευών.
- **Μεγάλη εμβέλεια:** Η συσκευή μπορεί να μεταδίδει το σήμα της μέχρι και 70 μέτρα.
- **Ενεργειακή απόδοση:** Το BLE είναι πολύ πιο αποδοτικό ενεργειακά από πολλές άλλες μεθόδους.

Τα beacons στέλνουν τους αριθμούς ταυτότητάς (ID) τους περίπου δέκα (10) φορές κάθε δευτερόλεπτο (μερικές φορές περισσότερο, μερικές φορές λιγότερο, ανάλογα με τις ρυθμίσεις του). Μια κοντινή συσκευή με δυνατότητα Bluetooth, όπως το τηλέφωνό μας, παίρνει αυτό το σήμα. Όταν μια εξειδικευμένη εφαρμογή την αναγνωρίζει, συνδέεται με μια ενέργεια ή ένα κομμάτι περιεχομένου αποθηκευμένο στο σύννεφο και το εμφανίζει στον χρήστη.

Interval	Tx Power	Expected Range	Expected Battery Life
100ms	3(-12 dBm)	35m (115')	Up to 7 months
300ms(default)	3(-12 dBm)	35m (115')	Up to 2 years
1000ms or 1s	3(-12 dBm)	35m (115')	Up to 4 years

Παρακάτω αναλύονται συνοπτικά ορισμένα στοιχεία σχετικά με τις προδιαγραφές και τις λεπτομέρειες των συσκευών beacons.

- ✓ **Διάρκεια μπαταρίας:** Οι περισσότερες συσκευές beacons ξεκινούν με διάρκεια μπαταρίας 18-24 μηνών. Ωστόσο, ορισμένες συσκευές με ορισμένες απαιτήσεις και χρήσεις διαρκούν περίπου έξι (6) έως οκτώ (8) μήνες. Επιπρόσθετα υπάρχει η δυνατότητα εξοικονόμησης ενέργειας, με αποτέλεσμα η διάρκεια της μπαταρίας να φτάσει ή και να ξεπεράσει τα πέντε χρόνια. Στο ερώτημα του πως μπορεί η διάρκεια ζωής της μπαταρίας να φτάσει αυτό το μεγάλο χρονικό διάστημα, έρχεται να δώσει απαντήσεις το Bluetooth που κάνει τις συσκευές τόσο αποδοτικές ενεργειακά.

- ✓ **Υποστηριζόμενη μορφή:** Οι συσκευές beacons υποστηρίζουν τα δύο βασικά πρωτόκολλα επικοινωνίας Eddystone και iBeacon. Και τα δύο βασίζονται στο Bluetooth 4.0 και προσφέρουν ουσιαστικά την ίδια λειτουργικότητα.
- ✓ **Διάστημα μετάδοσης σήματος:** Πόσο συχνά μπορεί μια συσκευή beacon να μεταδώσει το σήμα του. Το διάστημα αυτό είναι παραμετροποιήσιμο ανάλογα με το σενάριο που επιθυμεί να υλοποιήσει ο χρήστης. Ο χρόνος μετάδοσης υπολογίζεται σε χιλιοστά του δευτερολέπτου. Η τιμή κυμαίνεται μεταξύ 100ms και 2000ms. Όσο μικρότερη είναι η τιμή τόσο πιο σταθερό είναι το σήμα που εκπέμπει το beacon.
- ✓ **Tx Power:** Η ισχύς μετάδοσης περιγράφει πόσο μακριά μπορεί μία συσκευή beacon να στείλει το σήμα της. Αυτό μπορεί να είναι από μόλις 4 μέτρα, έως 70 με 100 μέτρα. Ωστόσο, δεν είναι απαραίτητο ο αριθμός αυτός να είναι τεράστιος. Ένας φανός εμβέλειας 50 μέτρων μπορεί να είναι εξίσου χρήσιμος με 90 μέτρα ανάλογα με τη συγκεκριμένη χρήση.
- ✓ **Πακέτα:** Το "πακέτο" ενός φάρου είναι τα δεδομένα που μεταδίδει. Αυτό περιγράφει ακριβώς το είδος των πληροφοριών που είναι σε θέση να μεταδώσει. Για παράδειγμα, το iBeacon περιέχει ένα πακέτο (το ίδιο το iBeacon), ενώ ο Eddystone έχει τρεις ξεχωριστές.
- ✓ **Αισθητήρες:** Τον τελευταίο καιρό τα beacons κατασκευάζονται με επιπλέον δυνατότητες. Μπορούν να περιλαμβάνουν επιταχυνσιόμετρα, αισθητήρες φωτός ή κίνησης.
- ✓ **NFC / RFID:** Οι φάροι-beacons εξακολουθούν να είναι πολύ νέοι. Για ορισμένους χρήστες, είναι πολύ σημαντικές οι τεχνολογίες παλαιού τύπου (π.χ ετικέτες NFC και RFID) και τα beacons να επικοινωνούν μεταξύ τους.
- ✓ **Τιμή:** Όσον αφορά τη τιμή των συσκευών beacons, ξεκινούν από μόλις 7-10€. Σε αυτό το σημείο αξίζει να σημειωθεί πως η απόφαση για αγορά τέτοιων συσκευών εξαρτάται εξ' ολοκλήρου από το σκοπό του εκάστοτε χρήστη.

Στον πίνακα που ακολουθεί αναγράφονται οι κύριοι τομείς που έχουν εφαρμοστεί και λειτουργούν εκατομμύρια συσκευές beacons ανα τον κόσμο.

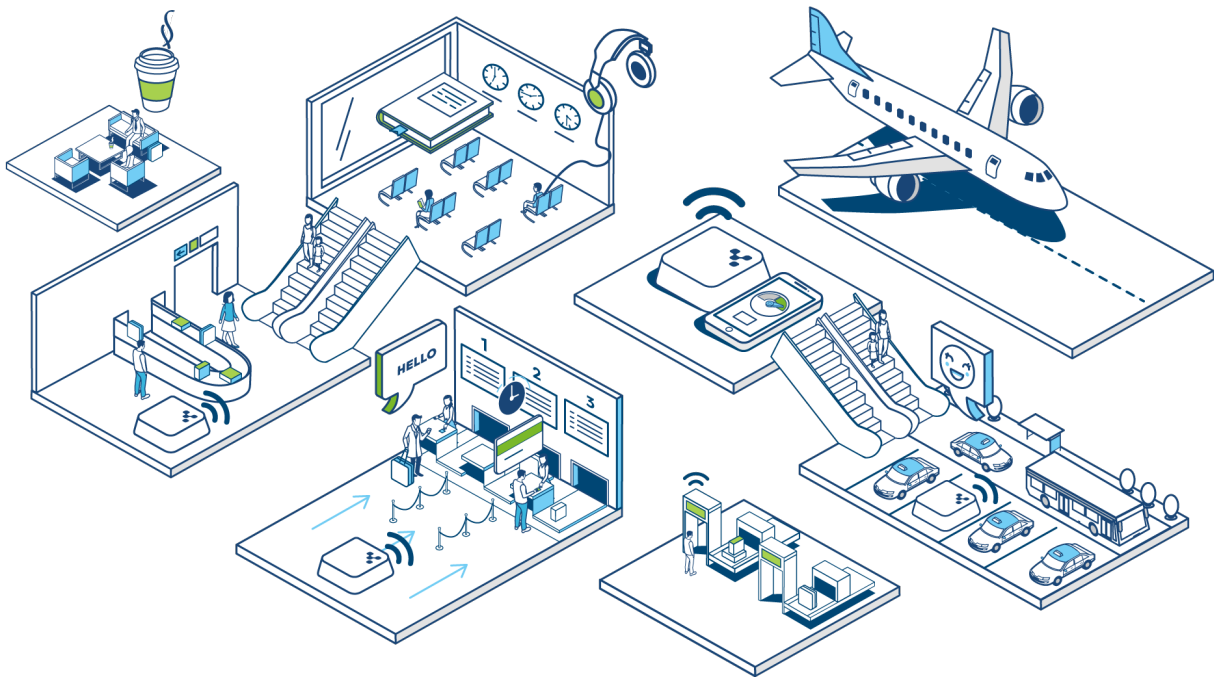
	Λιανικό Εμπόριο		Παραστάσεις
	Εκπαίδευση		Γραφεία
	Πολιτισμικά Γεγονότα		Νοσοκομειακή Περίθαλψη & Οικιακή Φροντίδα
	Αεροδρόμια		

Οι συσκευές beacons χρησιμοποιούνται ήδη σε:

- **Παρακολούθηση Προϊόντων:** Ένα από τα πιο πρακτικά κρούσματα χρήσης των φάρων είναι κάτι που πολλοί από εμάς δεν θα μαντέψαμε ποτέ. Στον τομέα της μεταποίησης και των μεταφορών, οι διαχειριστές πρέπει να γνωρίζουν ακριβώς πού βρίσκονται τα εμπορεύματα ανά πάσα στιγμή. Προσθέτοντας φάρους, μπορούν πάντα να έχουν αυτές τις πληροφορίες. Στην πραγματικότητα, μπορούν ακόμη να δουν τις πληροφορίες από προηγούμενες ημέρες ή εβδομάδες.
- **Πλοήγηση:** Η δημιουργία ακριβούς "GPS για εσωτερική πλοήγηση" είναι μια δημοφιλής περίπτωση χρήσης φανού. Ό, τι κάνουν οι Χάρτες Google για τους υπαίθριους χώρους, οι φάρους μπορούν να κάνουν για εσωτερικούς χώρους. Μπορούν να σας πουν πού είστε και πού πηγαίνετε σε ένα μουσείο, φεστιβάλ ή σιδηροδρομικό σταθμό.
- **Αλληλεπίδραση:** Οι συσκευές beacons μπορούν να κάνουν τις αντιδράσεις αυτοματοποιημένες και να ενεργοποιήσουν τα γεγονότα. Όταν μπαίνετε σε ένα δω-

μάτιο, η λειτουργία του beacon ξεκινά. Αποστέλλει ειδοποιήσεις στις συσκευές που είναι εντός εμβέλειας και έχουν ενεργοποιημένο το Bluetooth.

- **Ασφάλεια:** Εάν βεβαιωθείτε ότι οι ασθενείς δεν πηγαίνουν σε λάθος πτέρυγα ή ειδοποιούν τους εργαζοτες στο εργοστάσιο για επικίνδυνες αλλαγές, οι φάρους μπορούν να στέλνουν αυτόματα ειδοποιήσεις (είτε σε χρήστες εφαρμογών είτε σε ιδιοκτήτες ακινήτων) σχετικά με ένα ζήτημα ασφάλειας. Τα beacons μπορούν επίσης να συνδυαστούν με geofencing για να προσθέσετε ένα επιπλέον επίπεδο στην ασφάλεια των δεδομένων.
- **Ανάλυση:** Τα δεδομένα είναι ένα από τα μεγαλύτερα εργαλεία που διαθέτει η εταιρεία. Τα beacons βοηθούν στην παραγωγή δεδομένων σχετικά με το πού πηγαίνουν οι πελάτες ή όπου συμβαίνουν κοινά προβλήματα σε μια γραμμή συναρμολόγησης. Η ηλεκτρονική πλατφόρμα μπορεί να αποθηκεύσει πληροφορίες σχετικά με τους πομποδέκτες που ενεργοποιούνται και τον τρόπο αλληλεπίδρασης των χρηστών τους.



Εικόνα 6: Χώροι στους οποίους χρησιμοποιούνται συσκευές beacons

2.3 Τρόπος λειτουργίας συσκευών Beacons

Οι συσκευές beacons χρησιμοποιούν το Bluetooth Low Energy (BLE), το οποίο αποτελεί μια μορφή του κοινού σε όλους Bluetooth Πρωτοκόλλου. Είναι σχεδιασμένα με τέτοιο τρόπο ώστε η κατανάλωσή τους σε ενέργεια να είναι χαμηλή.

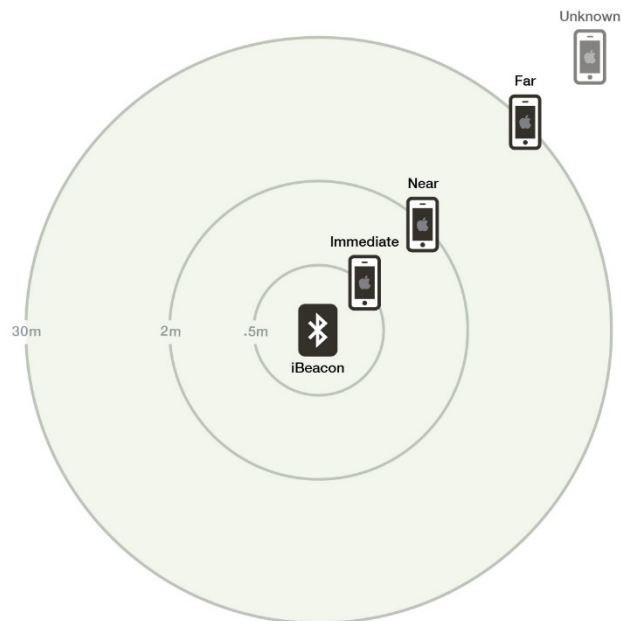
Πηγή ενέργειας τους είναι οι μπαταρίες του εμπορίου. Υπάρχουν δε και φορητές συσκευές beacons που χρησιμοποιούν θύρα USB, ρεύμα δικτύου ή ηλιακούς συλλέκτες. Η μορφή τους ποικίλει και πλέον κατασκευάζονται και σε μορφή πιστωτικής κάρτας ή αυτοκόλλητου τοίχου.

Το εύρος αναμετάδοσης του σήματος τους αλλάζει και προσαρμόζεται ανάλογα τις συνθήκες και τον χώρο στον οποίο θα εγκατασταθούν από το χρήστη. Σημαντική πληροφορία αποτελεί το γεγονός, πως το σήμα των beacons μειώνεται όταν μεσολαβεί τοίχος, κατασκευασμένος από μέταλλο ή τούβλα και πιο συγκεκριμένα αλλάζει σύμφωνα και με το πάχος αυτού.

2.4 Ζώνες απόστασης μεταξύ beacons και άλλων συσκευών

Η απόσταση μεταξύ μίας συσκευής beacon και μίας έξυπνης συσκευής χωρίζεται σε τρεις βασικές ζώνες, οι οποίες διαφοροποιούνται μεταξύ τους με βάση το πόσο κοντινή ή μακρινή είναι η απόσταση. Με βάση αυτό το χαρακτηριστικό των συσκευών beacons, ο εκάστοτε προγραμματιστής μιας εφαρμογής μπορεί να ορίσει τις δικές του λειτουργίες ή ειδοποιήσεις.

Οι ζώνες της απόστασης μεταξύ των συσκευών ορίζονται ως εξής:

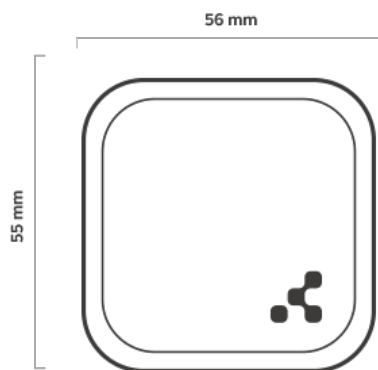


Εικόνα 7: Ζώνες απόστασης μεταξύ Beacon και έξυπνης συσκευής

- Η Μακρινή ζώνη (Far) ενεργοποιείται σε απόσταση μεταξύ 2 με 5 μέτρα και 30 έως 50, αναλογικά με τους εξωτερικούς παράγοντες που όπως περιγράφηκε παραπάνω διαφοροποιούν το εύρος μετάδοσης. Σχεδιάστηκε για τις πρώτες αποκρίσεις μιας συσκευής όταν εισέρχεται στο χώρο όπου λαμβάνει σήματα από κάποιο beacon. Όπως όταν ο χρήστης περιδιαβαίνει το χώρο έξω από ένα κατάστημα που έχουν εγκατασταθεί beacons.
- Η Κοντινή ζώνη (Near) ενεργοποιείται σε απόσταση μεταξύ 50 εκατοστών και 2 έως 5 μέτρα περίπου, σχεδιάστηκε για την αλληλεπίδραση του smart device και του beacon εντός του ιδίου χώρου. Όπως όταν ο χρήστης περιδιαβαίνει το χώρο εντός του καταστήματος εγκατάστασης τους.
- Η ζώνη Ελάχιστης Απόστασης (Immediate) ενεργοποιείται σε απόσταση μικρότερη των 50 εκατοστών, σχεδιάστηκε για την ανταπόκριση κατά την εικονική επαφή smart device και beacon. Όπως όταν ο χρήστης σκανάρει μια προσφορά/διαφήμιση με τη συσκευή του ή βρίσκεται στο στάδιο ολοκλήρωσης της παραγγελίας σε κάποιο κατάστημα.

Τα ραδιοκύματα, ειδικά σε κλειστούς χώρους, διαχέονται, αυτός είναι και ο λόγος που ο διαχωρισμός των ζωνών απόστασης δεν είναι ακριβής. Για τον ίδιο λόγο οι καθυστερή-

σεις εισάγονται τεχνητά έτσι οι ζώνες απόστασης που προαναφέρθηκαν μπορούν άνετα να ελαχιστοποιηθούν στην περίπτωση που η έξυπνη συσκευή - smart device μετακινείται κοντά στα όρια κάθε μίας από αυτές. Κατά την είσοδο της συσκευής σε μία περιοχή γίνεται αισθητή σχεδόν αμέσως, όμως κατά την έξοδο από αυτή, υπάρχει μια κάποια μικρή καθυστέρηση δευτερολέπτων για την ενημέρωση του beacon. Τα σήματα εκπομπής επιτρέπουν επίσης στις εφαρμογές των smart devices να αναγνωρίζουν αν ο χρήστης πλησιάζει το κατάστημα ή απομακρύνεται από αυτό ούτως ώστε να αποστέλλονται τα κατάλληλα μηνύματα καλωσορίσματος ή αποχαιρετισμού. Ακόμα πιο σημαντικό είναι ότι μπορεί να εντοπιστεί από την κίνηση του χρήστη εντός του καταστήματος, ο τομέας αυτού, που υπήρξε μεγαλύτερη αλληλεπίδραση.



Εμβέλεια σήματος συσκευής beacon → 0 - 70m

2.5 Τεχνικά χαρακτηριστικά συσκευής Beacon

Διαστάσεις και Βάρος

Ύψος: 15mm	Πλάτος: 55mm	Βάρος: 35γρ.
------------	--------------	--------------

Processors - Επεξεργαστές

Επεξεργαστής	32-bit ARM Cortex™ M0 CPU core
Bluetooth	Nordic Semiconductor nRF51822
Μνήμη	256KB flash - 16KB RAM

Επικοινωνία

Bluetooth Bluetooth Low Energy ασύρματη τεχνολογία 2.4GHz RF

Μπαταρία και Ισχύς

Μπαταρία: 2 x 1.000mAh CR2477 - Δυνατότητα αλλαγής μπαταρίας

Ισχύς Μετάδοσης Σήματος - Broadcasting power

-30dBm έως 4dBm

Περιβαλλοντικές Απαιτήσεις

Θερμοκρασία-20 ° C / + 60 ° C (-4 ° F / + 140 ° F)
Υγρασία από 0% μέχρι 95%

2.6 Ρυθμίσεις συσκευών Beacons

Για την επίτευξη του σκοπού της παρούσης διπλωματικής εργασίας χρησιμοποιήθηκαν τρεις (3) συσκευές beacons, τις οποίες ο χρήστης μπορεί να τοποθετήσει στους εσωτερικούς χώρους της κατοικίας του και να δώσει ονόματα και περιγραφή σε αυτές.

Η αλλαγή των ρυθμίσεων των συσκευών πραγματοποιείται είτε από εφαρμογή για κινητά τηλέφωνα, είτε από τη διαδικτυακή εφαρμογή του κατασκευαστή Kontakt.io. Γράφοντας τον κωδικό παραγγελίας μας εμφανίζεται ο αντίστοιχος αριθμός beacons και οι προεπιλεγμένες ρυθμίσεις αυτών.

Οι ρυθμίσεις όσον αφορά την ισχύ μετάδοσης του σήματος των συσκευών beacons έχουν ως εξής:

Το πρωτόκολλο επικοινωνίας που επιλέχθηκε είναι iBeacon. Κύρια χαρακτηριστικά αυτού είναι οι τρεις τιμές οι οποίες μπορούν να αλλάξουν από το χρήστη όποτε εκείνος επιθυμεί. Στην επόμενη σελίδα επεξηγούνται με πιο αναλυτικό τρόπο αυτές οι τιμές.

Universally Unique ID (UUID)

Το UUID είναι ένας μοναδικός αριθμός που χρησιμοποιείται για τον προσδιορισμό υπηρεσιών, χαρακτηριστικών και περιγραφικών στοιχείων, γνωστών και ως χαρακτηριστικών. Αυτά τα αναγνωριστικά μεταδίδονται στον αέρα έτσι ώστε π.χ. ένα περιφερειακό μπορεί να ενημερώσει ένα κεντρικό σχετικά με το ποιές υπηρεσίες παρέχει. Για να αποθηκεύσετε τον χρόνο εκπομπής και τον χώρο μνήμης στο nRF52, υπάρχουν δύο είδη UUID:

Ο πρώτος τύπος είναι ένα σύντομο UUID 16-bit. Η προκαθορισμένη υπηρεσία καρδιακού ρυθμού, για παράδειγμα, έχει το UUID 0x180D και ένα από τα κλειστά χαρακτηριστικά της, το χαρακτηριστικό μέτρησης καρδιακού ρυθμού, έχει το UUID 0x2A37. Το UUID 16-bit είναι ενεργειακής και μνήμης αποδοτικής, αλλά επειδή παρέχει μόνο έναν σχετικά περιορισμένο αριθμό μοναδικών αναγνωριστικών, υπάρχει κανόνας. μπορείτε να μεταδώσετε μόνο τα προκαθορισμένα Bluetooth SIG UUID's απευθείας στον αέρα. Ως εκ τούτου, υπάρχει ανάγκη για έναν δεύτερο τύπο UUID, ώστε να μπορείτε να μεταδώσετε και τα δικά σας προσαρμοσμένα UUIDs.

Ο δεύτερος τύπος είναι ένα UUID 128-bit, που μερικές φορές αναφέρεται ως UUID συγκεκριμένου πωλητή. Αυτός είναι ο τύπος UUID που πρέπει να χρησιμοποιήσετε όταν κάνετε τις δικές σας υπηρεσίες και χαρακτηριστικά. Φαίνεται έτσι: "4A98xxxx-1CC4-E7C1-C757-F1267DD021E8" και ονομάζεται "UUID βάσης". Τα τέσσερα **x** αντιπροσωπεύουν ένα πεδίο όπου θα εισαγάγετε τα δικά σας αναγνωριστικά 16-bit για τις προσαρμοσμένες υπηρεσίες και τα χαρακτηριστικά σας και θα χρησιμοποιηθούν ακριβώς όπως ένα προκαθορισμένο UUID. Με αυτόν τον τρόπο μπορούμε να αποθηκεύσουμε το UUID βάσης μία φορά στη μνήμη, να μην ασχοληθούμε ξανά με αυτό και να εργαστούμε με τα αναγνωριστικά 16-bit κανονικά. Μας δίνεται η δυνατότητα να δημιουργήσουμε UUIDs βάσης χρησιμοποιώντας το **nRFgo Studio**.

Τιμές Major & Minor: Οι κύριες και οι δευτερεύουσες τιμές είναι οι αριθμοί που αντιστοιχούν στις συσκευές με πρωτόκολλο επικοινωνίας iBeacon, προκειμένου να τα αναγνωρίσετε με μεγαλύτερη ακρίβεια από τη χρήση μόνο του UUID.

Οι Κύριες και οι Δευτερεύουσες τιμές μπορούν να ρυθμιστούν σε τιμές μεταξύ 0 και 65535.

Το πρότυπο iBeacon απαιτεί την ανάθεση μιας κύριας και μιας δευτερεύουσας τιμής.

Στην παρακάτω Εικόνα εμφανίζεται ένα παράδειγμα για την καλύτερη κατανόηση των προαναφερθέντων τριών τιμών (UUID, Major & Minor).



Εικόνα 9: Παράδειγμα τιμών συσκευής beacon (UUID, Major, Minor)

Στην περίπτωση της εφαρμογής για τη παρακολούθηση των καθημερινών δραστηριοτήτων του χρήστη, οι ρυθμίσεις που χρησιμοποιήθηκαν είναι οι προεπιλεγμένες από τον κατασκευαστή των συσκευών Beacons.

Beacon UUID: **f7826da6-4fa2-4e98-8024-bc5b71e0893e**

Το παραπάνω UUID είναι το προκαθορισμένο από τον κατασκευαστή των συσκευών beacons που χρησιμοποιήσαμε για λογαριασμό της διπλωματικής εργασίας.

Tx Power: **1** ≈ **4 μέτρα**

Διάστημα Αναμετάδοσης σήματος: **625ms**

2. 7 Εγκατάσταση συσκευής Beacon - Δημιουργία περιοχής χώρου/Region

Παρακάτω παρουσιάζεται και επεξηγείται ο κώδικας ο οποίος χρησιμοποιήθηκε για να δημιουργηθεί μία περιοχή/region με ενσωματωμένη μία συσκευή beacon στην επιθυμητή περιοχή.


```

private ProximityManager proximityManager;

public static final ArrayList<IBeaconRegion> homeRegions = new ArrayList<>();
public IBeaconRegion homeRegion;

proximityManager = ProximityManagerFactory.create(this);

proximityManager.configuration()
    .scanPeriod(ScanPeriod.RANGING)
    .scanMode(ScanMode.BALANCED);
homeRegion = new BeaconRegion.Builder()
    .identifier(regionGet)
    .proximity(UUID.fromString("f7826da6-4fa2-4e98-8024-bc5b71e0893e"))
    .major(majorGet)
    .build();

homeRegions.add(homeRegion);

proximityManager.spaces()
    .iBeaconRegions(homeRegions);

```

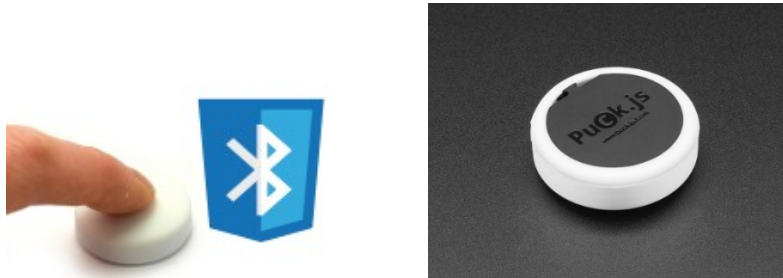


Εικόνα 10: Κώδικας δημιουργίας περιοχής/region

Για τη δημιουργία μιας περιοχής/region στην οποία θα τοποθετηθεί μια συσκευή beacon θα χρειαστεί να εισαχθούν 3 τιμές σύμφωνα με το πρωτόκολλο IBeacon:

- Η τιμή του αναγνωριστικού του beacon **.identifier()**
- Το UUID του beacon (όπως αναφέρθηκε και πρωτίστως στα πλαίσια υλοποίησης της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε το προεπιλεγμένο UUID του κατασκευαστή → **.proximity(UUID.fromString("f7826da6-4fa2-4e98-8024-bc5b71e0893e"))**)
- Η τελευταία τιμή που πρέπει να εισαχθεί είναι η κύρια τιμή(major) του beacon **.major()**
-

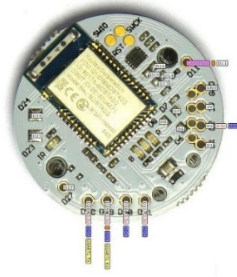
Κεφάλαιο 3. Συσκευή Espruino Puck.js



Στα πλαίσια της εφαρμογής χρησιμοποιήθηκε η συσκευή Bluetooth χαμηλής ενέργειας (Bluetooth Low Energy-BLE), για τη λήψη μέσω αυτής δεδομένων όπως θερμοκρασία, φωτεινότητα. Τεχνικά χαρακτηριστικά της συσκευής αναφέρονται πιο αναλυτικά παρακάτω. Η συσκευή Puck ανήκει στην οικογένεια Espruino και αποτελεί προϊόν χρηματοδότησης, η οποία αποτέλεσε άμεσα εφικτή την παραγωγή της συσκευής.

Το Espruino Puck.js είναι ένα έξυπνο κουμπί-συσκευή Bluetooth με χαμηλή κατανάλωση ενέργειας, το οποίο μπορεί να προγραμματιστεί και να διορθωθεί με ασύρματο τρόπο στέλνοντας εντολές σε JavaScript. Είναι πολυλειτουργικό και εύκολο στη χρήση, με ενσωματωμένο κύκλωμα, το τελευταίο μικρό chip **JS nRF52832** της εταιρείας Nordic Semiconductor. Επίσης διαθέτει Bluetooth χαμηλής ενέργειας- Bluetooth LE, NFC, μαγνητόμετρο, αισθητήρα θερμοκρασίας και φωτεινότητας, φωτάκια LED RGB (πράσινο, κόκκινο, μπλέ), υπέρυθρο και πολλά άλλα, τα οποία περικλείονται σε μια μικρή θήκη από σιλικόνη.

Το Puck.js μπορεί να συνδεθεί με άλλες συσκευές BLE σε λειτουργία “κεντρική” σαν να ήταν τηλέφωνο και μπορεί ακόμη και να λαμβάνει διαφημίσεις δεδομένων άλλων συσκευών. Όλα τα παραπάνω με μία μπαταρία CR2032 που κοστίζει μόνο 0.20 - 0.30€.



Εικόνα 11: Πλακέτα συσκευής Puck



Εικόνα 12: Μικροελεγκτής Nordic Semiconductor nRF52832

3.1 Τεχνικά χαρακτηριστικά συσκευής Puck.js

- Bluetooth χαμηλής ενέργειας - Bluetooth Low Energy
- Espruino JavaScript προεγκατεστημένος διερμηνέας
- nRF52832 SoC - 64MHz ARM Cortex M4, 64kB RAM, 512kB Flash
- 8 x 0.1" GPIO (capable of PWM, SPI, I2C, UART, Analog Input)
- 9 x SMD GPIO (capable of PWM, SPI, I2C, UART)
- Κάλυμμα σιλικόνης με κουμπί αφής
- MAG3110 Μαγνητόμετρο (Magnetometer)
- IR Πομπός
- Ενσωματωμένο θερμομέτρο, σένσορας φωτεινότητας και σένσορας επιπέδου μπαταρίας.
- Red, Green and Blue φώτα LED
- NFC ετικέτα προγραμματιζόμενη σε JavaScript
- Pin ικανό για ανίχνευση χωρητικότητας
- Βάρος συσκευής: 14g με την πλαστική θήκη, 20g με τη συσκευασία
- Διαστάσεις πλαστικής θήκης: 36mm διάμετρος, 12.5mm πάχος
- Διαστάσεις κενής-σκέτης πλακέτας: 29mm διάμετρος, 9mm πάχος

3.2 Επαναφορά συσκευής Puck

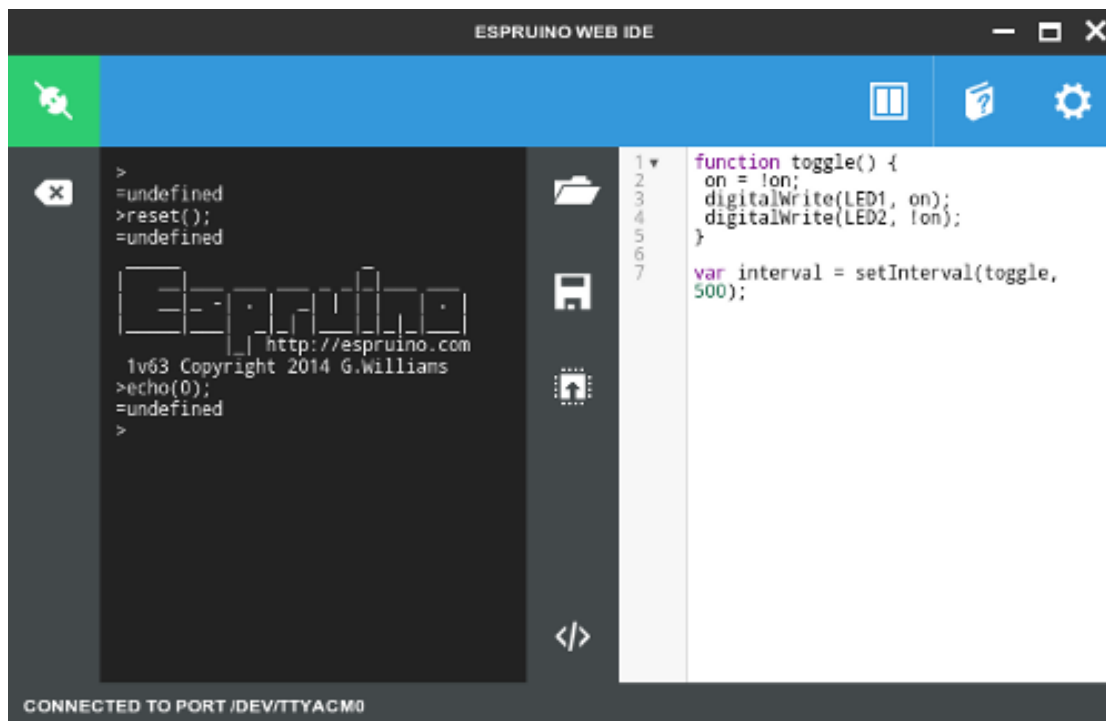
Ενίστε μπορεί να χρειαστεί να πραγματοποιήσουμε επαναφορά της συσκευής Puck στις αρχικές της ρυθμίσεις. Για να το πετύχουμε αυτό πρέπει να ακολουθήσουμε τα εξής παρακάτω βήματα:

- Αφαιρούμε την πλακέτα PCB από τη θήκη σιλικόνης
- Τοποθετούμε κάτι μυτερό με πολύ μικρή επιφάνεια μεταξύ της μπαταρίας και της πλακέτας PCB κοντά στο κείμενο `Puck.js 1.x`

- Τραβάμε απαλά την μπαταρία από την πλακέτα PCB για 1 δευτερόλεπτο και την αφήνουμε
- Στη συνέχεια το κόκκινο φως LED θα πρέπει να αναβοσβήσει 2-3 φορές υποδεικνύοντας ότι η συσκευή Ruck έχει επανεκκινήσει.
- Κατα την επαναφορά ο κώδικας προγραμματισμού της συσκευής Ruck θα έχει χαθεί, με αποτέλεσμα να πρέπει να προγραμματιστεί εκ νέου.



3.3 Περιβάλλον ανάπτυξης Χρήστη - Espruino Web IDE



Εικόνα 13: Περιβάλλον ανάπτυξης Χρήστη - Espruino Web IDE

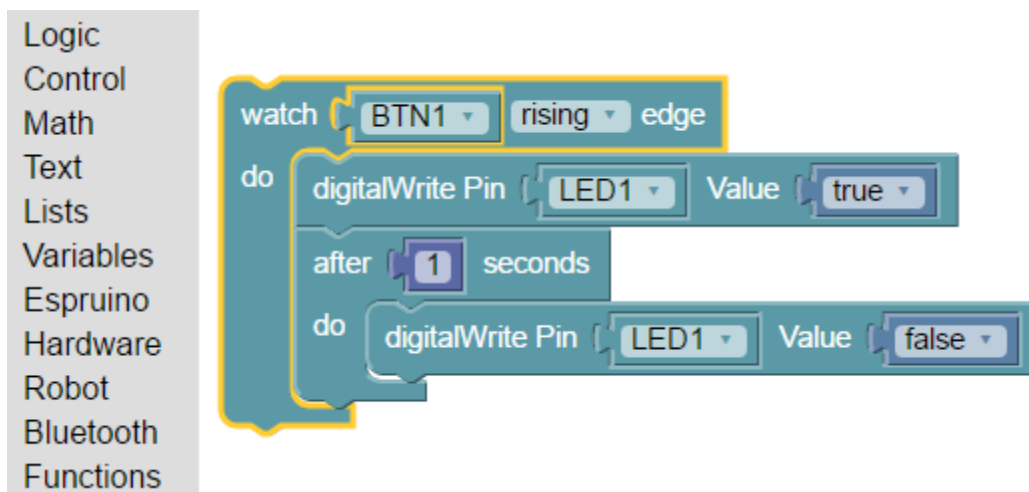
Το περιβάλλον ανάπτυξης του Espruino Web κάνει ακόμα πιο εύκολο τον προγραμματισμό της συσκευής Ruck.

Παρέχει ένα σύγχρονο επεξεργαστή σύνταξης με δυνατότητες ολοκλήρωσης κώδικα και ακόμη και εντοπισμό σφαλμάτων. Επίσης παρέχει τη δυνατότητα στο χρήστη να φορτώνει οποιεσδήποτε μονάδες που χρειάζεται από το διαδίκτυο, για πιο εύκολο και γρήγορο προγραμματισμό της συσκευής.

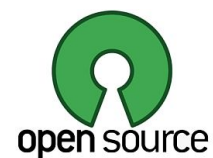
Παρακάτω εμφανίζεται ο σύνδεσμος στον οποίο βρίσκεται το εκτελέσιμο αρχείο του περιέχει το περιβάλλον χρήστη Espruino για τον προγραμματισμό της συσκευής Puck η και άλλων συσκευών της οικογένειας espruino.

<https://www.espruino.com/Quick+Start+BLE#with-an-application>

Μια βασική δυνατότητα που παρέχει το περιβάλλον είναι ο προγραμματισμός των συσκευών με γραφικό σχεδιασμό. Υπάρχουν στο αριστερό μέρος καρτέλες με όλες τις μεθόδους και αντικείμενα, ώστε ο χρήστης να μπορεί εύκολα να ενώσει σαν παζλ τα blocks κώδικα και να προγραμματίσει έτσι την εκάστοτε συσκευή που διαθέτει. Πατώντας την επιλογή στο κάτω μέρος του παραθύρου θα εμφανιστεί στη δεξιά οθόνη το εξής παρακάτω πάζλ με blocks κώδικα, που παρέχονται από το περιβάλλον Espruino και απλώς επιλέγοντας, σύροντας και ενώνοντας τα κομμάτια



Εικόνα 14: Graphical Designer περιβάλλοντος προγραμματισμού Espruino




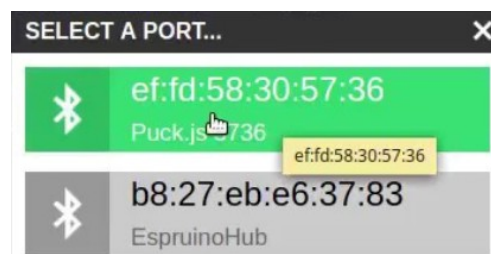
Το λογισμικό της συσκευής Puck είναι ανοιχτού κώδικα και η κοινότητα προγραμματιστών ολοένα και αυξάνεται λόγω των γρήγορων ρυθμών ανάπτυξης των εφαρμογών του διαδικτύου πραγμάτων και την αύξηση και ποικιλομορφία αυτών.

3.4 Σύνδεση συσκευής Puck μέσω του Web IDE


Ο ευκολότερος τρόπος και πιο γρήγορος ώστε ο χρήστης να προγραμματίσει όπως εκείνος επιθυμεί τη συσκευή Puck είναι να συνδεθεί με αυτήν μέσω του περιβάλλοντος Espruino. Τα βήματα τα οποία ακολουθούμε ώστε να πραγματοποιήσουμε σύνδεση με τη συσκευή Puck είναι τα εξής παρακάτω και απαραίτητη προϋπόθεση είναι η ενεργοποίηση του Bluetooth της συσκευής μας.

1) Ανοίγουμε το περιβάλλον χρήστη Espruino.

2) Κάνοντας click στην επιλογή  στο αριστερό επάνω μέρος του παραθύρου θα εμφανιστεί η επιλογή “SELECT A PORT” με τις επιλογές των διαθέσιμων συσκευών, στο κέντρο της εφαρμογής.



Εικόνα 15: Εύρεση διαθέσιμων συσκευών

3) Κάντε click στην επιλογή (π.χ **ef:fd:58:30:57:36**) το οποίο αποτελεί την διεύθυνση MAC της συσκευής. Εφόσον το κάνετε, πρέπει να περιμένετε μερικά δευτερόλεπτα εως ότου να πραγματοποιηθεί η σύνδεση και μετά είναι όλα έτοιμα για τον προγραμματισμό της συσκευής Puck. Το πορτοκαλί εικονίδιο που συμβολίζει τη σύνδεση θα έχει πλέον 




το πράσινο χρώμα, μέχρι και πάλι να γίνει η αποσύνδεση.



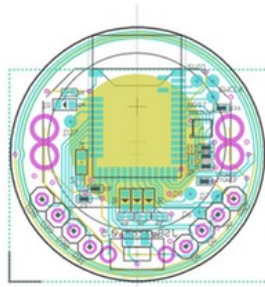
Κάτω δεξιά στο περιβάλλον χρήστη Espruino εφόσον πραγματοποιηθεί σύνδεση με τη συσκευή Puck θα εμφανιστεί η ένδειξη “Connected to Port” (mac address της συσκευής)

CONNECTED TO PORT E9:A8:AF:13:16:F2

Εικόνα 16: Connected to Port

	<p>Κουμπί με το οποίο μπορούμε να φορτώσουμε στο περιβάλλον κάποιο αρχείο κώδικα που έχουμε γράψει.</p>
	<p>Κουμπί με το οποίο μπορούμε να αποθηκεύσουμε τον κώδικα που έχουμε συντάξει, ώστε πολύ εύκολα όποια στιγμή θελήσουμε να το ανοίξουμε φορτώνοντας τον κώδικα.</p>
	<p>Κουμπί με το οποίο στέλνουμε και προγραμματίζουμε τη συσκευή Puck απλά σε λιγότερο από ένα δευτερόλεπτο. Θα εμφανιστεί η ένδειξη «SENT» στο δεξί κάτω μέρος του περιβάλλοντος.</p>

3.5 Προεγκατεστημένες Μέθοδοι - Λειτουργίες Puck.js



Η συσκευή Bluetooth Low Energy Puck.js υποστηρίζει πληθώρα μεθόδων και δυνατοτήτων οι οποίες κάνουν αρκετά εύκολο τον προγραμματισμό της συσκευής, ακόμα και από λιγότερο έμπειρους προγραμματιστές, όσον αφορά σε θέματα λειτουργιών αυτοματοποίησης και επικοινωνίας με άλλες συσκευές. Παρακάτω παραθέτονται και αναλύονται κάποιες βασικές μέθοδοι γραμμένες σε JavaScript, οι οποίες χρησιμοποιήθηκαν για τον προγραμματισμό του Puck για λογαριασμό της διπλωματικής αυτής εργασίας.

3.5.1 Φωτάκια - LEDs

- Μπορούμε να χρησιμοποιήσουμε τη μέθοδο `digitalWrite(LED1,1)` για να ανάψουμε το πρώτο LED (κόκκινο), και τη μέθοδο `digitalWrite(LED1,0)` για να το σβήσουμε. Αντίστοιχα μπορούμε να κάνουμε το ίδιο για το πράσινο LED `LED2` και για το μπλε `LED3`.
- Ένας άλλος τρόπος για να ανάψουμε κάποιο LED είναι να γράψουμε `LED1.write(1)` για να το ανάψουμε και `LED1.write(0)` για να το σβήσουμε.
- As well as `LED1.set ()` and `LED1.reset()` to get the same effect
- Μπορούμε επίσης να ανάψουμε και τα τρία LED μαζί χρησιμοποιώντας το δυαδικό αριθμό γράφοντας `digitalWrite([LED3, LED2,LED1], 7)`
- Ακόμα μία δυνατότητα που παρέχεται στο χρήστη είναι να αναβοσβήσουμε κάποιο LED για ορισμένο χρονικό διάστημα - `digitalPulse(LED2,1,500);` Αυτή η γραμμή κώδικα θα ανάψει και πράσινο φως (`LED2`) για 500ms και μετά αυτό θα σβήσει.

3.5.2 Κουμπί - Button

Μία βασική δυνατότητα που παρέχει η συσκευή Puck και θα αναλυθεί και αναλυτικότερα παρακάτω, είναι το κουμπί που διαθέτει:

- Μπορούμε να έχουμε πρόσβαση στην κατάσταση του κουμπιού με τη μέθοδο `digitalRead(BTN)` ή `BTN.read()`.
- Για να ορίσουμε κάποια λειτουργία με το πάτημα του κουμπιού μπορούμε να χρησιμοποιήσουμε τη μέθοδο **setWatch**, ώστε να καλούμε κάποια άλλη μέθοδο προς εκτέλεση κάθε φορά που πατάμε το κουμπί του Puck.

```
setWatch(function () {
  console.log("Pressed");
```



```
}, BTN, {edge:"rising", debounce:50, repeat:true});
```

3.5.3 Λειτουργία Bluetooth

Η λειτουργία Bluetooth παρέχεται από το αντικείμενο NRF - [NRF object](#). Υπάρχουν πολλές λειτουργίες και τρόποι με τους οποίους μπορούμε να συνδέσουμε τη συσκευή Puck με άλλες Bluetooth συσκευές. Στον παρακάτω σύνδεσμο παρουσιάζονται και επεξηγούνται αυτές οι λειτουργίες και οι μέθοδοι. [NRF.requestDevice](#)

3.5.4 Μαγνητόμετρο - Magnetometer

Βασική δυνατότητα του Puck είναι το ενσωματωμένο μαγνητόμετρο που διαθέτει. Μπορούμε να χρησιμοποιήσουμε τη μέθοδο [Puck.magOn\(\)](#) για να διαβάσουμε τις τιμές των αξόνων **x**, **y** και **z**. Έτσι μπορούμε να

Μπορούμε επίσης να αφήσουμε το μαγνητόμετρο διαρκώς ενεργό, ώστε να λαμβάνουμε τιμές κάθε κάποιο χρονικό διάστημα που επιθυμούμε. Η διαρκής χρήση του μαγνητόμετρου από την άλλη μεριά θα καταναλώσει την μπαταρία σε πολύ γρήγορο χρονικό διάστημα, για αυτό το λόγο θα πρέπει να προγραμματιστεί με τέτοιο τρόπο η συσκευή, ώστε “διαφημίζει” τα δεδομένα(τις τιμές του μαγνητόμετρου) περιοδικά.

```
Puck.magOn();
Puck.on('mag', function(xyz) {
  console.log(xyz);
});
// Απενεργοποιούμε το μαγνητόμετρο με τη μέθοδο Puck.magOff();
```

3.5.5 NFC - Near Field Communications

Η τεχνολογία NFC (Near Field Communications) αφορά την ασύρματη μετάδοση δεδομένων σε απόσταση μικρότερη από τα 10 εκατοστά (συνήθως < 5 cm). Επιτρέπει την γρήγορη ανάγνωση και εγγραφή δεδομένων (σχετικά λίγων- ενδεικτικά 48Byte - 9 Kilobyte) και εκλαμβάνεται ως απόδειξη φυσική παρουσίας.

Το Puck δίνει τη δυνατότητα στους χρήστες να μπορούν να χρησιμοποιούν την τεχνολογία NFC. Με τη μέθοδο [NRF.nfcURL\(\)](#), όπως φαίνεται και αμέσως μετά, μπορούν να ορίσουν κάποια ηλεκτρονική διεύθυνση. Με αυτόν τον τρόπο μπορούν να πλησιάσουν την συσκευή τους, η οποία θα πρέπει να υποστηρίζει την τεχνολογία NFC και να οδηγηθούν στην προκαθορισμένο URL..

```
NRF.nfcURL("https://diam0ndr.github.io/examplerepo/index.html.com");
```

Για να απενεργοποιήσουν τη δυνατότητα του NFC μπορούν να γράψουν:

```
NRF.nfcURL();
```

3.5.6 Σένσορας φωτεινότητας - Light sensor

Για να λάβουμε την τιμή φωτεινότητας από τον ενσωματωμένο σένσορα μπορούμε να καλέσουμε τη μέθοδο `Puck.light()`. Το αποτέλεσμα που θα μας επιστρέψει είναι μια μη βαθμονομημένη τιμή μεταξύ **0** και **1**. Μπορούμε να πάρουμε την τιμή φωτεινότητας σε ποσοστό επί τοις εκατό(%) γράφοντας το εξής: **`Math.round(Puck.light()*100)`**

3.5.7 Σένσορας Θερμοκρασίας

Μπορούμε εύκολα να λάβουμε τη θερμοκρασία στο σημείο που βρίσκεται η συσκευή Puck με τη μέθοδο **`E.getTemperature()`**. Η τιμή που επιστρέφεται σαν αποτέλεσμα είναι η θερμοκρασία σε βαθμούς Κελσίου **°C**.

3.5.8 Επίπεδο Μπαταρίας - Battery level

Μια ακόμη δυνατότητα του Puck είναι να μπορούμε να διαβάσουμε ανα πάσα στιγμή απλά και εύκολα το ποσοστό της μπαταρίας του (με βάση τη μπαταρία CR2032). Αυτό μπορεί να επιτευχθεί με τη μέθοδο `Puck.getBatteryPercentage()`. Μπορούμε επίσης να δούμε το ποσοστό της μπαταρίας σε Volt χρησιμοποιώντας τη μέθοδο `NRF.getBattery()`. Αν θέλουμε να λάβουμε το ποσοστό της μπαταρίας επί τοις εκατό(%) μπορούμε να γράψουμε **`Math.round(NRF.getBattery() * 100)/100`**

3.6 Μετάδοση/Διαφήμιση δεδομένων με τη συσκευή Puck

Βασική δυνατότητα της συσκευής Puck στην οποία βασίστηκε σε μεγάλο βαθμό η υλοποίηση της διπλωματικής αυτής εργασίας, είναι η επικοινωνία του Puck με τις συσκευές beacons αλλά και του κινητού τηλεφώνου του χρήστη.

Μπορούμε να επικοινωνήσουμε με το Puck.js μέσω Bluetooth LE με τρεις βασικούς τρόπους:

1. Μπορούμε να συνδεθούμε με το Puck.js από άλλη συσκευή και να στείλουμε δεδομένα κάτω από την υπηρεσία Nordic UART (ή μια προσαρμοσμένη) - οι περισσότερες συσκευές (PC, Mac, Linux, Android) μπορούν να συνδεθούν μόνο σε 5 ή 6 συσκευές τη φορά.

2. Συνδέση του Puck με την επιθυμητή συσκευή και επικοινωνία με τις υπηρεσίες της συγκεκριμένης συσκευής - οι περισσότερες συσκευές μπορούν να αποδεχθούν μόνο μία ταυτόχρονη σύνδεση.
3. Ένας άλλος τρόπος είναι το Puck.js να διαφημίζει τα δεδομένα του και κάθε συσκευή εντός εμβέλειας να μπορεί να διαβάσει αυτά τα δεδομένα - ενώ μόνο σε μία κατεύθυνση (από το Puck.js σε μια συσκευή ακρόασης), μπορούμε να έχουμε όσες συσκευές Puck να στέλνουν δεδομένα.

Ο κανονικός τρόπος λειτουργίας για το Puck.js είναι η διαφήμιση. Αυτό σημαίνει ότι μεταδίδει μερικά byte πληροφορίες κάθε δευτερόλεπτο ή παραπάνω (είναι παραμετροποιήσιμο). Δεν εμπλέκεται σε αμφίδρομη επικοινωνία.

Από προεπιλογή μεταδίδει το όνομά του και τις υπηρεσίες που υλοποιεί.

Σημείωση: Η μέθοδος `NRF.setAdvertising (...)` χρησιμοποιείται για παραδείγματα σχετικά με τον τρόπο ρύθμισης της διαφήμισης στο Puck.js. Μπορούμε επίσης να χρησιμοποιήσουμε τις μεθόδους `NRF.findDevices (...)` και `NRF.setScan (...)` για να σαρώσουμε και να ανιχνεύσουμε διαφημιστικά πακέτα που εκπέμπονται από άλλες συσκευές beacons.

Πρώτον, πρέπει να διαφημίσουμε ορισμένα στοιχεία. Αυτό μπορεί να γίνει με τη μέθοδο [NRF.setAdvertising](#).

Υπάρχουν δύο (2) κύριοι τύποι διαφημίσεων:

- **Υπηρεσίες - Services**: κάθε υπηρεσία έχει UUID υπηρεσίας. Αυτά μπορεί να είναι 16 bit ή 128 bit. Τα UUID 16 bit εκχωρούνται από το [Bluetooth SIG](#) - Bluetooth Special Interest Group, ώστε να χρειαστεί να χρησιμοποιήσετε τα σωστά. Μπορείτε να χρησιμοποιήσετε τα δικά σας UUID 128 bit εφόσον είναι τυχαία, ωστόσο οι διαφημίσεις Bluetooth LE είναι τόσο μικρές ώστε δεν υπάρχει συνήθως αρκετό περιθώριο για πολλά δεδομένα αφού χρησιμοποιήσετε UUID 128 bit. Θα μπορούσαμε, για παράδειγμα, να χρησιμοποιήσουμε το Service UUID 0xFFFF (το οποίο δεν πρέπει να χρησιμοποιηθεί στην παραγωγή) με την εντολή `NRF.setAdvertising ({0xFFFF: "Hello"});`;
- **Δεδομένα Κατασκευαστή** Από την έκδοση 1v95 και αργότερα επιτρέπεται να καθοριστούν τα δεδομένα κατασκευαστή. Αυτό είναι σχεδόν το ίδιο με τη χρήση Υπηρεσιών με UUID 16 bit, ωστόσο το Espruino / Puck.js έχει το δικό του 16 bit UUID δεσμευμένο (0x0590), το οποίο μπορούμε να χρησιμοποιήσουμε για

οποιαδήποτε χρήση θέλουμε. Αυτός είναι και ο τρόπος που εφαρμόστηκε για τη διαφήμιση και ανάγνωση των δεδομένων της συσκευής Puck.

Η συσκευή Puck.js μπορεί να ακούσει άλλες διαφημίσεις από συσκευές που βρίσκονται εντός εμβέλειας χρησιμοποιώντας είτε τη μέθοδο `NRF.setScan` (για συνεχή ακρόαση) είτε **`NRF.findDevices`** (για να ακούσει μια καθορισμένη χρονική περίοδο και να συγκεντρώσει όλες τις απαντήσεις). Για παράδειγμα στο Puck θα μπορούσαμε να εκτελέσουμε τον ακόλουθο κώδικα:

```
NRF.findDevices(print)
```

Το αποτέλεσμα με τον αριθμό των συσκευών που ανιχνεύτηκαν εντός εμβέλειας εμφανίζεται ως εξής:

```
[
  BluetoothDevice {
    "id": "de:70:d9:0c:eb:86 random",
    "rssi": -44,
    "data": new Uint8Array([2, 1, 5, 4, 255, 144, 5, 11, 20, 9, 69, 115, 112, 114, 117, 105, 110, 111, 32, 78, 82, 70, 53, 50, 56, 51, 50, 68, 75]).buffer,
    "manufacturer": 1424,
    "manufacturerData": new Uint8Array([0]).buffer,
    "name": "Puck.js eb86"
  }
]
```

Στη συνέχεια, μπορούμε να ελέγξουμε ότι ο αριθμός κατασκευαστή ισούται με `manufacturer == 0x0590` και να διαβάσουμε την τιμή του πεδίου κατασκευαστή - `manufacturerData` για να λάβουμε τα διαβιβαζόμενα δεδομένα.

Σημείωση → Η σάρωση για διαφημιστικά δεδομένα χρησιμοποιεί πολλή ισχύ, οπότε θα πρέπει να γίνεται σπάνια.

Στο παρακάτω παράδειγμα εμφανίζεται κώδικας με τον οποίο διαφημίζουμε τις φορές που έχει πατηθεί το κουμπί του Puck.

```
var presses = 0;
NRF.setAdvertising({}, {manufacturer: 0x0590, manufacturerData:[presses]});

setWatch(function() {
  presses++;
  NRF.setAdvertising({}, {manufacturer: 0x0590, manufacturerData:[presses]});
}, BTN, {edge:"rising", repeat:1, debounce:20})
```

Αυτό θα διαφημίσει ένα μόνο byte, **0**, το οποίο θα αυξάνεται κάθε φορά που πατάμε το κουμπί.

Σημείωση → Ο κατασκευαστής Data είναι μια σειρά από bytes (τιμές μεταξύ 0 και 255). Οι μεγαλύτερες τιμές θα περικοπούν. Μπορείτε επίσης να ορίσετε μια συμβολοσειρά, για παράδειγμα με το `NRF.setAdvertising({}, {manufacturer: 0x0590, manufacturerData: "Hello"})`; . Ο καθορισμός υπερβολικών δεδομένων (δηλαδή περισσότερο από 21 bytes) μπορεί να έχει ως αποτέλεσμα μια εξαίρεση **DATA_SIZE**.

3.7 Υπηρεσίες και Χαρακτηριστικά

Μόλις συνδεθεί μια κεντρική συσκευή, αποκτά πρόσβαση στις Υπηρεσίες και στα χαρακτηριστικά που έχει ένα Περιφερειακό (στην περιπτώση μας το Puck.js). Αυτό ονομάζεται GATT - το γενικό προφίλ χαρακτηριστικών, όπου και θα αναλυθεί περισσότερο παρακάτω.

Δίνουμε ονόματα υπηρεσιών και χαρακτηριστικών, αλλά ορίζονται από ένα διεθνώς μοναδικό αναγνωριστικό (UUID). Παίρνετε αυτά τα 16 κομμάτια τα οποία είναι μοναδικά επειδή το Bluetooth SIG τα ορίζει και παίρνετε 128 bit που είναι ελεύθερα στη χρήση και μοναδικά γιατί όσο βρίσκεστε τυχαία, οι πιθανότητες να είναι οι ίδιες με αυτές ενός άλλου είναι λεπτά. Οι υπηρεσίες είναι πραγματικά απλές ομάδες χαρακτηριστικών και κάθε χαρακτηριστικό αντιπροσωπεύει έναν τύπο δεδομένων.

Υπάρχουν τρεις κύριες λειτουργίες που μπορούν να εκτελεστούν με ένα χαρακτηριστικό:

→ **READ** - η κεντρική συσκευή στέλνει ένα αίτημα και το περιφερειακό αποκρίνεται με την τρέχουσα τιμή του χαρακτηριστικού.

→ **WRITE** - η κεντρική συσκευή στέλνει δεδομένα και αίτηση για εγγραφή και η περιφερική ενημερώνει την τιμή του χαρακτηριστικού (ανάλογα με τον τύπο της εγγραφής μπορεί να ανταποκριθεί λέγοντας ότι όλα είναι εντάξει)

→ **NOTIFY / INDICATE** - μια κεντρική συσκευή μπορεί να ζητήσει να λάβει ειδοποιήσεις. Στη συνέχεια, όταν η τιμή του χαρακτηριστικού στην περιφερειακή αλλαγή, θα «ωθήσει» τα δεδομένα στο κεντρικό, χωρίς να χρειάζεται να ελέγξει. Αυτός είναι ο καλύτερος τρόπος για την αποστολή δεδομένων. Για παράδειγμα, μπορεί να θέλετε να στείλετε δύο ταυτόσημες τιμές - μπορείτε να το κάνετε εύκολα με ειδοποίηση, αλλά θα ήταν δύσκολο να κάνετε με επαναλαμβανόμενες αναγνώσεις. Οι ειδοποιήσεις και οι ενδείξεις είναι πολύ παρόμοιες, αλλά όχι ταυτόσημες. Δείτε εδώ για μια εξήγηση - αλλά βασικά θα πρέπει πάντα να προσπαθήσετε και να χρησιμοποιήσετε την ειδοποίηση, μην υποδείξετε.

Σημείωση → Ένα χαρακτηριστικό μπορεί να έχει οποιοδήποτε συνδυασμό αυτών των 3 τύπων. Στην πραγματικότητα, είναι πολύ συνηθισμένο να έχετε ένα χαρακτηριστικό που δεν μπορεί να διαβαστεί, αλλά αυτό έχει ειδοποιήσει ώστε μια κεντρική συσκευή να μπορεί να ειδοποιηθεί όταν αλλάξει κάτι. Για παράδειγμα, μπορεί να έχετε: Μια υπηρεσία Light, η οποία διαθέτει 2 γράψιμα (και ίσως αναγνώσιμα) χαρακτηριστικά για τη φωτεινότητα και την απόχρωση. Μια υπηρεσία κουμπιών, η οποία έχει αναγνωρίσιμη και αναγνωρίσιμη ιδιότητα για το Button State Μια υπηρεσία κίνησης, η οποία έχει απλώς ένα χαρακτηριστικό που μπορεί να επισημανθεί και στα οποία αποστέλλονται δεδομένα κατά την κίνηση

3.8 Υπηρεσίες συσκευής Puck.js

Από προεπιλογή, το Puck.js διαθέτει την υπηρεσία Nordic UART (UUID 6e400001-b5a3-f393-e0a9-e50e24dcca9e) που μας επιτρέπει να επικοινωνήσουμε με τον διεργμηνέα JS. Αυτή η υπηρεσία προσφέρει αμφίδρομες επικοινωνίες.

Έχει δύο (2) χαρακτηριστικά → **RX** και **TX**:

Το χαρακτηριστικό TX → (UUID 6e400002-b5a3-f393-e0a9-e50e24dcca9e) σας επιτρέπει να στείλετε δεδομένα στο Puck.js. Μπορείτε να γράψετε έως και 20 bytes δεδο-

μένων σε αυτό και κάθε φορά που γράφετε, οι χαρακτήρες που στέλνετε πηγαίνουν κατευθείαν στον διεργαστή JS.

Το χαρακτηριστικό RX → (UUID 6e400003-b5a3-f393-e0a9-e50e24dcca9e) σας επιτρέπει να πάρετε τα δεδομένα πίσω από το Puck.js. Δεν μπορείτε να το διαβάσετε, αλλά μπορείτε να εγγραφείτε για να ειδοποιηθείτε και έτσι μπορείτε να λάβετε τυχόν χαρακτήρες καθώς στέλνονται.

Εάν θέλουμε για παράδειγμα να ανάψουμε ένα φως LED στο Puck, μπορούμε απλά να συνδεθούμε και να γράψουμε LED.set () και μια νέα γραμμή "/n" στο χαρακτηριστικό TX, και η εντολή θα εκτελεστεί! Αυτό λειτουργεί για τις λειτουργίες που έχουμε ορίσει προηγουμένως.

3.9 GATT Profile - Γενικό προφίλ χαρακτηριστικών

Η GATT είναι ένα ακρωνύμιο για το Γενικό Προφίλ Χαρακτηριστικών και ορίζει τον τρόπο με τον οποίο δύο συσκευές χαμηλής ενέργειας Bluetooth μεταφέρουν δεδομένα εμπρός και πίσω με τη χρήση εννοιών που ονομάζονται υπηρεσίες (services) και χαρακτηριστικά (characteristics). Χρησιμοποιεί ένα πρωτόκολλο γενικών δεδομένων που ονομάζεται Πρωτόκολλο Χαρακτηριστικών (ATT) - Attribute Protocol, το οποίο χρησιμοποιείται για την αποθήκευση Υπηρεσιών, Χαρακτηριστικών και συναφών δεδομένων σε έναν απλό πίνακα αναζήτησης χρησιμοποιώντας αναγνωριστικά 16-bit για κάθε καταχώρηση στον πίνακα.

Το πιο σημαντικό πράγμα που πρέπει να έχουμε κατά νου για το GATT και τις συνδέσεις είναι ότι οι συνδέσεις είναι αποκλειστικές. Αυτό σημαίνει ότι ένα περιφερειακό BLE μπορεί να συνδεθεί μόνο σε μία κεντρική συσκευή (κινητό τηλέφωνο, tablet κ.λ.π.) κάθε φορά. Μόλις ένα περιφερειακό συνδεθεί με μια κεντρική συσκευή, θα σταματήσει τη διαφήμιση και οι άλλες συσκευές δεν θα είναι πλέον σε θέση να το δουν ή να συνδεθούν με αυτό μέχρι να διακοπεί η υπάρχουσα σύνδεση.

Η δημιουργία μιας σύνδεσης είναι επίσης ο μόνος τρόπος για να επιτραπεί η αμφίδρομη επικοινωνία, όπου η κεντρική συσκευή μπορεί να στείλει σημαντικά δεδομένα στο περιφερειακό και αντίστροφα.

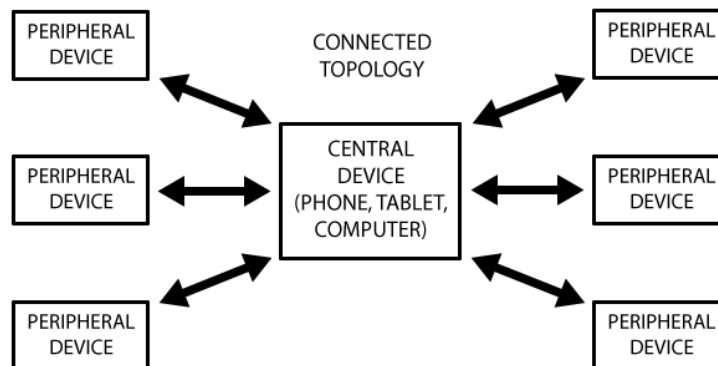
3.9.1 Τοπολογία συνδεδεμένου δικτύου - Connected Network Topology

Το παρακάτω διάγραμμα θα πρέπει να εξηγήσει τον τρόπο με τον οποίο οι συσκευές χαμηλής ενέργειας Bluetooth λειτουργούν σε ένα συνδεδεμένο περιβάλλον. Ένα περιφε-

ριακό μπορεί να συνδεθεί μόνο σε μία κεντρική συσκευή (όπως ένα κινητό τηλέφωνο) τη φορά, αλλά η κεντρική συσκευή μπορεί να συνδεθεί σε πολλαπλές περιφερειακές συσκευές.

Εάν χρειάζεται ανταλλαγή δεδομένων μεταξύ δύο περιφερειακών, θα πρέπει να εφαρμοστεί ένα προσαρμοσμένο σύστημα αποθήκευσης μηνυμάτων, όπου όλα τα μηνύματα περνούν από την κεντρική συσκευή.

Μόλις δημιουργηθεί μια σύνδεση μεταξύ περιφερειακών και κεντρικής συσκευής, η επικοινωνία μπορεί να πραγματοποιηθεί και προς τις δύο κατευθύνσεις, η οποία είναι διαφορετική από την προσέγγιση μονής κατεύθυνσης που χρησιμοποιεί μόνο διαφημιστικά δεδομένα και GAP.



Εικόνα 17: Connected Topology

Το GAP είναι ένα το ακρωνύμιο για το προφίλ γενικής πρόσβασης και ελέγχει τις συνδέσεις και τη διαφήμιση στο Bluetooth. Το GAP είναι αυτό που καθιστά τη συσκευή σας ορατή στον εξωτερικό κόσμο και καθορίζει τον τρόπο με τον οποίο δύο συσκευές μπορούν (ή δεν μπορούν) να αλληλεπιδρούν μεταξύ τους.

3.9.2 Ενέργειες GATT

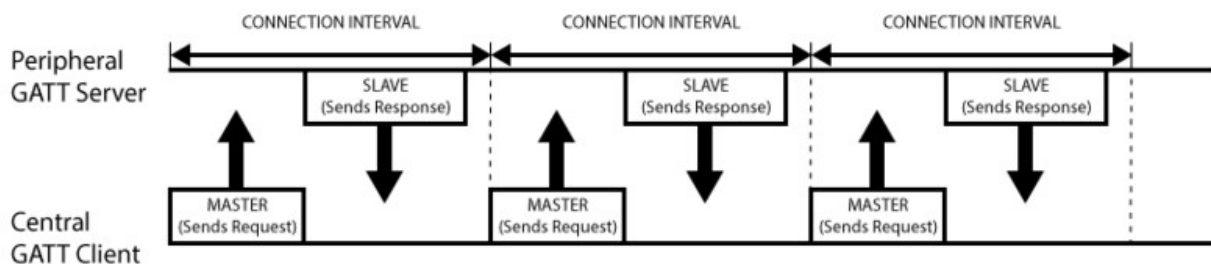
Μια σημαντική έννοια που πρέπει να κατανοηθεί με τη GATT είναι η σχέση διακομιστή / πελάτη.

Η περιφέρεια είναι γνωστή ως ο GATT Server, ο οποίος περιέχει τα δεδομένα και τις υπηρεσίες αναζήτησης και τους χαρακτηριστικούς ορισμούς και το GATT Client (το τηλέφωνο / tablet) που στέλνει αιτήσεις σε αυτόν τον διακομιστή.

Όλες οι συναλλαγές ξεκινούν από τη βασική συσκευή, τον πελάτη GATT, η οποία λαμβάνει απάντηση από τη συσκευή υποτελής, τον GATT Server.

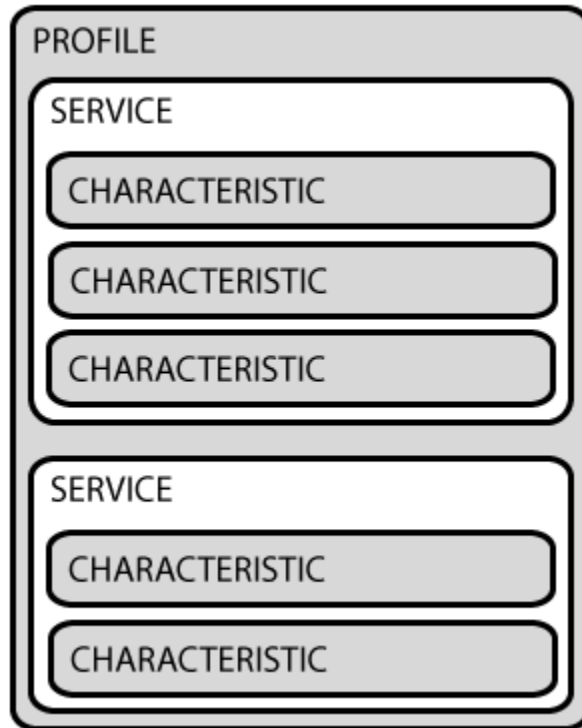
Κατά τη δημιουργία μιας σύνδεσης, το περιφερειακό θα προτείνει ένα 'Interval σύνδεσης' στην κεντρική συσκευή και η κεντρική συσκευή θα προσπαθήσει να επανασυνδέσει κάθε διάστημα σύνδεσης για να δει αν υπάρχουν νέα δεδομένα κ.λπ. Είναι σημαντικό να θυμάστε ότι αυτή η σύνδεση το διάστημα είναι πραγματικά μια πρόταση, όμως! Η κεντρική συσκευή σας μπορεί να μην είναι σε θέση να ανταποκριθεί στο αίτημα, επειδή είναι απασχολημένος να μιλάει σε άλλο περιφερειακό ή οι απαιτούμενοι πόροι του συστήματος απλά δεν είναι διαθέσιμοι.

Το ακόλουθο διάγραμμα πρέπει να απεικονίζεται στη διαδικασία ανταλλαγής δεδομένων μεταξύ ενός περιφερειακού (του GATT Server) και μιας κεντρικής συσκευής (πελάτης GATT), με την κύρια συσκευή που εκκινεί κάθε συναλλαγή:



Εικόνα 18: Send Request/Response between Central and Peripheral

Όπως αναφέρθηκε και παραπάνω οι ενέργειες του Γενικού Προφίλ Χαρακτηριστικών ανάμεσα σε συσκευές Bluetooth χαμηλής ενέργειας/BLE βασίζονται σε αντικείμενα υψηλού επιπέδου, εμφωλευμένα αντικείμενα που ονομάζονται προφίλ, υπηρεσίες και χαρακτηριστικά, τα οποία φαίνονται στην παρακάτω Εικόνα:



Εικόνα 19: Profile/Service/Characteristics

Για την επίτευξη σύνδεσης μεταξύ της κινητής συσκευής του χρήστη και της συσκευής Puck ακολουθήθηκε ο επίσημος οδηγός της Google για Android συσκευές και άλλες BLE συσκευές. Στον παρακάτω σύνδεσμο παρουσιάζονται και επεξηγούνται όλα τα παραπάνω καθώς και πολλές άλλες πληροφορίες για το πως θα πετύχουμε τις συνδέσεις που επιθυμούμε.

<https://developer.android.com/guide/topics/connectivity/bluetooth-le#java>

3.10 Προγραμματισμός συσκευής Puck

Η συσκευή Puck αποτελεί σημαντικό παράγοντα στην υλοποίηση της διπλωματικής εργασίας και χρησιμοποιήθηκε για τη συλλογή δεδομένων που λαμβάνουμε από τους ενσωματωμένους σένσορες της συσκευής.

Στην επόμενη σελίδα εμφανίζεται ο κώδικας που γράφτηκε για την επίτευξη του σκοπού της εργασίας και την σύνδεση και επικοινωνία της συσκευής Puck τόσο με την έξυπνη κινητή συσκευή του χρήστη, όσο και με τις συσκευές beacons.

Η συσκευή Puck προγραμματίστηκε έτσι ώστε να “διαφημίζει” τα δεδομένα τα οποία προκύπτουν από τους ενσωματωμένους σένσορες της κατόπιν της επικοινωνίας της με τις κοντινές συσκευές beacons. Η “διαφήμιση” γίνεται με τη μέθοδο `NRF.setAdvertising` νδιαφημίζοντας τα δεδομένα της συσκευής με κατασκευαστή `manufacturer = 0x0590(Puck)`.

Λόγω του υψηλού ποσοστού κατανάλωσης της μπαταρίας της μεθόδου ανίχνευσης για κοντινές συσκευές beacons έχει προγραμματιστεί να πραγματοποιείται κάθε 5 λεπτά για 3 δευτερόλεπτα. Εφόσον ανιχνευτεί κάποια συσκευή με την τιμή κατασκευαστή να ισούται με 76 `"manufacturer" == 76`, δηλαδή την τιμή του κατασκευαστή των IBeacons, τότε θα διαφημίσει τα δεδομένα του και αμέσως μετά θα σταματήσει να εκτελείται η μέθοδος ανίχνευσης συσκευών `NRF.findDevices`.

```
var data = Math.round(Puck.light()*100) + "-" + Math.round(E.getTemperature()) + "-" + Math.round(NRF.getBattery() * 100)/100 + "-" + myState + "-" + (dev.id).substring(0,2);
```

Ο χρήστης πατώντας το κουμπί είτε παρατεταμένα είτε απλά η συσκευή Puck θα εκτελέσει τη λειτουργία ανίχνευσης για κοντινές συσκευές beacons και εφόσον κάποια ανιχνευθεί θα διαφημίσει τα παραπάνω δεδομένα του(data).

```

var myState = 0;
function advertiseBeacon(dev) {
  ~~~
  var uuid = dev.id
  .substr(0,17).split(":").map(x=>parseInt(x,16));
  var data = Math.round(Puck.light()*100) + "-" +
  Math.round(E.getTemperature()) + "-" +
  Math.round(NRF.getBattery() * 100)/100 + "-" + myState +
  "-" + (dev.id).substring(0,2);
  NRF.setAdvertising({}, {
    manufacturer:0x590, // espruino's manufacturer
    manufacturerData:data
  });
  console.log(data);
  blueLed();
  ~~~
  myState = 0;
}
function doScan() {
  NRF.findDevices(function(devs) {
    var found = undefined;
    devs.forEach(function(dev) {
      if (dev.manufacturer==76)
        found = dev;
    });

    if (found) advertiseBeacon(found); // advertise one
we found
    else NRF.setAdvertising({}, {}); // don't advertise
any data
  }, 3000); // scan for 3 seconds
};
setInterval(doScan, 5*60*1000); // 5 minutes

```

Εικόνα 19: Κομμάτι κώδικα Puck(1ο μέρος).

Το μήνυμα που περιλαμβάνει τις πέντε (5) τιμές όπου και “διαφημίζει” το Puck θα εμφανίζεται ως εξής στο περιβάλλον χρήστη Espruino Web IDE:

Οι τιμές αυτές αφορούν ένα τυχαίο δοκιμαστικό μήνυμα:

6-24-2.77-1-f8

- Η πρώτη τιμή είναι η τιμή της φωτεινότητας (6)
- Η δεύτερη τιμή είναι η τιμή της θερμοκρασίας (24)
- Η τρίτη τιμή είναι η τιμή του ποσοστού της μπαταρίας της ίδιας της συσκευής Puck (2.77)
- Η τέταρτη τιμή είναι η τιμή κατάσταση που προκύπτει από το πάτημα του κουμπιού(παρατεταμένο πάτημα - myState = 2 και απλό πάτημα - myState = 1)

- Η πέμπτη και τελευταία τιμή είναι οι 2 πρώτοι χαρακτήρες της διεύθυνσης (mac address) της συσκευής beacon που ανίχνευσε η συσκευή Puck. (f8)

```

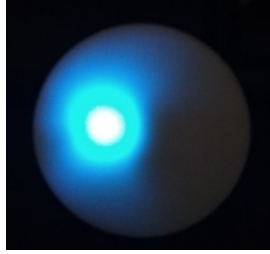
setWatch(function(e) {
  var longPress = e.time - e.lastTime;
  if(longPress <= 0.25) {
    greenLed();
    myState = 2;
    doScan();
  } else {
    myState = 1;
    doScan();
    redLed();
  }
}, BTN, {edge:"falling", repeat:true, debounce:50});

function greenLed() {
  digitalPulse(LED2,1,500);
}
function redLed() {
  digitalPulse(LED1,1,600);
}
function blueLed() {
  digitalPulse(LED3,1,600);
}

```

Εικόνα 20: Κομμάτι κώδικα Puck(2ο μέρος)

Σε αυτό το κομμάτι κώδικα χρησιμοποιείται η μέθοδος `setWatch`, με τη οποία μπορούμε να χρησιμοποιήσουμε το Puck ως κουμπί με το οποίο ο χρήστης πατώντας το να πραγματοποιηθούν κάποιες ενέργειες. Επιπλέον δίνεται η δυνατότητα του απλού πατήματος του κουμπιού και του παρατεταμένου πατήματος. Αυτό προκύπτει σύμφωνα με το χρόνο (ms) που θα χρειαστεί να αφήσει το κουμπί ο χρήστης από την ώρα που το έχει πατήσει. Στο απλό πάτημα, δηλαδή $ms \leq 0.25$ το Puck θα διαφημίσει την τιμή 2 της μεταβλητής `myState`, ενώ στην αντίθετη περίπτωση θα διαφημίσει την τιμή 1. Στο απλό πάτημα θα αναβοσβήσει το πράσινο LED, ενώ στο παρατεταμένο θα αναβοσβήσει το κόκκινο LED. Όταν η συσκευή Puck θα διαφημίσει τα δεδομένα το μπλε LED θα αναβοσβήσει.



Εικόνα 21: Αναμμένο μπλε LED - Διαφήμιση των δεδομένων Puck

3.11 Κατανάλωση Μπαταρίας - Power Consumption

Η κατανάλωση ενέργειας του Puck.js σε πολύ μεγάλο βαθμό όχι μόνο από το μέγεθος του κώδικα JavaScript που εκτελείται, αλλά πόσο πολύ μεταδίδονται δεδομένα/τιμές, πόσο συχνά και σε ποιο επίπεδο ισχύος.

Οι τιμές των επιπέδων κατανάλωσης της μπαταρίας είναι περίπου οι εξής παρακάτω:

- Το Puck δεν κάνει τίποτα - 3uA
- Μην κάνετε τίποτα, βλέποντας το κουμπί για press - 12uA
- Διαφήμιση, 375ms 0dBm (προεπιλεγμένη λειτουργία) - 20uA
- Διαφήμιση, 375ms 0dBm (προεπιλεγμένη λειτουργία), βλέποντας το κουμπί - 25uA
- Διαφήμιση, ανάγνωση μαγνητόμετρου 0.63 Hz - 50uA
- Διαφήμιση, ανάγνωση μαγνητομέτρου 10 Hz - 200uA
- Συνδεδεμένο μέσω BLE - 200uA
- Ένα LED ανάβει - 1-2mA
- 100% χρήση CPU με JavaScript - 4mA
- Όλα τα φωτάκια LED ανάβουν, η χρήση του CPU 100% που εκτελείται με JavaScript - 10mA
- Χρησιμοποιώντας το εργαλείο NRF.findDevices για τη σάρωση συσκευών - 12mA
- Το Puck.js στέλνει - “διαφημίζει” τα δεδομένα χωρίς να εκτελεί κώδικα JavaScript.

Η Nordic Semiconductors παρέχει ένα εργαλείο για την επεξεργασία της κατανάλωσης ενέργειας για τη διαφήμιση δεδομένων στον σύνδεσμο:

<https://devzone.nordicsemi.com/power/>

Κεφάλαιο 4. Firebase - Cloud Firestore

Για την αποθήκευση των δεδομένων που λαμβάνουμε μέσω των συσκευών beacons και της συσκευής Puck.js χρησιμοποιήθηκε η βάση δεδομένων Cloud Firestore της Firebase.

Το Cloud Firestore είναι μια ευέλικτη, κλιμακούμενη βάση δεδομένων NoSQL που φιλοξενείται σε “νέφος” - cloud για την ανάπτυξη κινητών, web και διακομιστών από το Firebase και το Google Cloud Platform. Όπως και η βάση δεδομένων Firebase Realtime, διατηρεί τα δεδομένα σε συγχρονισμό με εφαρμογές πελάτη μέσω ακροατών σε πραγματικό χρόνο και προσφέρει υποστήριξη χωρίς σύνδεση για κινητά και ιστούς, ώστε να μπορούμε να δημιουργούμε εφαρμογές με δυνατότητα απόκρισης που λειτουργούν ανεξάρτητα από την καθυστέρηση του δικτύου ή τη σύνδεση στο Internet. Το Cloud Firestore προσφέρει επίσης απρόσκοπτη ολοκλήρωση με άλλα προϊόντα Firebase και Google Cloud Platform, συμπεριλαμβανομένων των λειτουργιών Cloud.



Εικόνα 22: Cloud Firestore

Μετά το μοντέλο δεδομένων NoSQL του Cloud Firestore, τα δεδομένα αποθηκεύονται σε έγγραφα (documents) που περιέχουν πεδία τα οποία αντιστοιχούν σε τιμές. Αυτά τα έγγραφα αποθηκεύονται σε συλλογές (collections), οι οποίες είναι δοχεία για τα έγγραφά σας, τα οποία μπορείτε να χρησιμοποιήσετε για να οργανώσετε τα δεδομένα σας και να δημιουργήσετε ερωτήμα-



τα. Τα έγγραφα υποστηρίζουν πολλούς διαφορετικούς τύπους δεδομένων, από απλές λέξεις και αριθμούς, έως σύνθετα, ένθετα αντικείμενα, συλλογές κ.λπ. Μπορείτε επίσης να δημιουργήσετε δευτερεύουσες συλλογές εγγράφων και να δημιουργήσετε ιεραρχικές δομές δεδομένων που κλιμακώνονται όσο αυξάνεται η βάση δεδομένων μας. Το μοντέλο δεδομένων Cloud Firestore υποστηρίζει όποια δομή δεδομένων λειτουργεί καλύτερα για την εφαρμογή σας.

Επιπλέον, η αναζήτηση στο Cloud Firestore είναι εκφραστική, αποτελεσματική και ευέλικτη. Μπορούμε να δημιουργήσουμε γρήγορα και εύκολα υπερωτήματα στη βάση μας για την ανάκτηση δεδομένων σε επίπεδο εγγράφου χωρίς να χρειαστεί να ανακτήσουμε ολόκληρη τη συλλογή ή οποιοσδήποτε ένθετος δευτερεύουσες συλλογές. Επιπλέον μπορούμε να προσθέσουμε την ταξινόμηση, το φιλτράρισμα και τα όρια(limit) στα ερωτήματα ή τους δρομείς μας για να προβάλετε τα αποτελέσματά σας. Για να διατηρήσετε τρέχοντα δεδομένα στις εφαρμογές σας, χωρίς να ανακτάτε ολόκληρη τη βάση δεδομένων σας κάθε φορά που γίνεται ενημέρωση, προσθέστε ακροατές σε πραγματικό χρόνο. Η προσθήκη ακροατών σε πραγματικό χρόνο στην εφαρμογή σας σας ενημερώνει με ένα στιγμιότυπο δεδομένων όποτε τα δεδομένα που οι εφαρμογές των πελατών σας ακούνε αλλαγές, ανακτώντας μόνο τις νέες αλλαγές.

Ως βασική λειτουργία η Cloud Firestore παρέχει στην προστασία κατά την πρόσβαση στα δεδομένα μας με το Firebase Authentication και Cloud Firestore Security Rules για Android, iOS και JavaScript ή Διαχείριση ταυτότητας και πρόσβασης (IAM) για τις γλώσσες του διακομιστή.

4.1 Μοντέλο δεδομένων Cloud Firestore Data

Το Cloud Firestore είναι μια βάση δεδομένων που βασίζεται σε έγγραφα. Σε αντίθεση με μια βάση δεδομένων SQL, δεν υπάρχουν πίνακες ή γραμμές. Αντίθετα, αποθηκεύετε δεδομένα σε έγγραφα που είναι οργανωμένα σε συλλογές.

Κάθε έγγραφο περιέχει ένα σύνολο ζευγών κλειδιών-τιμών. Το Cloud Firestore είναι βελτιστοποιημένο για την αποθήκευση μεγάλων συλλογών μικρών εγγράφων.

Όλα τα έγγραφα πρέπει να αποθηκεύονται σε συλλογές. Τα έγγραφα μπορούν να περιέχουν δευτερεύουσες συλλογές και ένθετα αντικείμενα, τα οποία μπορεί να περιλαμβάνουν πρωτόγονα πεδία όπως χορδές ή σύνθετα αντικείμενα όπως λίστες.

Οι συλλογές και τα έγγραφα δημιουργούνται σιωπηρά στο Cloud Firestore. Απλά αναθέστε δεδομένα σε ένα έγγραφο μέσα σε μια συλλογή. Αν η συλλογή ή το έγγραφο δεν υπάρχει, το Cloud Firestore το δημιουργεί.

4.1.1 Έγγραφα - Documents

Στο Cloud Firestore, η μονάδα αποθήκευσης είναι το έγγραφο. Ένα έγγραφο είναι μια ελαφριά έγγραφη που περιέχει πεδία, τα οποία αντιστοιχούν σε τιμές. Κάθε έγγραφο αναγνωρίζεται από ένα όνομα.

Ένα έγγραφο που αναπαριστά μια κατανομή πεδίων του χρήστη μπορεί να μοιάζει με:



```
"User"
```

```
first: "Ada"
last: "Lovelace"
userName: "Love"
```

Σημείωση → Το Cloud Firestore υποστηρίζει διάφορους τύπους δεδομένων για τιμές: boolean, αριθμός, συμβολοσειρά, γεωγραφικό σημείο, δυαδικό blob και timestamp. Μπορούμε επίσης να χρησιμοποιήσουμε συστοιχίες ή ένθετα αντικείμενα, που ονομάζονται χάρτες, για τη δομή δεδομένων μέσα σε ένα έγγραφο.

Τα σύνθετα, ένθετα αντικείμενα σε ένα έγγραφο ονομάζονται χάρτες. Για παράδειγμα, θα μπορούσατε να διαρθρώσετε το όνομα του χρήστη από το παραπάνω παράδειγμα με ένα χάρτη, όπως αυτό:



```
"User"
```

```
name :
  first : "Ada"
  last : "Lovelace"
  userName : "Love"
born : 1990
```

Παρατηρείται ότι τα έγγραφα μοιάζουν πολύ με το JSON. Στην πραγματικότητα, βασικά είναι. Υπάρχουν ορισμένες διαφορές (για παράδειγμα, τα έγγραφα υποστηρίζουν επιπλέον τύπους δεδομένων και έχουν περιορισμένο μέγεθος έως 1 MB), αλλά γενικά μπορούμε να επεξεργάζομαστε τα έγγραφα ως ελαφριά αρχεία JSON.

4.1.2 Συλλογές - Collections

Τα έγγραφα διαμένουν σε συλλογές, οι οποίες είναι απλά δοχεία για έγγραφα. Για παράδειγμα, θα μπορούσατε να έχετε μια συλλογή χρηστών που να περιέχει τους διάφορους χρήστες σας, εκ των οποίων κάθε ένα αντιπροσωπεύεται από ένα έγγραφο.



"Users"



alovelace

```
first : "Ada"
last  : "Lovelace"
born  : 1990
```



aturing

```
first : "Alan"
last  : "Turing"
born  : 2000
```

Το Cloud Firestore δεν έχει Εικόνα:, επομένως έχουμε πλήρη ελευθερία σε σχέση με τα πεδία που τοποθετούμε σε κάθε έγγραφο και ποιους τύπους δεδομένων αποθηκεύουμε σε αυτούς τους τομείς. Τα έγγραφα μέσα στην ίδια συλλογή μπορούν να περιέχουν διαφορετικά πεδία ή να αποθηκεύουν διαφορετικούς τύπους δεδομένων σε αυτά τα πεδία. Ωστόσο, είναι καλή ιδέα να χρησιμοποιήσετε τα ίδια πεδία και τύπους δεδομένων σε πολλά έγγραφα, ώστε να μπορείτε να αναζητήσετε τα έγγραφα πιο εύκολα.

Μια συλλογή περιέχει έγγραφα και τίποτα άλλο. Δεν μπορεί να περιέχει άμεσα ακατέργαστα πεδία με τιμές και δεν μπορεί να περιέχει άλλες συλλογές. (Δείτε Ιεραρχικά Δεδομένα για μια εξήγηση για τον τρόπο δομής πιο σύνθετων δεδομένων στο Cloud Firestore.)

Τα ονόματα των εγγράφων μέσα σε μια συλλογή είναι μοναδικά. Μπορείτε να δώσετε τα δικά σας κλειδιά, όπως τα αναγνωριστικά χρήστη, ή μπορείτε να αφήσετε το Cloud Firestore να δημιουργήσει τυχαία αναγνωριστικά για εσάς αυτόματα (για παράδειγμα, κλώνοντας το `add()`).

Δεν είναι αναγκαίο να δημιουργούμε ή να διαγράφουμε συλλογές. Αφού δημιουργήσετε το πρώτο έγγραφο σε μια συλλογή, υπάρχει η συλλογή. Εάν διαγράψετε όλα τα έγγραφα μιας συλλογής, δεν υπάρχει πλέον.

4.2 Βασικές Δυνατότητες Cloud Firestore

- ✓ **Flexibility** → Το μοντέλο δεδομένων Cloud Firestore υποστηρίζει ευέλικτες, ιεραρχικές δομές δεδομένων. Αποθηκεύστε τα δεδομένα σας σε έγγραφα, οργα-

νωμένα σε συλλογές. Τα έγγραφα μπορούν να περιέχουν πολύπλοκα ένθετα αντικείμενα εκτός από τις δευτερεύουσες συλλογές.

- ✓ **Expressive querying** → Στο Cloud Firestore, μπορείτε να χρησιμοποιήσετε ερωτήματα για να ανακτήσετε μεμονωμένα συγκεκριμένα έγγραφα ή για να ανακτήσετε όλα τα έγγραφα σε μια συνάθροιση που ταιριάζει με τις παραμέτρους του ερωτήματος σας. Τα ερωτήματά σας μπορούν να περιλαμβάνουν πολλά, αλυσιδωτά φίλτρα και να συνδυάζουν το φιλτράρισμα και τη διαλογή. Έχουν επίσης δυνατότητα ευρετηρίου(index) από προεπιλογή, οπότε η απόδοση ερωτήματος είναι ανάλογη με το μέγεθος του αποτελέσματος που έχετε επιλέξει, και όχι με το σύνολο δεδομένων.
- ✓ **Realtime updates** → Όπως και η βάση δεδομένων πραγματικού χρόνου, το Cloud Firestore χρησιμοποιεί συγχρονισμό δεδομένων για την ενημέρωση δεδομένων σε οποιαδήποτε συνδεδεμένη συσκευή. Ωστόσο, έχει επίσης σχεδιαστεί για να παράγει αποτελεσματικά απλά, μοναδικά ερωτήματα λήψης.
- ✓ **Offline support** → Το Cloud Firestore αποθηκεύει δεδομένα που χρησιμοποιεί η εφαρμογή σας ενεργά, έτσι ώστε η εφαρμογή να μπορεί να γράφει, να διαβάζει, να ακούει και να ερωτά δεδομένα ακόμη και αν η συσκευή είναι εκτός σύνδεσης. Όταν η συσκευή επανέλθει στο διαδίκτυο, το Cloud Firestore syn-chronizes τυχόν τοπικές αλλαγές πίσω στο Cloud Firestore.
- ✓ **Designed to scale** → Το Cloud Firestore σας προσφέρει τις καλύτερες δυνατότητες της ισχυρής υποδομής του Google Cloud Platform: αυτόματη αναπαραγωγή δεδομένων πολλών περιοχών, εγγυήσεις ισχυρής συνέπειας, λειτουργίες ατομικής παρτίδας και πραγματική υποστήριξη συναλλαγών. Σχεδιάσαμε το Cloud Firestore να χειρίζεται τα πιο σκληρά φορτία βάσης δεδομένων από τις μεγάλες εφαρμογές του κόσμου.

Κάθε αίτημα βάσης δεδομένων από μια βιβλιοθήκη πελάτη Cloud Firestore για κινητά/προγράμματα web εκτιμάται με βάση τους κανόνες ασφαλείας πριν από την ανάγνωση ή την εγγραφή οποιωνδήποτε δεδομένων. Εάν οι κανόνες αποκλείουν την πρόσβαση σε οποιαδήποτε από τις καθορισμένες διαδρομές εγγράφων, αποτυγχάνεται ολόκληρο το αίτημα. Παρακάτω παρατίθενται ορισμένα παραδείγματα βασικών συνόλων κανόνων. Παρόλο που αυτοί οι κανόνες είναι έγκυροι, δεν συνιστώνται για εφαρμογές παραγωγής:

* Να επιτρέπεται η πρόσβαση ανάγνωσης/εγγραφής σε όλα τα έγγραφα σε οποιονδήποτε χρήστη έχει συνδεθεί στην εφαρμογή.

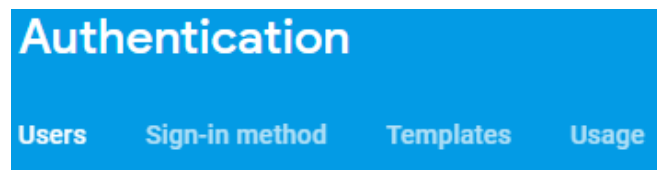
AUTH REQUIRED

```
// Allow read/write access on all documents to any user signed in to the application
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth.uid != null;
    }
  }
}
```

Εικόνα 23: Κανόνες Βασης Δεδομένων Cloud Firestore

Η **{document=**}** διαδρομή που χρησιμοποιείται στο παραπάνω παράδειγμα κανόνων αντιστοιχεί σε οποιοδήποτε έγγραφο σε ολόκληρη τη βάση δεδομένων.

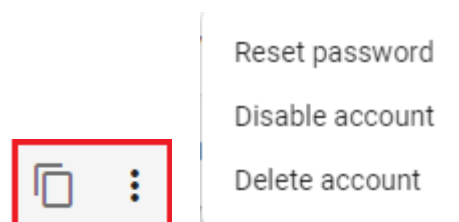
Το Cloud Firestore υποστηρίζει πληθώρα επιλογών και λειτουργιών στους χρήστες που πρόκειται να το χρησιμοποιήσουν στην εκάστοτε εφαρμογή τους. Μια βασική δυνατότητα είναι αυτή της επιλογής των στοιχείων αυθεντικοποίησης του χρήστη και η εύκολη και γρήγορη διαγραφή ή απενεργοποίηση κάποιου λογαριασμού χρήστη από την πλατφόρμα της Cloud Firestore.



Εικόνα 24: Καρτέλα Authentication Cloud Firestore

Identifier	Providers	Created	Signed In	User UID ↑
diamond@gmail.com		10 Sep 2018	1 Oct 2018	6VbQE15bcNUAqecDnnsQXjCKffr1

Εικόνα 25: Λίστα λογαριασμών χρηστών











Αν θέλουμε να διαγράψουμε, να απενεργοποιήσουμε ή να επιτρέψουμε στον χρήστη να αλλάξει των κωδικό πρόσβασης του μπορούμε να πατήσουμε δεξιά από τα στοιχεία του χρήστη και να εμφανιστεί το παραπάνω μενού.

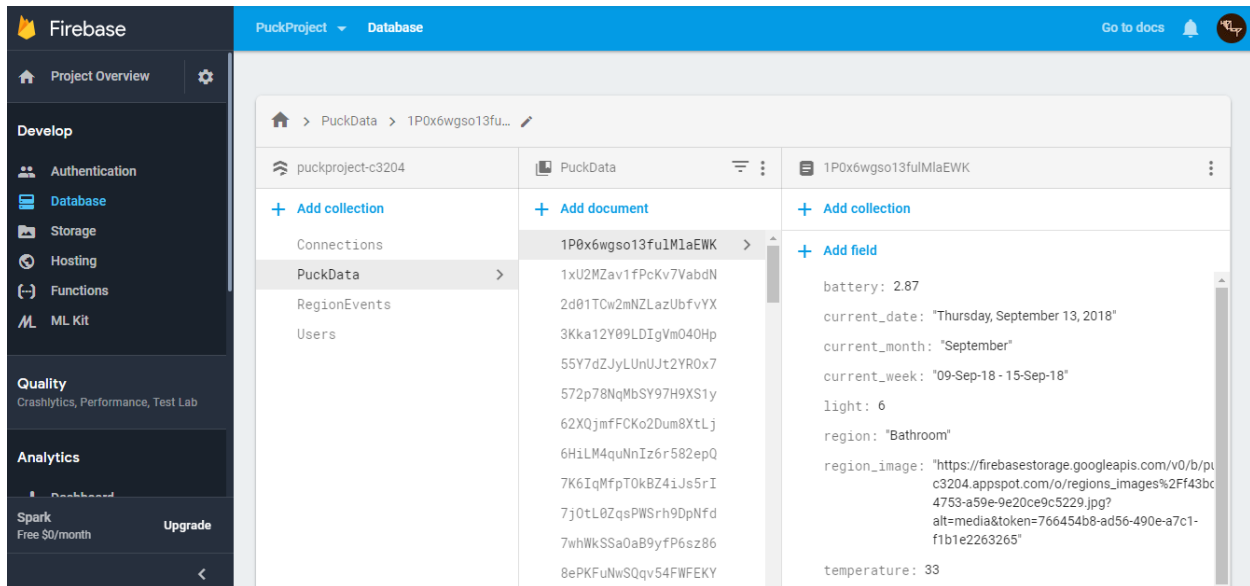
Sign-in method

4.3 Επιλογή τρόπου σύνδεσης του χρήστη στην Εφαρμογή

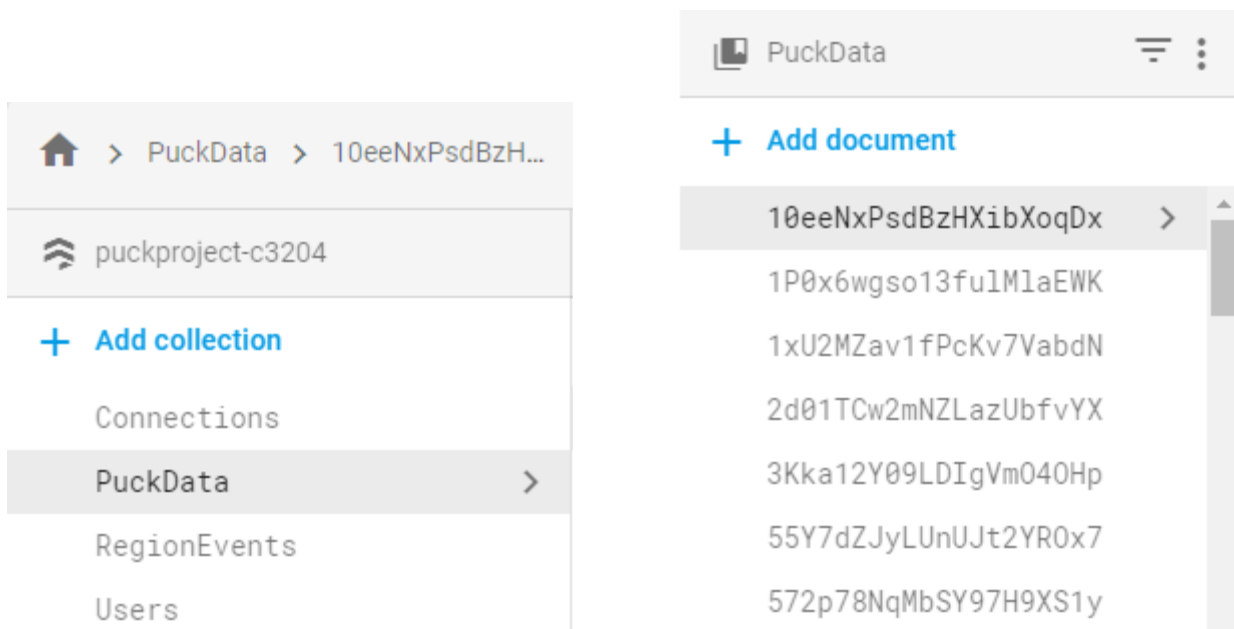
Ο επιλεγμένος τρόπος για την πραγματοποίηση σύνδεσης των χρηστών με την εφαρμογή είναι αυτός του συνδυασμού e-mail και password. Έπισης, έχουμε τη δυνατότητα για σύνδεση των χρηστών με το Google Account τους ή το Facebook, ακόμα και με το Github Account.

Sign-in providers	
Provider	Status
 Email/Password	Enabled
 Phone	Disabled
 Google	Disabled
 Play Games	Disabled
 Facebook	Disabled
 Twitter	Disabled
 GitHub	Disabled
 Anonymous	Disabled

Εικόνα 26: Πάροχοι Σύνδεσης στην Εφαρμογή



Εικόνα 27: Περιβάλλον Cloud Firestore



Εικόνα 28: Συλλογή PuckData

Εικόνα 29: Έγγραφο της συλλογής PuckData

Παραπάνω φαίνονται όλες οι βασικές συλλογές που έχουν δημιουργηθεί στη βάση δεδομένων Cloud Firestore, καθώς και τα έγγραφα που ανήκουν σε μία απ' αυτές. (**Puck-Data** collection)

Κεφάλαιο 5. Προδιαγραφές και Αρχιτεκτονική Συστήματος

5.1 Προδιαγραφές Συστήματος

Πριν γίνει αναφορά στην αρχιτεκτονική της εφαρμογής *Activities of Daily Living*, δηλαδή στα συστατικά μέρη που την αποτελούν και πως αυτά συνδέονται και επικοινωνούν μεταξύ τους, στην παράγραφο αυτή θα γίνει μια σύντομη περιγραφή των προδιαγραφών λειτουργίας του συστήματος. Συγκεκριμένα, θα αναλυθεί ποιος είναι ο χρήστης της εφαρμογής, ποια τα χαρακτηριστικά του και ποιες οι απαιτήσεις του από αυτή.

Στην παρακάτω λίστα παρουσιάζονται τα βασικά χαρακτηριστικά του χρήστη:

- Ο χρήστης να διαθέτει τις “έξυνες” συσκευές *beacons*, ώστε να τις τοποθετήσει στα σημεία που επιθυμεί και στη συνέχεια να καθορίσει τις απαραίτητες περιοχές για την ανίχνευση της τοποθεσίας του σε εσωτερικούς χώρους της κατοικίας του.
- Ο χρήστης θα πρέπει να διαθέτει κοντά του ή πάνω του τη συσκευή *Puck* με την οποία θα ανιχνεύονται κοντινές συσκευές *beacons* και με την οποία θα μπορεί να πατήσει το κουμπί, ώστε να δηλώσει την κατάστασή του.
- Να μπορεί να συνδεθεί μία και μόνο φορά με τη συσκευή *Puck* και αφού γίνει αυτό κάθε φορά που η κινητή του συσκευή βρίσκεται εντός εμβέλειας με τη συσκευή *Puck.js* να συνδέονται αυτόματα, χωρίς την παρέμβαση του χρήστη.
- Να μπορεί να ενημερώνεται για την κατάσταση της μπαταρίας της συσκευής *Puck* έτσι ώστε να μην διακοπεί η λειτουργία της λόγω έλλειψης επαρκούς μπαταρίας.

Με βάση τα παραπάνω χαρακτηριστικά, ο χρήστης έχει ορισμένες απαιτήσεις από την εφαρμογή *Activities of Daily Living*.

Οι απαιτήσεις, λοιπόν, που έχει ο χρήστης από το σύστημα είναι οι εξής:

- Να μπορεί να κάνει εγγραφή και να δημιουργήσει προσωπικό λογαριασμό, καθώς και να μπορεί να επεξεργαστεί το προφίλ του.
- Να μπορεί να διασφαλίσει το απόρρητο του λογαριασμού, ώστε μόνο αυτός να έχει πρόσβαση.
- Να μπορεί, στην συνέχεια, να αποσυνδέσει το προφίλ του αυτό από το σύστημα, σε περίπτωση που σταματήσει να το χρησιμοποιεί.

- Να έχει πρόσβαση σε μια εύχρηστη και αισθητικά ελκυστική διεπαφή, μέσω της οποίας να μπορεί να παρακολουθεί τις καθημερινές του δραστηριότητες, καθώς και να μπορεί να κρατήσει σημαντικές σημειώσεις σχετικά με τις περιοχές.
- Καταχώρηση χ αριθμού ηλεκτρονικών διευθύνσεων συγγενικών ή φιλικών του προσώπων, ώστε να μπορέσει να τους ειδοποιήσει μέσω ηλεκτρονικού μηνύματος.
- Να έχει τη δυνατότητα να διαγράψει κάποια από τις περιοχές ή και όλες και να μπορεί να δηλώσει εκ νέου τις περιοχές έτσι όπως εκείνος επιθυμεί.

Κεφάλαιο 6. Υλοποίηση Συστήματος

6.1 Αρχική Οθόνη - Welcome Screen



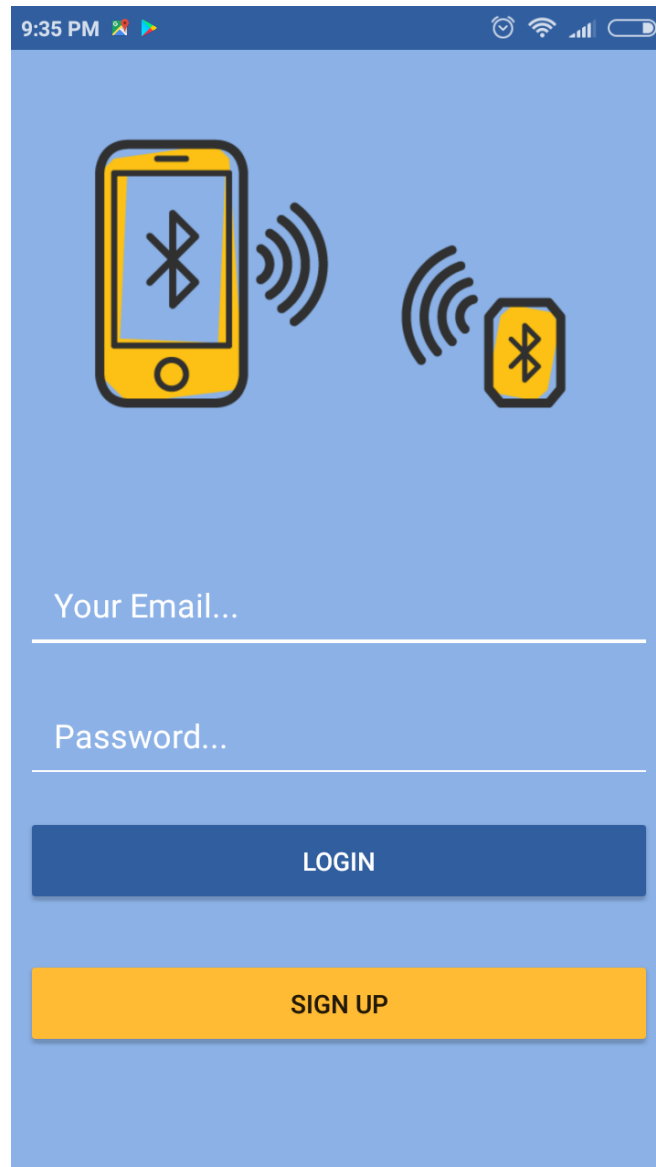
Εικόνα 30: Σύνδεσης Χρήστη στην Εφαρμογή - LoginActivity

Ο χρήστης μόλις πατήσει το εικονίδιο της εφαρμογής θα αντικρίσει την αρχική οθόνη (welcome screen) στην οποία παραμένει για δύο δευτερόλεπτα και μετά οδηγείται στην κεντρική οθόνη της εφαρμογής αν έχει συνδεθεί ήδη ή στην οθόνη σύνδεσης με την εφαρμογή αν δεν είναι ήδη συνδεδεμένος σε αυτή.

6.2 Διεπαφή Σύνδεσης Χρήστη στην Εφαρμογή - LoginActivity

Ο χρήστης καλείται να συμπληρώσει το e-mail του και τον κωδικό πρόσβασης(password) που εισήγαγε κατά τη διαδικασία εγγραφής του στην εφαρμογή.

Για την επίτευξη της σύνδεσης του χρήστη χρησιμοποιείται η μέθοδος της Cloud Firestore [signInWithEmailAndPassword](#) με την οποία ζητούνται τα 2 παραπάνω πεδία.



Εικόνα 31: Διεπαφή Σύνδεσης Χρήστη στην Εφαρμογή - LoginActivity

```

mAuth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                // Sign in success, update UI with the signed-in user's information
                Log.d(TAG, "signInWithEmail:success");
                FirebaseUser user = mAuth.getCurrentUser();
                updateUI(user);
            } else {
                // If sign in fails, display a message to the user.
                Log.w(TAG, "signInWithEmail:failure", task.getException());
                Toast.makeText(EmailPasswordActivity.this, "Authentication failed.",
                    Toast.LENGTH_SHORT).show();
                updateUI(null);
            }

            // ...
        }
    });

```

Εικόνα 32: Μέθοδος signInWithEmailAndPassword της Cloud Firestore

Στην περίπτωση που ο χρήστης κατά τη διαδικασία εγγραφής του εισάγει το e-mail του σε μη έγκυρη μορφή τότε θα εμφανιστεί στο κάτω μέρος της οθόνης του το μήνυμα:

Error: The email address is badly formatted.

Στην περίπτωση που ο χρήστης κατά τη διαδικασία εγγραφής του εισάγει μη έγκυρο κωδικό πρόσβασης τότε θα εμφανιστεί στο κάτω μέρος της οθόνης του το μήνυμα:

Error: The password is invalid or the user does not have a password.

Ο χρήστης μπορεί όποτε θέλει να αποσυνδεθεί από την εφαρμογή πατώντας την επιλογή Logout στο κεντρικό μενού επάνω δεξιά της οθόνης του. Η λειτουργία παρέχεται από την Cloud Firestore όπως φαίνεται παρακάτω.

```

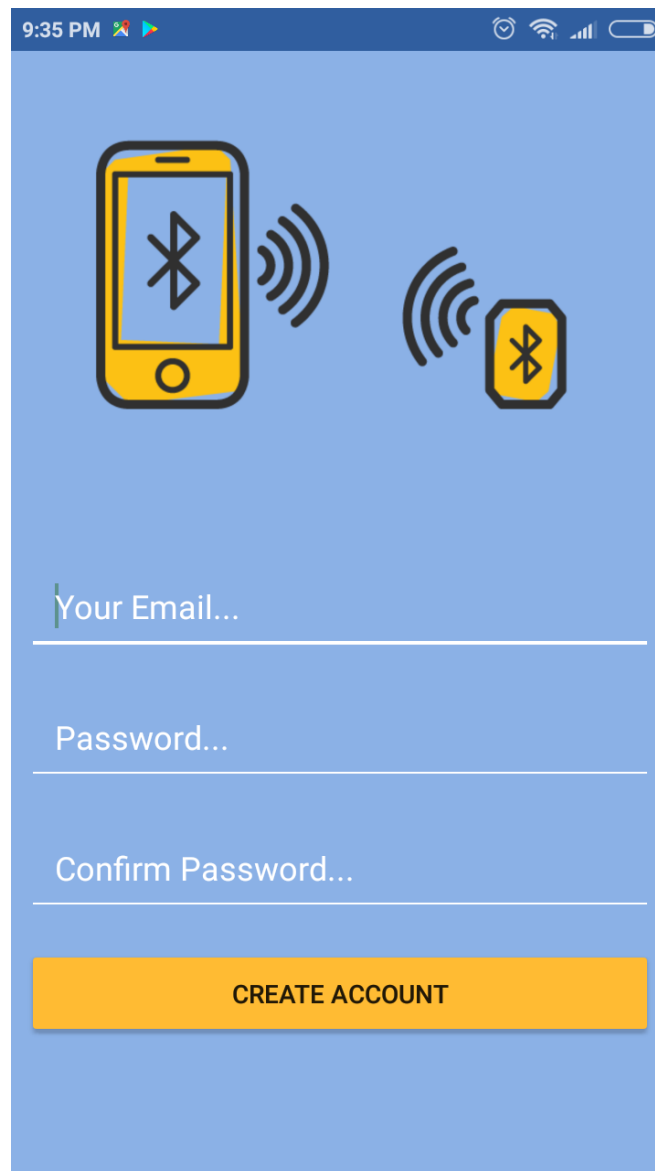
FirebaseAuth.getInstance().signOut();

```

Ο χρήστης θα οδηγηθεί στη διεπαφή σύνδεσης με την εφαρμογή αφού αποσυνδεθεί.

6.3 Διεπαφή Εγγραφής Νέου Χρήστη – RegisterActivity

Σε αυτή τη διεπαφή ο χρήστης καλείται να εισάγει το προσωπικό του email και τον κωδικό πρόσβασης(x2) στις αντίστοιχες περιοχές των υποχρεωτικών πεδίων που επιθυμεί ώστε να δημιουργεί ο λογαριασμός του στη βάση δεδομένων της Cloud Firestore.



Εικόνα 33: Οθόνη Εγγραφής Χρήστη/Δημιουργίας Λογαριασμού

Όπως αναφέρθηκε και πρωτίστως σημαντική λειτουργία που παρέχεται από την πλατφόρμα της Firestore είναι αυτή της αυθεντικοποίησης και των μηνυμάτων που εμφανίζο-

νται, αν σε περίπτωση που ο χρήστης κατά την πληκτρολόγηση των προσωπικών του στοιχείων είτε κατά τη σύνδεση του είτε κατά την εγγραφή του παράλείψει να συμπληρώσει κάποιο πεδίο ή δώσει λανθασμένα στοιχεία.

Στην περίπτωση που ο χρήστης κατά τη διαδικασία εγγραφής του εισάγει μη έγκυρο e-mail τότε θα εμφανιστεί στο κάτω μέρος της οθόνης του το μήνυμα:

Error: The email address is badly formatted.

Στην περίπτωση που ο χρήστης κατά τη διαδικασία εγγραφής του εισάγει διαφορετικούς κωδικούς πρόσβασης στα πεδία "Password" και "Confirm Password" τότε θα εμφανιστεί στο κάτω μέρος της οθόνης του το μήνυμα:

Passwords don't match..

Στην περίπτωση που ο χρήστης κατά τη διαδικασία εγγραφής του εισάγει κωδικό πρόσβασης με λιγότερο από 6 χαρακτήρες τότε θα εμφανιστεί στο κάτω μέρος της οθόνης του το μήνυμα:

Error: The given password is invalid. [Password should be at least 6 characters]

Στην παρακάτω Εικόνα φαίνεται η μέθοδος εγγραφής νέου χρήστη της Cloud Firestore.

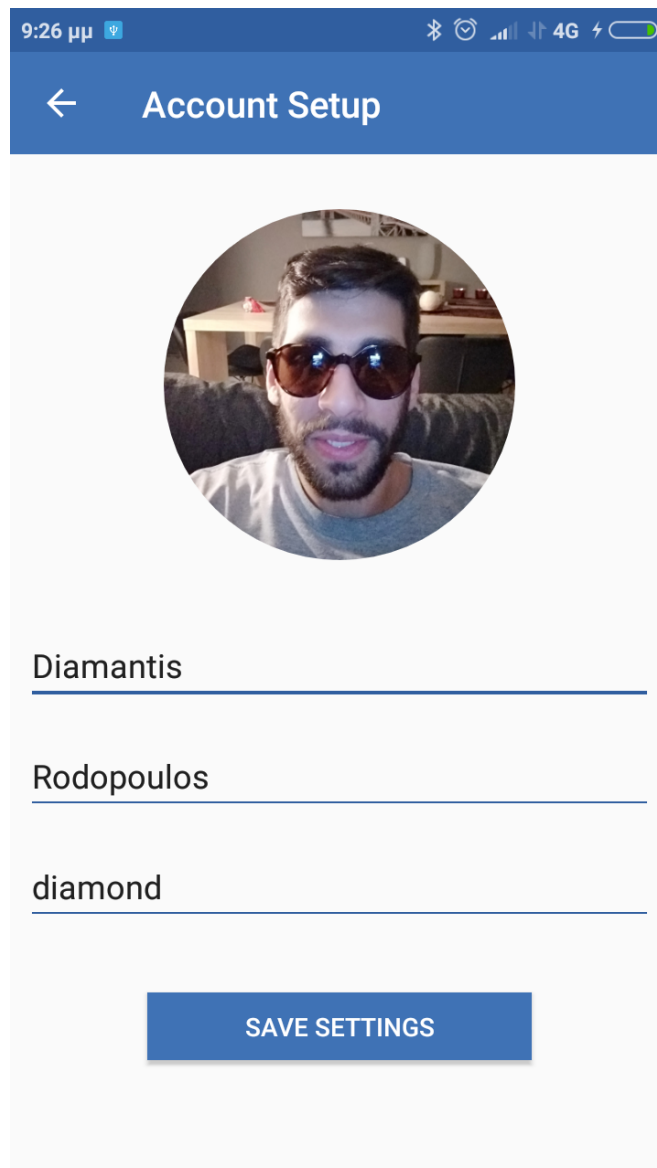
```
mAuth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                // Sign in success, update UI with the signed-in user's information
                Log.d(TAG, "createUserWithEmail:success");
                FirebaseUser user = mAuth.getCurrentUser();
                updateUI(user);
            } else {
                // If sign in fails, display a message to the user.
                Log.w(TAG, "createUserWithEmail:failure", task.getException());
                Toast.makeText(EmailPasswordActivity.this, "Authentication failed.",
                    Toast.LENGTH_SHORT).show();
                updateUI(null);
            }

            // ...
        }
    });
```

Εικόνα 34: Μέθοδος Εγγραφής Νέου Χρήστη της Cloud Firestore

6.4 Διεπαφή Καταχώρησης Προσωπικών Στοιχείων Χρήστη

Αφού ολοκληρωθεί η διαδικασία εγγραφής του χρήστη θα οδηγηθεί στην παραπάνω διεπαφή(Account Setup) καταχώρησης περεταίρω προσωπικών του στοιχείων. Τα στοιχεία που θα του ζητηθεί από την εφαρμογή να συμπληρώσει είναι: το όνομα, το επώνυμο του, ένα όνομα χρήστη καθώς και μια φωτογραφία του. Εφόσον όλα τα πεδία είναι συμπληρωμένα θα πρέπει να πατήσει το κουμπί “Save Settings”.



9:26 μμ

← Account Setup

Diamantis

Rodopoulos

diamond

SAVE SETTINGS

Εικόνα 35: Διεπαφή Καταχώρησης Προσωπικών Στοιχείων Χρήστη

Σε αυτό το σημείο στη βάση μας θα έχει δημιουργηθεί η συλλογή(collection) με το όνομα "Users" και μέσα σε αυτήν θα υπάρχει το τυχαίο ID (auto id) του χρήστη ως ένα από τα έγγραφα(document), το οποίο παράγεται μέσω της Cloud Firestore.

Τα στοιχεία αυτά που μόλις καταχώρησε ο χρήστης μπορεί ανά πάσα στιγμή να τα αλλάξει. (κυρίως αναφερόμαστε στη φωτογραφία και το username)

Εφόσον πραγματοποιήσει αυτό το βήμα μπορεί να πλοηγηθεί στο ευχάριστο περιβάλλον που προσφέρει η εφαρμογή και να ανακαλύψει όλες τις λειτουργίες της, οι οποίες αναφέρονται παρακάτω αναλυτικά.

```
Map<String, String> userMap = new HashMap<>();
userMap.put("firstname", user_fname);
userMap.put("lastname", user_lname);
userMap.put("username", user_uname);
userMap.put("image", download_uri.toString());

firebaseFirestore.collection("Users").document(userId).set(userMap).addOnCompleteListener(new OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {

        if(task.isSuccessful()) {

            Toast.makeText(SetupActivity.this, "Account Settings Saved",
Toast.LENGTH_SHORT).show();
            Intent mainIntent = new Intent(SetupActivity.this, MainActivity.class);
            startActivity(mainIntent);
            finish();

        } else {

            String error = task.getException().getMessage();
            Toast.makeText(SetupActivity.this, "(FIRESTORE Error) : " + error,
Toast.LENGTH_LONG).show();

        }

        setupProgress.setVisibility(View.INVISIBLE);

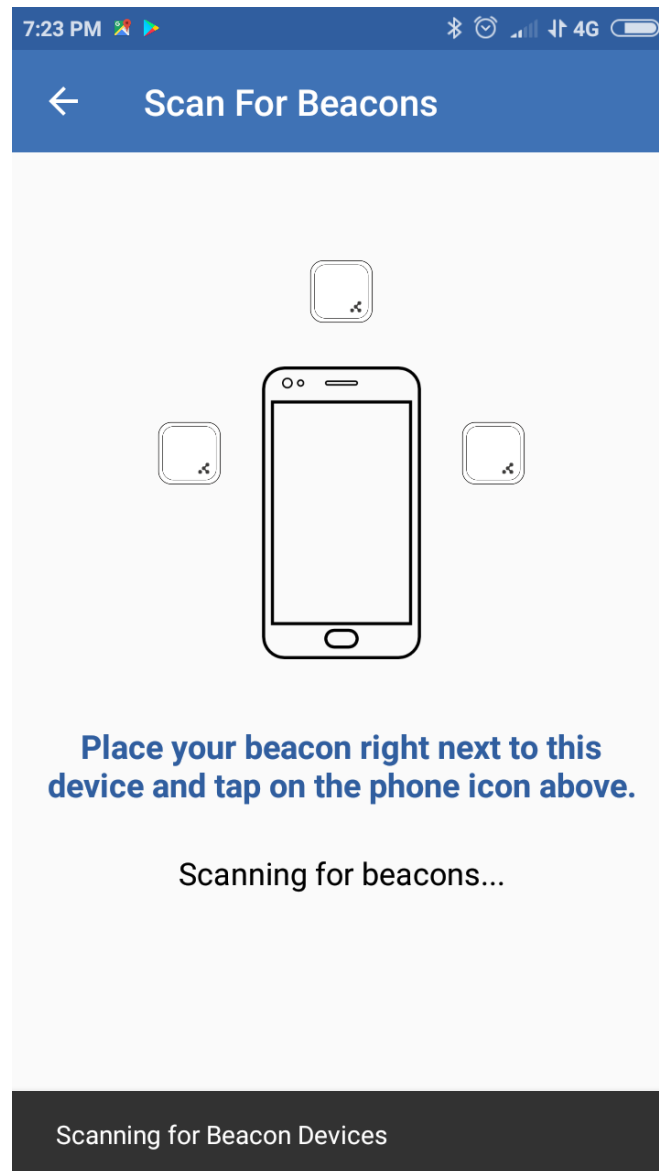
    }
});
```

Με τον παραπάνω κώδικα δημιουργείται η συλλογή «Users» και το έγγραφο-document με ID αυτό του συνδεδεμένου χρήστη και τα πεδία που αναφέρθηκαν παραπάνω.



Εικόνα 36: Έγγραφο με το ID χρήστη και στοιχεία αυτού

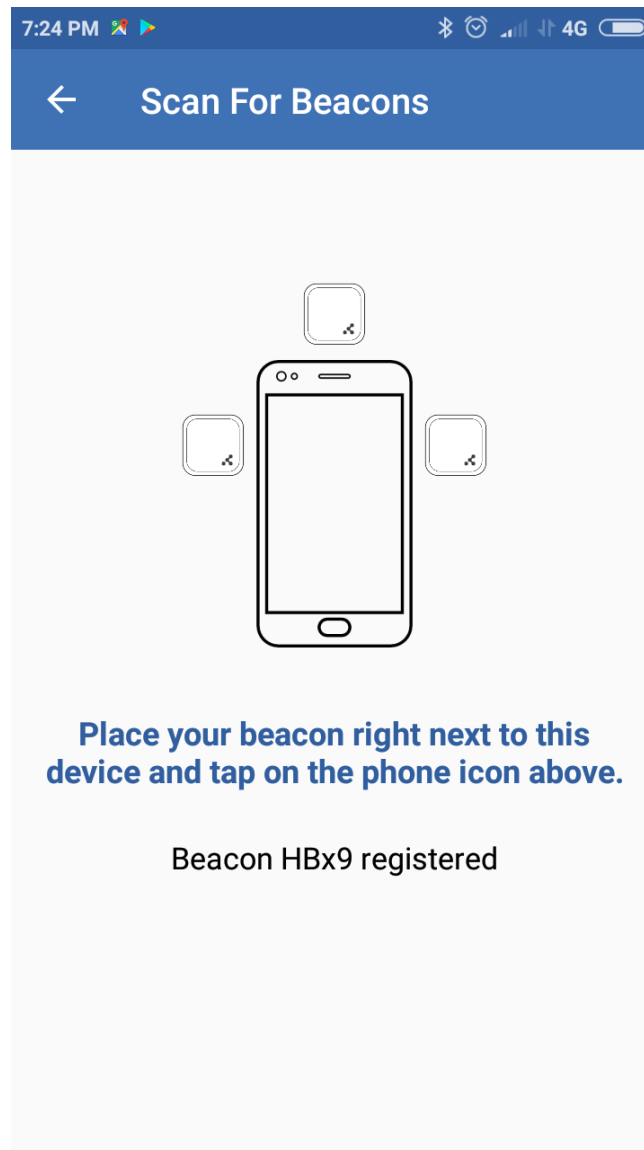
6.5 Διεπαφή Ανίχνευσης συσκευών Beacons



Εικόνα 37: Διεπαφή Ανίχνευσης συσκευών Beacons

Στην οθόνη ανίχνευσης για συσκευές beacons ο χρήστης θα πρέπει να τοποθετήσει τη συσκευή beacon που επιθυμεί σε απόσταση μικρότερη των 5 εκατοστών (<5cm) από τη συσκευή του κινητού του τηλεφώνου και να πατήσει το εικονίδιο στο κέντρο της οθόνης, ώστε να ξεκινήσει η λειτουργία ανίχνευσης.

Στο κάτω μέρος της οθόνης θα εμφανιστεί το μήνυμα "Scanning for Beacon Devices".

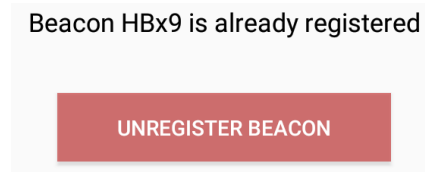


Εικόνα 38: Οθόνη Ανίχνευσης συσκευών Beacons

Εφόσον ο χρήστης έχει πατήσει την επιλογή ανίχνευσης και η διαδικασία εύρεσης κοντινών beacons έχει ξεκινήσει θα εμφανιστεί η επιγραφή “Beacon **HBx9** registered”. Αφού συμβεί αυτό η ανίχνευση θα σταματήσει.

Αν ο χρήστης προσπαθήσει να ανιχνεύσει την ίδια συσκευή θα εμφανιστεί το μήνυμα “Beacon HBx9 is already registered” και το κουμπί με την ετικέτα “Unregister beacon” θα εμφανιστεί στο κάτω μέρος της οθόνης. Η διαδικασία απαιτεί την ενεργοποίηση του λειτουργίας Bluetooth γι’ αυτό σε περίπτωση που δεν είναι ενεργοποιημένο ήδη θα εμφανιστεί στο κάτω μέρος της οθόνης το μήνυμα “ADL App θέλει να ενεργοποιήσει το Bluetooth”. Στην περίπτωση που ο χρήστης δεν ενεργοποιήσει το Bluetooth δε θα μπορεί να πραγματοποιηθεί η διαδικασία ανίχνευσης για BLE συσκευές.

HBx9: Αναγνωριστικό συσκευής Beacon

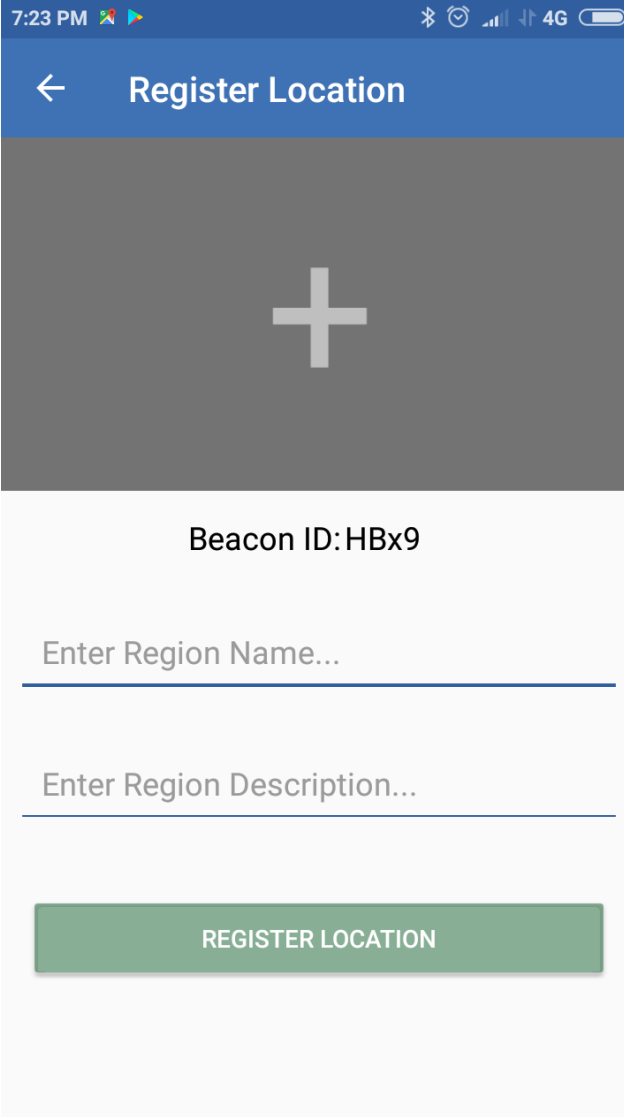


Εικόνα 39: Επιλογή Unregister Beacon

Αφού πατήσει την επιλογή “Unregister Beacon” θα εμφανιστεί το μήνυμα “IBeacon: HBx9 unregistered successfully”. Τότε η ανίχνευση θα αρχίσει ξανά και η συσκευή beacon θα μπορεί να δηλωθεί εκ νέου.

Σε οποιαδήποτε περίπτωση αφού μια συσκευή beacon ανιχνευθεί και δηλωθεί τότε ο χρήστης θα οδηγηθεί μετά από 3 δευτερόλεπτα στη διεπαφή καταχώρησης μιας νέας περιοχής.

6.6 Διεπαφή Καταχώρησης Περιοχής/Beacon Region - NewRegionActivity



7:23 PM

← Register Location

+

Beacon ID: HBx9

Enter Region Name...

Enter Region Description...

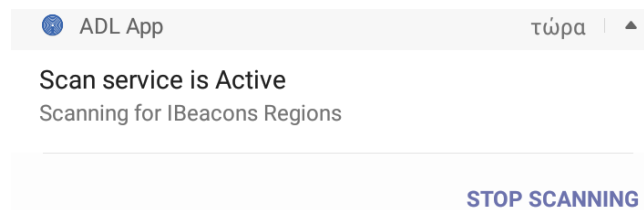
REGISTER LOCATION

Εικόνα 40: Διεπαφή Καταχώρησης Περιοχής/Beacon Region

Σε αυτήν τη διεπαφή οδηγείται ο χρήστης αφού ολοκληρωθεί η διαδικασία ανίχνευσης και δήλωσης της συσκευής beacon που τοποθέτησε δίπλα στη συσκευή του κινητού του τηλεφώνου. Ο χρήστης καλείται να συμπληρώσει τα 2 πεδία. Το πρώτο αφορά στο όνομα της περιοχής και το δεύτερο σε μια σύντομη περιγραφή αυτής. Ακόμα σημαντικό πεδίο αποτελεί και αυτό της φωτογραφίας του χώρου γι' αυτό είναι και αυτό το πεδίο υποχρεωτικό να συμπληρωθεί. Εφόσον ο χρήστης πληκτρολογήσει όνομα και περιγραφή και διαλέξει τη φωτογραφία που θέλει μπορεί να πατήσει το κουμπί με την επιγραφή "Register Location". Μόλις ολοκληρωθεί η διαδικασία καταχώρησης νέας περιοχής από το χρήστη θα εμφανιστεί στο κάτω μέρος της οθόνης του το μήνυμα: "Location is Registered". Ο χρήστης αυτό που μένει να κάνει είναι να τοποθετήσει τη συσκευή beacon σε κάποιο σημείο μέσα στο χώρο που δήλωσε. (π.χ κουζίνα, μπάνιο, κρεβατοκάμαρα

κ.λ.π). Για την υλοποίηση της εφαρμογής χρειάστηκαν τρεις(3) συσκευές beacons και γι' αυτό το λόγο ο χρήστης θα πρέπει να πραγματοποιήσει τη διαδικασία ανίχνευσης beacon και καταχώρησης νέας περιοχής τρεις φορές.

Αφού καταχωρήσει την τρίτη και τελευταία συσκευή beacon θα ξεκινήσει η λειτουργία(service) της εφαρμογής η οποία ανιχνεύει συνεχώς για σήμα των συσκευών αυτών κατά την είσοδο και έξοδο του χρήστη από κάποια περιοχή. Μια ειδοποίηση με την εξής παρακάτω μορφή θα εμφανιστεί στο πάνω μέρος της οθόνης της συσκευής του κινητού του τηλεφώνου.



Εικόνα 41: Παράθυρο ειδοποίησης για ανίχνευση beacons

Ο χρήστης μπορεί ανά πάσα στιγμή να σταματήσει αυτή τη λειτουργία πατώντας την επιλογή "Stop Scanning" στο δεξί κάτω μέρος.

Σε αυτό το σημείο στη βάση μας θα έχει δημιουργηθεί μέσα στο έγγραφο του συνδεδεμένου χρήστη(user ID) η υπό-συλλογή(sub-collection) με το όνομα "Regions".

```
Map<String, Object> regionsMap = new HashMap<>();
regionsMap.put("image_url", downloadUri);
regionsMap.put("thumb_image", downloadThumbUri);
regionsMap.put("beacon_id", beacon_id);
regionsMap.put("beacon_major", beacon_major);
regionsMap.put("beacon_address", beacon_mac);
regionsMap.put("status", "default");
regionsMap.put("region_name", region_name);
regionsMap.put("desc", region_description);
regionsMap.put("user_id", current_user_id);
regionsMap.put("timestamp", FieldValue.serverTimestamp());

firebaseFirestore.collection("Users/" + current_user_id + "/Regions").add(re-
gionsMap).addOnCompleteListener(task1 -> {

    if (task1.isSuccessful()) {

        Snackbar.make(v, "Location is Registered", Snackbar.LENGTH_SHORT)
            .setAction("Action", null).show();

        Intent mServiceIntent = new Intent(NewRegionActivity.this, Foreground-
ScanService.class);
        mServiceIntent.putExtra("region_name", region_name);
```

```

mServiceIntent.putExtra("beacon_major", beacon_major);
startService(mServiceIntent);

Timer timer = new Timer();
timer.schedule(new TimerTask() {
    @Override
    public void run() {

        Intent mainIntent = new Intent(NewRegionActivity.this, MainAc-
tivity.class);
        startActivity(mainIntent);
        finish();
    }
}, 2000);

} else {

}

regionProgress.setVisibility(View.INVISIBLE);
});

```

Στο παραπάνω κώδικα δημιουργείται η νέα περιοχή με τα στοιχεία που εισήγαγε ο χρήστης καθώς και στοιχεία που καταχωρούνται όπως:

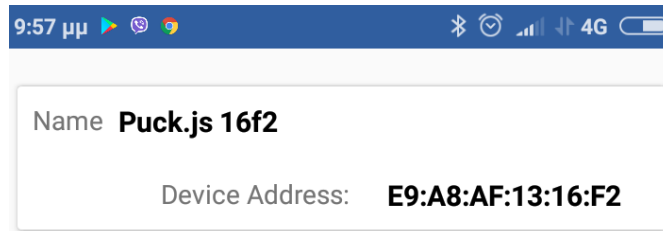
- ✓ Το `image_url` της φωτογραφίας που επέλεξε `image_url`
- ✓ Η κύρια τιμή της συσκευής Beacon `beacon_major`
- ✓ Το id της συσκευής Beacon `beacon_id`
- ✓ Το όνομα της περιοχής `region_name`
- ✓ Το ID του παρόντος συνδεδεμένου χρήστη `user_id`
- ✓ Περιγραφή της περιοχής `desc`
- ✓ Status περιοχής(δηλαδή αν βρίσκεται στην εκάστοτε περιοχή ο χρήστης) `status`
- ✓ Ημέρα και ώρα καταχώρησης της περιοχής `timestamp`

Σε περίπτωση που ο χρήστης βρίσκεται σε κάποια από τις περιοχές που έχει δηλώσει τότε δίπλα στο όνομα της περιοχής θα εμφανιστεί το εικονίδιο που φαίνεται στα αριστερά κάνοντας έτσι τη δραστηριότητα του χρήστη «ζωντανή».

 > Users > xoHXsNhiXGdE... > Regions > 5XbXJ0cCOawE...

Στο παραπάνω μονοπάτι εμφανίζεται η συλλογή Users όπου μέσα σε αυτήν υπάρχει το έγγραφο με το ID του συνδεδεμένου με την εφαρμογή χρήστη και μέσα σε αυτό έχει δημιουργηθεί η υπο-συλλογή Regions με τα έγγραφα των καταχωρημένων περιοχών.

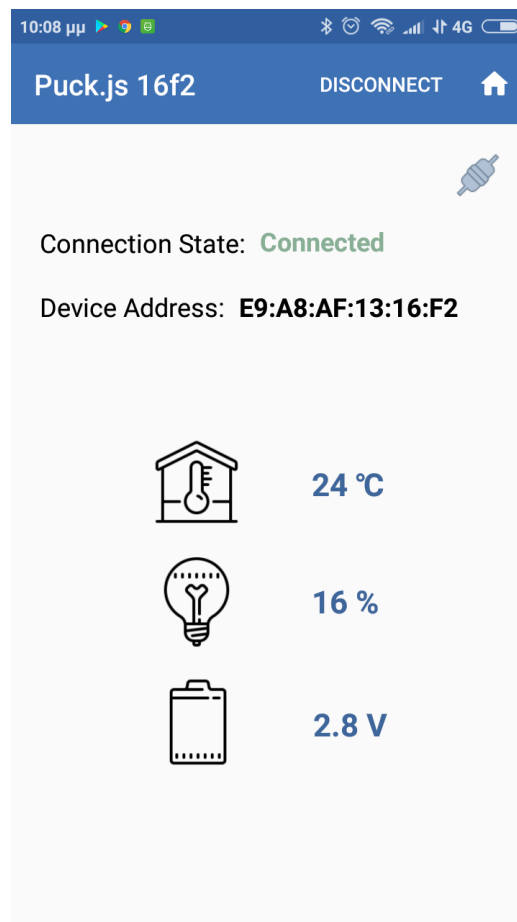
6.7 Διεπαφή Ανίχνευσης συσκευής Puck



Εικόνα 42: Διεπαφή Ανίχνευση συσκευής Puck

Σε αυτή τη διεπαφή θα οδηγηθεί ο χρήστης εφόσον επιθυμεί να συνδεθεί με τη συσκευή Puck και πατήσει την επιλογή “BLE” στο αρχικό μενού, έτσι ώστε αυτή να μπορεί να επικοινωνεί με τις συσκευές Beacons καθώς και να μπορεί ο χρήστης να τη χρησιμοποιεί πατώντας το κουμπί της για να δηλώσει την κατάσταση στην οποία βρίσκεται. Για το σκοπό της διπλωματικής εργασίας η κινητή συσκευή του χρήστη μπορεί να ανιχνεύσει μόνο τη συσκευή Puck με το συγκεκριμένο όνομα και διεύθυνση (mac address).

6.8 Διεπαφή Σύνδεσης με τη συσκευή Puck



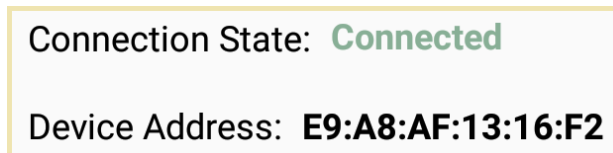
Εικόνα 43: Διεπαφή σύνδεσης με τη συσκευή Puck



Εικόνα 44: Γραμμή Εργαλείων - Μενού Διεπαφής Σύνδεσης με Puck

Παραπάνω εμφανίζεται το μενού στη γραμμή εργαλείων της διεπαφής χρήστη σύνδεσης με τη συσκευή Puck. Στο αριστερό μέρος εμφανίζεται το όνομα της συσκευής **Puck.js 16f2** και στα δεξιά εμφανίζονται οι επιλογές Connect/Disconnect στις οποίες μπορεί να πατήσει ο χρήστης και να συνδεθεί ή να αποσυνδεθεί αντίστοιχα με τη συσκευή Puck. Επίσης εμφανίζεται και το εικονίδιο “Home” το οποίο μπορεί να πατήσει ο χρήστης και να οδηγηθεί πίσω στην κύρια διεπαφή χρήστη.

Η εφαρμογή θα ξεκινήσει αυτόματα τη σύνδεση με τη συσκευή Puck και εφόσον συνδεθεί με εκείνη το πεδίο “Connection State:” θα αλλάξει σε **Connected** που σημαίνει ότι η σύνδεση ήταν επιτυχημένη και πως πλέον η συσκευή μπορεί να επικοινωνεί με τα beacons που υπάρχουν στον ευρύτερο χώρο κατοικίας του χρήστη. Επιπλέον στο πεδίο “Device Address” εμφανίζεται η διεύση MAC της συσκευής Puck.



Εικόνα 45: Connection State & Device Address

Τη στιγμή που θα πραγματοποιηθεί η σύνδεση με τη συσκευή Puck στέλνουμε την εντολή(doScan) που αναφέραμε παραπάνω για πραγματοποίηση ανίχνευσης και διαφήμισης των δεδομένων του Puck.



Θερμοκρασία στην περιοχή Region που βρίσκεται το Puck



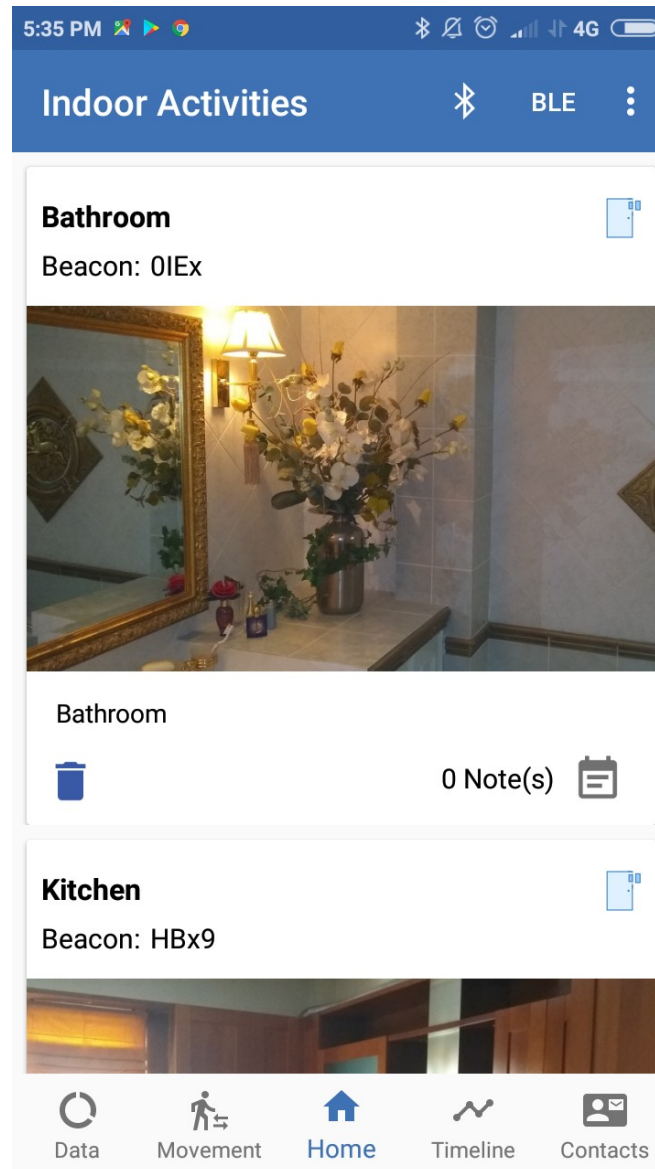
Φωτεινότητα στην περιοχή-Region που βρίσκεται το Puck



Ποσοστό μπαταρίας Puck σε Volt(V)

Ο χρήστης πλέον μπορεί να συνεχίσει να χρησιμοποιεί την εφαρμογή και να χρησιμοποιήσει το Puck.

6.9 Καρτέλα Λίστας Περιοχών Χρήστη - HomeFragment

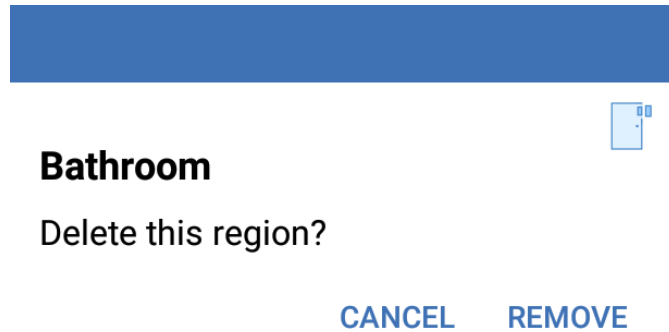


Εικόνα 46: Καρτέλα Λίστας Περιοχών Χρήστη - HomeFragment

Στην παραπάνω Εικόνα εμφανίζονται σε μορφή καρτελών τα έγγραφα που ανήκουν στη συλλογή "Regions" με κάποια από τα πεδία που αποθηκεύτηκαν κατά τη καταχώρηση μιας περιοχής από το χρήστη, όπως η φωτογραφία, το όνομα της περιοχής και η περιγραφή της περιοχής.

Ο χρήστης μπορεί ανά πάσα στιγμή να διαγράψει κάποια από τις δηλωμένες περιοχές-regions που επιθυμεί. Για να το πετύχει αυτό θα πρέπει να πατήσει το κουμπί στο κάτω αριστερό μέρος


της καρτέλας. 




Εικόνα 47: Παράθυρο Διαγραφής Περιοχής Region

Εφόσον πατήσει το κουμπί διαγραφής μιας περιοχής θα εμφανιστεί το παραπάνω παράθυρο διαγραφής περιοχής με τις επιλογές “Cancel” και “Remove”.

Αν πατήσει την επιλογή Cancel η περιοχή που έχει επιλέξει δε θα διαγραφεί ενώ αν πατήσει την επιλογή Remove η περιοχή θα διαγραφεί και δεν θα υπάρχει πλέον στη λίστα με όλες τις περιοχές. Αφού πραγματοποιηθεί η διαγραφή θα εμφανιστεί στο κάτω μέρος της οθόνης το μήνυμα «Region Deleted»

0 Note(s) 

Μια ακόμη δυνατότητα που παρέχει η εφαρμογή στο χρήστη είναι αυτή της καταχώρησης σημαντικών σημειώσεων που αφορούν στη εκάστοτε επιθυμητή περιοχή/region. Πατώντας το παραπάνω εικονίδιο θα οδηγηθεί στη διεπαφή χρήστη καταχώρησης σημείωσης.

2 Note(s) 

Παραπάνω φαίνεται ο αριθμός των σημειώσεων που έχει καταχωρήσει ο χρήστης στην εκάστοτε επιθυμητή περιοχή.

```
Map<String, Object> notesMap = new HashMap<>();
notesMap.put("title", note_title);
notesMap.put("text", note_text);
notesMap.put("region_name", home_region_name);
notesMap.put("region_id", home_region_id);

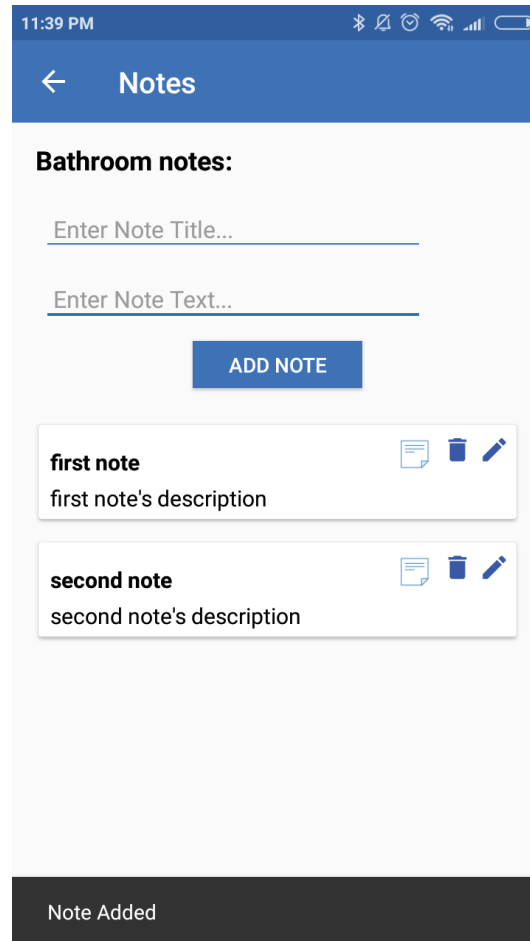
firebaseFirestore.collection("Users").document(currentUserId).collection("Regions").document(home_region_id)
    .collection("Notes")
    .add(notesMap).addOnCompleteListener(task -> {
});
```

Η υπό-συλλογή “Notes” θα δημιουργηθεί μέσα στη συλλογή “Regions” η οποία βρίσκεται μέσα στο έγγραφο με ID αυτό του συνδεδεμένου χρήστη.

Για να λάβουμε το ID του συνδεδεμένου χρήστη μπορούμε να χρησιμοποιήσουμε τη μέθοδο **firebaseAuth.getCurrentUser().getUid()**; αφού πρώτα αρχικοποιήσουμε την αυθεντικοποίηση της Cloud Firestore **firebaseAuth = FirebaseAuth.getInstance()**;

```
currentUserId = firebaseAuth.getCurrentUser().getUid();
```

6.10 Διεπαφή Χρήστη Καταχώρησης Σημειώσεων Περιοχής Region - NotesActivity



Εικόνα 48: Διεπαφή χρήστη καταχώρησης σημειώσεων περιοχής/region

Ο χρήστης θα πρέπει να συμπληρώσει τα δύο πεδία που εμφανίζονται στην παραπάνω Εικόνα, δηλαδή αυτό του τίτλου και του κειμένου της σημείωσης. Αν κάποιο απ' αυτά δεν είναι συμπληρωμένο δεν θα μπορεί να καταχωρηθεί η σημείωση.

Εφόσον λοιπόν έχει συμπληρώσει και τα 2 πεδία τότε μπορεί να πατήσει το κουμπί με την επιγραφή "Add note" και μόλις καταχωρηθεί η σημείωση επιτυχώς ο χρήστης θα δει στο κάτω μέρος της οθόνης του το μήνυμα → "Note Added"



Αν ο χρήστης πατήσει το εικονίδιο επεξεργασίας θα εμφανιστεί στην οθόνη του το εξής παρακάτω παράθυρο. Μέσα στο παράθυρο θα υπάρχουν 2 πεδία στα οποία θα αναγράφονται ο τίτλος και το κείμενο της σημείωσης που θέλει να επεξεργαστεί και στο οποίο θα πρέπει να εισάγει τα νέα στοιχεία. Εφόσον πατήσει το κουμπί με την επιγραφή “Update” η αλλαγή θα αποθηκευτεί στη βάση δεδομένων και συγκεκριμένα στο έγγραφο με το ID της συγκεκριμένης σημείωσης. Αν πατήσει την επιλογή “Cancel” θα επιστρέψει πίσω στην καρτέλα με τη λίστα των αποθηκευμένων σημειώσεων.

Edit Note:

first note

first note's description

CANCEL UPDATE

Εικόνα 49: Παράθυρο Επεξεργασίας Σημείωσης



Αν ο χρήστης πατήσει το εικονίδιο διαγραφής θα εμφανιστεί στην οθόνη του το εξής παρακάτω παράθυρο. Πατώντας την επιλογή “Remove” η σημείωση θα διαγραφεί, ενώ με την επιλογή “Cancel” θα επιστρέψει πίσω στην καρτέλα με τη λίστα των αποθηκευμένων σημειώσεων της περιοχής.

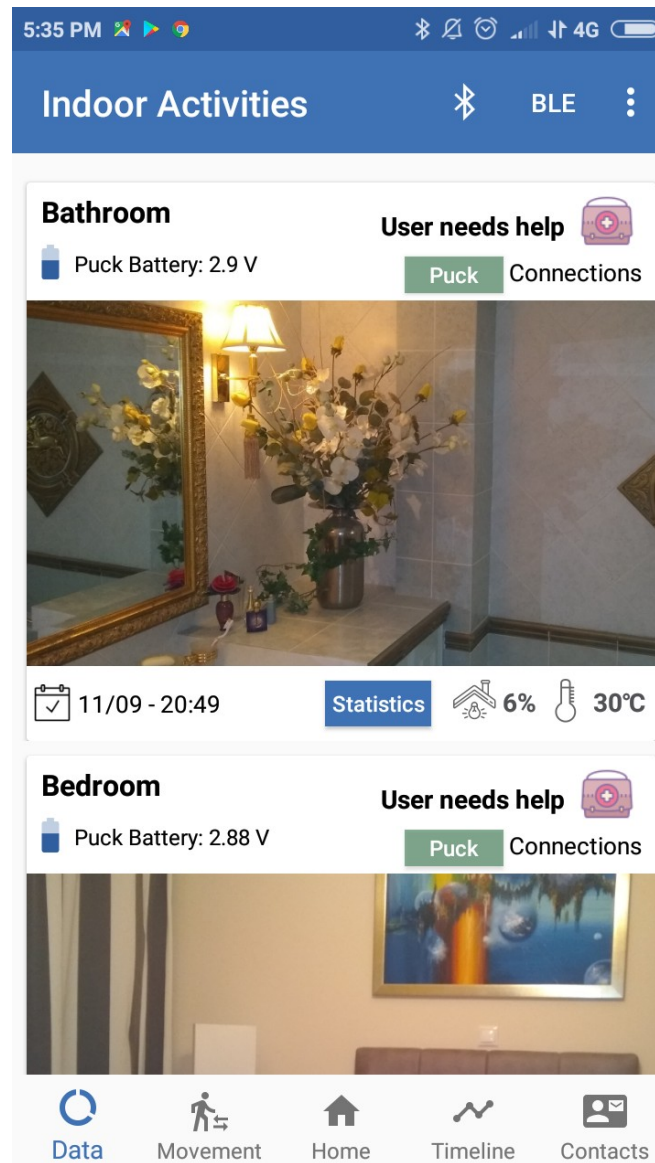
first note

Delete this note?

CANCEL REMOVE

Εικόνα 50: Παράθυρο Διαγραφής Σημείωσης

6.11 Καρτέλα Λίστας Εγγράφων Συλλογής PuckData



Εικόνα 51: Καρτέλα Λίστας Εγγράφων Συλλογής PuckData

Στην καρτέλα με το όνομα "Data" που απεικονίζεται στην παραπάνω Εικόνα εμφανίζονται όλα τα έγγραφα της συλλογής "PuckData" τα οποία προκύπτουν από τη συσκευή Puck και λαμβάνονται όπως έχει αναφερθεί στο κεφάλαιο 3 κάθε 5 λεπτά ή κάθε φορά που ο χρήστης θα πατήσει το κουμπί της συσκευής. Η συσκευή Puck ανιχνεύει κοντινά Beacons και διαφημίζει τα δεδομένα της για την περιοχή στην οποία βρίσκεται η κοντινότερη συσκευή Beacon.

Οι καρτέλες εμφανίζονται σε φθίνουσα ταξινόμηση με την πιο πρόσφατη να εμφανίζεται στο πάνω μέρος και τις παλαιότερες να ακολουθούν καθώς κάνουμε scroll down.

```

(myState == 1 && home_location != null) {
    puckDataMap.put("temperature", temperature);
    puckDataMap.put("light", light);
    puckDataMap.put("battery", battery);
    puckDataMap.put("region", home_location);
    puckDataMap.put("region_image", regionImage);
    puckDataMap.put("user_status", "User feels good");
    puckDataMap.put("current_date", currentDate);
    puckDataMap.put("current_week", w1 + " - " + w2);
    puckDataMap.put("current_month", current_month);
    puckDataMap.put("timestamp", FieldValue.serverTimestamp());

    firebaseFirestore.collection("PuckData").add(puckDataMap).addOnCom-
pleteListener(task -> {

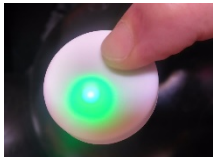
        if (!task.isSuccessful()) {

            String error_message = task.getException().getMessage();
            Log.e(TAG, "Error: " + error_message);
        } else {

            Log.i(TAG, "Data Stored");
        }
    });
}

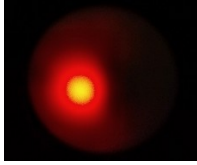
```

Ο χρήστης εφόσον έχει συνδεθεί με τη συσκευή Puck μπορεί όποτε θελήσει να πατήσει το κουμπί απλά ή παρατεταμένα. Με το απλό πάτημα του κουμπιού το πράσινο LED θα αναβοσβήσει. Έτσι, δηλώνει πως βρίσκεται σε καλή κατάσταση.



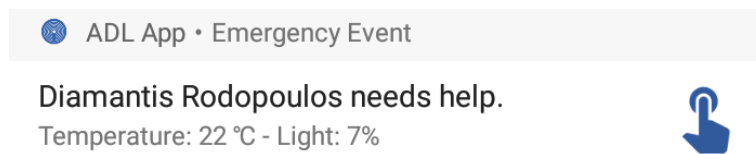
Στη συλλογή PuckData θα υπάρχουν τα εξής πεδία:

- ✓ Η τιμή της θερμοκρασίας `temperature`
- ✓ Η τιμή της φωτεινότητας `light`
- ✓ Η τιμή της μπαταρίας του Puck `battery`
- ✓ Η τιμή της κατάστασης του χρήστη ("User Feels Good") `user_status`
- ✓ Η τρέχουσα εβδομάδα `current_week`
- ✓ Ο τρέχων μήνας `current_month`
- ✓ Η τρέχουσα ώρα και ημέρα `timestamp`



Με το παρατεταμένο πάτημα του κουμπιού το κόκκινο LED θα αναβοσβήσει. Έτσι, δηλώνει πως βρίσκεται σε κατάσταση έκτακτης ανάγκης.

Η παρακάτω ειδοποίηση θα εμφανιστεί στην κινητή συσκευή καθώς και θα συνοδευτεί από ήχο και δόνηση της συσκευής, η οποία δηλώνει πως ο χρήστης έχει πατήσει το κουμπί έκτακτης ανάγκης και στην οποία αναγράφονται: το ονοματεπώνυμο και οι τιμές θερμοκρασίας και φωτεινότητας εκείνη την ώρα.



Εικόνα 52: Ειδοποίηση Έκτακτης Ανάγκης

```
(myState == 2 && home_location != null) {
    puckDataMap.put("temperature", temperature);
    puckDataMap.put("light", light);
    puckDataMap.put("battery", battery);
    puckDataMap.put("region", home_location);
    puckDataMap.put("region_image", regionImage);
    puckDataMap.put("user_status", "User feels good");
    puckDataMap.put("current_date", currentDate);
    puckDataMap.put("current_week", w1 + " - " + w2);
    puckDataMap.put("current_month", current_month);
    puckDataMap.put("timestamp", FieldValue.serverTimestamp());

    firebaseFirestore.collection("PuckData").add(puckDataMap).addOnCom-
pleteListener(task -> {

        if (!task.isSuccessful()) {

            String error_message = task.getException().getMessage();
            Log.e(TAG, "Error: " + error_message);
        } else {

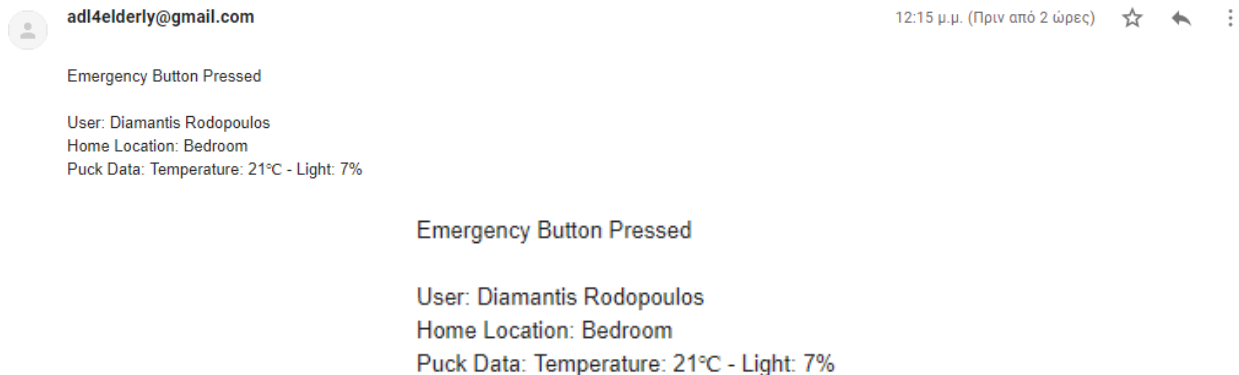
            Log.i(TAG, "Data Stored");
        }
    });
};
```

Τότε θα δημιουργηθεί ένα ακόμα έγγραφο με τη μόνη διαφορά ότι στο πεδίο user status θα υπάρχει ως τιμή το κείμενο «User needs help»

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο πατώντας το κουμπί έκτακτης ανάγκης θα αποσταλεί ηλεκτρονικό μήνυμα σε όλες τις ηλεκτρονικές διευθύνσεις που έχει καταχωρήσει ο χρήστης στην καρτέλα e-mail.

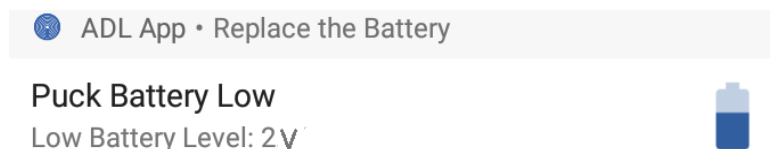
Για την επίτευξη αυτής της λειτουργίας χρησιμοποιήθηκε το [JavaMail API](#).

Στο e-mail που θα λάβουν τα πρόσωπα των οποίων οι ηλεκτρονικές διευθύνσεις έχουν δηλωθεί από το χρήστη θα αναγράφονται τα εξής: Η επιγραφή «Emergency Button Pressed», το ονοματεπώνυμο του χρήστη, το όνομα της περιοχής στην οποία ο χρήστης πάτησε το κουμπί έκτακτης ανάγκης, καθώς και τα δεδομένα (θερμοκρασία και φωτεινότητα) τη στιγμή εκείνη.

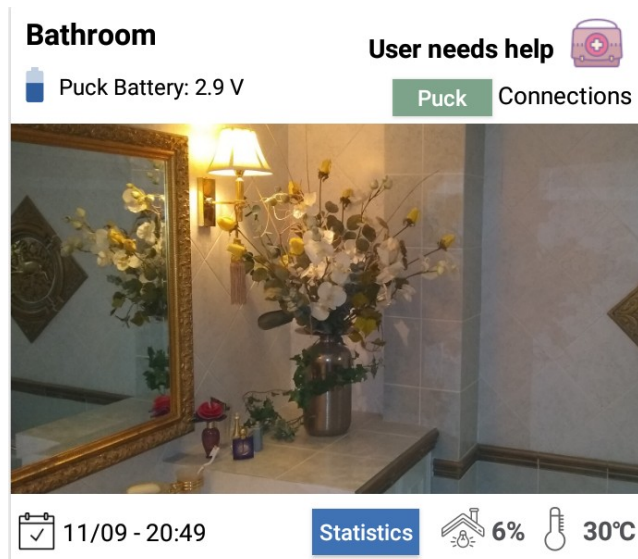


Εικόνα 53: e-mail έκτακτης ανάγκης

Επιπλέον αν σε περίπτωση η μπαταρία της συσκευής Puck βρίσκεται σε ποσοστό κάτω των 2.4 Volts θα εμφανιστεί στην οθόνη του χρήστη η ειδοποίηση πως το ποσοστό της μπαταρίας Puck είναι χαμηλό και πως θα πρέπει να αλλάξει μπαταρία, διότι δε θα μπορεί να συνδεθεί σε αυτήν και να λάβει τα δεδομένα ή να ειδοποιήσει για την κατάστασή του.

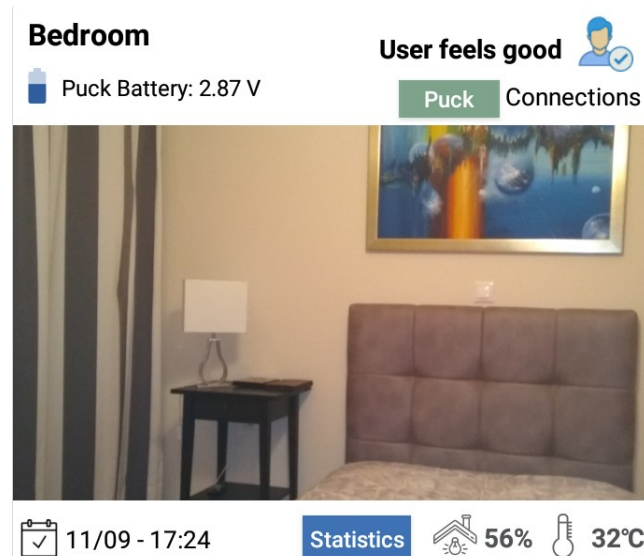


Εικόνα 54: Ειδοποίηση Χαμηλής Μπαταρίας



Εικόνα 55: Καρτέλα Δεδομένων Κατάστασης Έκτακτης ανάγκης

Όπως φαίνεται και στην παραπάνω Εικόνα ο χρήστης έχει πατήσει το κουμπί Puck για να δηλώσει έκτακτη ανάγκη. Επάνω δεξιά εμφανίζεται η τιμή του πεδίου “status” του εκάστοτε εγγράφου της συλλογής “PuckData” με την επιγραφή «User needs help»



Εικόνα 56: Καρτέλα Δεδομένων Καλής κατάστασης χρήστη

Όπως φαίνεται και στην παραπάνω Εικόνα ο χρήστης έχει πατήσει το κουμπί Puck για να δηλώσει πως βρίσκεται σε καλή κατάσταση. Επάνω δεξιά εμφανίζεται η τιμή του πεδίου “status” του εκάστοτε εγγράφου της συλλογής “PuckData” με την επιγραφή «User feels good».

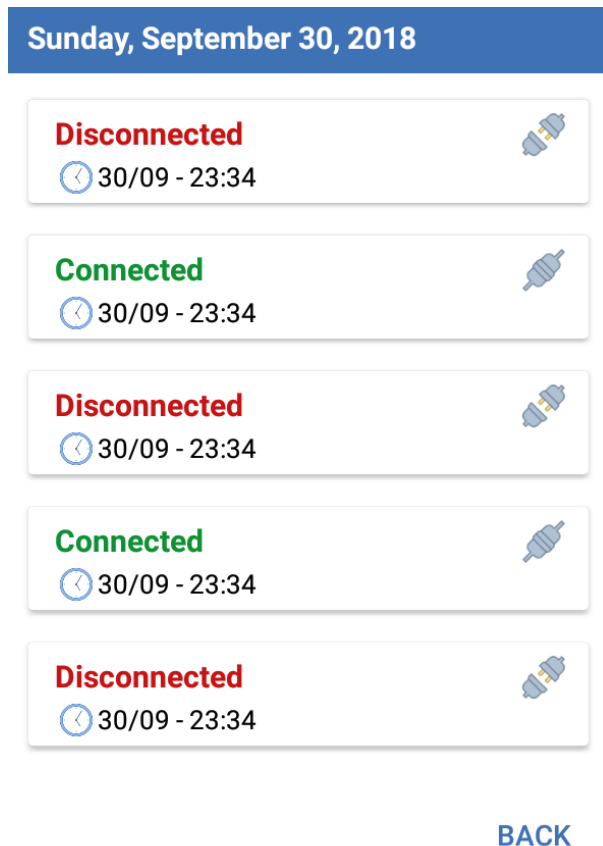
Στην περίπτωση που τα δεδομένα προέλθουν κατά την ανίχνευση και διαφήμιση των δεδομένων της συσκευής Puck που πραγματοποιείται ανα 5 λεπτά όπως έχει προαναφερθεί τότε δεν θα εμφανίζεται κάτι και η τιμή του πεδίου “user_status” θα είναι άγνωστη. (Unknown).

Στην καρτέλα του μενού “Data” όπως αναφέραμε πιο πάνω εμφανίζονται όλα τα έγγραφα(documents) της συλλογής PuckData.

Puck Connections

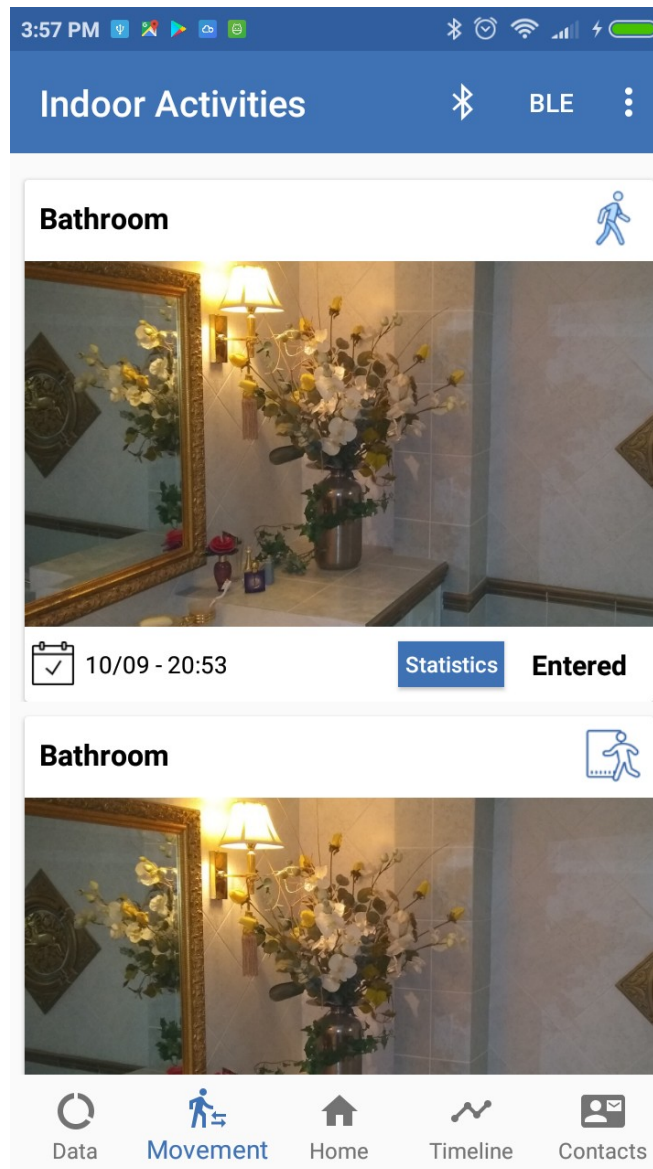
Ο χρήστης μπορεί να πατήσει το κουμπί με την επιγραφή Puck Connections και να δει όλες τις συνδέσεις της συσκευής Puck με την κινητή του συσκευή την εκάστοτε ημέρα.

Κατά τη διάρκεια σύνδεσης/αποσύνδεσης δημιουργείται έγγραφο στη συλλογή με το όνομα “Connections”. Εμφανίζονται σε σειρά φθίνουσας ταξινόμησης από το πιο πρόσφατο στο παλαιότερο έγγραφο. Πατώντας το κουμπί με την επιγραφή “Back” θα επιστρέψει στην καρτέλα όλων των εγγράφων της συλλογής “PuckData”



Εικόνα 57: Παράθυρο Συνδέσεων Puck-Κινητής Συσκευής Χρήστη

6.12 Καρτέλα Λίστας Εγγράφων Συλλογής RegionEvents



Εικόνα 58: Καρτέλα Λίστας Εγγράφων Συλλογής RegionEvents

Στην καρτέλα “Movement” εμφανίζονται σε λίστα σαν καρτέλες όλες οι δραστηριότητες εισόδου/εξόδου από μια περιοχή/region του χρήστη οι οποίες προκύπτουν από την επικοινωνία της κινητής του συσκευής με τις συσκευές beacons που υπάρχουν στους

διάφορους χώρους της κατοικίας του. Οι καρτέλες εμφανίζονται σε φθίνουσα ταξινόμηση με την πιο πρόσφατη να εμφανίζεται στο πάνω μέρος και τις παλαιότερες να ακολουθούν καθώς κάνουμε scroll down.

```
Map<String, Object> enterMap = new HashMap<>();

enterMap.put("region_name", region_name);
enterMap.put("region_image", regionImage);
enterMap.put("action_type", "Entered");
enterMap.put("current_date", currentDate);
enterMap.put("timestamp", FieldValue.serverTimestamp());

firebaseFirestore.collection("RegionEvents").add(enterMap).addOnSuccessListener(documentReference -> {

}).addOnFailureListener(e1 -> {
    String error = e1.getMessage();
    Log.e(TAG, "Error: " + error);
});
Log.i(TAG, "New Region entered: " + region.getIdentifier());

Map<String, Object> data = new HashMap<>();
data.put("status", "access");

firebaseFirestore.collection("Users").document(userId).collection("Regions").document(region_id)
    .set(data, SetOptions.merge());
```

Στον παραπάνω κώδικα ανάλογα με τη δραστηριότητά του εισοδο/έξοδο προστίθεται κάθε φορά ένα έγγραφο στην αρχική συλλογή με το όνομα "RegionEvents". Τα πεδία που υπάρχουν μέσα σε κάθε έγγραφο είναι τα εξής:

- ✓ Το όνομα της περιοχής/region `region_name`
- ✓ Η Εικόνα της περιοχής που έχει δηλώσει ο χρήστης `region_image`
- ✓ Το είδος της δραστηριότητας(είσοδος-έξοδος) `action_type`
- ✓ Η τρέχουσα ημερομηνία `current_date`
- ✓ Η ώρα και ημέρα της δραστηριότητας `timestamp`

Και στις δύο περιπτώσεις εισόδου/εξόδου η τιμή του πεδίου status του εγγράφου της περιοχής της συλλογής "Regions" που αναφέρθηκε παραπάνω, όπου συμβαίνει η δραστηριότητα του χρήστη αλλάζει ζωντανά.

Όταν θα εισέλθει στην περιοχή το πεδίο status θα πάρει την τιμή “access” ενώ όταν θα εξέλθει από την περιοχή θα πάρει την τιμή “default”.

Με αυτόν τον τρόπο επιτυγχάνεται η ζωντανή παρακολούθηση της δραστηριότητας του χρήστη.

Στην αντίθετη περίπτωση που ο χρήστης εξέρχεται από μια περιοχή η τιμή του πεδίου action_type στο έγγραφο της εκάστοτε δραστηριότητας της συλλογής “RegionEvents” θα είναι “Exited”.

Bathroom



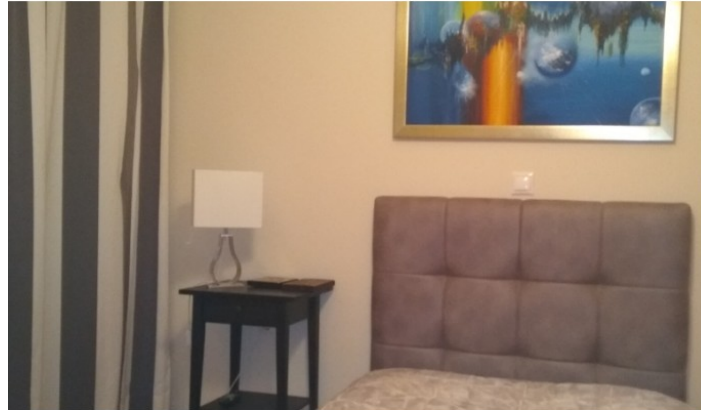
10/09 - 20:53

Statistics

Entered

Εικόνα 59: Καρτέλα δεδομένων εισόδου στην περιοχή/region

Bedroom



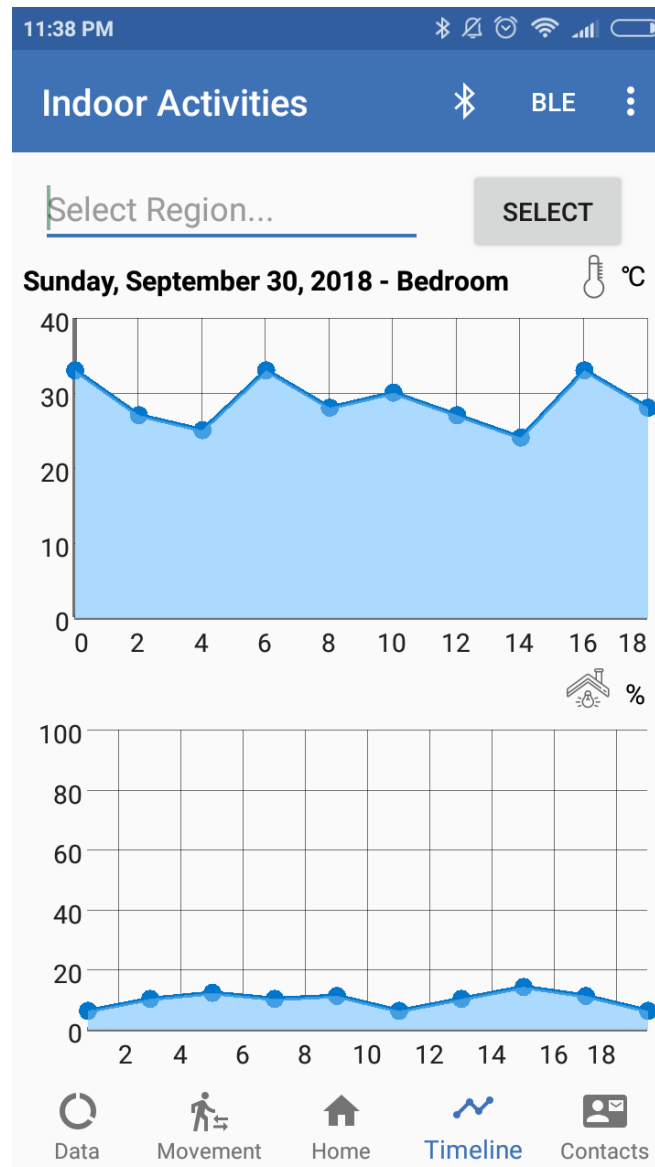
10/09 - 19:21

Statistics

Exited

Εικόνα 60: Καρτέλα δεδομένων εξόδου από την περιοχή/region

6.13 Διεπαφή Χρήστη Απεικόνισης Δεδομένων σε Γραφήματα-TimelineFragment



Εικόνα 61: Διεπαφή Χρήστη - Γραφήματα Θερμοκρασίας και Φωτεινότητας Περιοχής

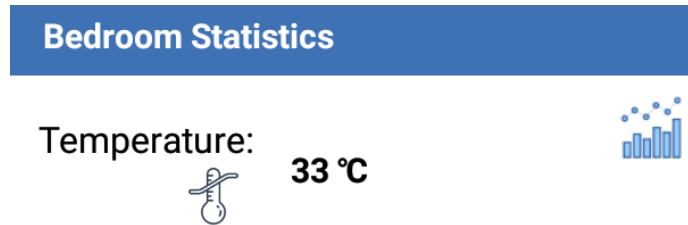
Ο χρήστης μπορεί να εισάγει στο πεδίο "Select Region" το όνομα της περιοχής/χώρου που επιθυμεί έτσι ώστε να λάβει τα δεδομένα της περιοχής αυτής την εκάστοτε ημέρα και τα οποία θα εμφανιστούν ως σημεία (data points) πάνω στα γραφήματα. Αφού πατήσει την επιλογή "SELECT" θα σχηματιστούν τα δύο γραφήματα. Επάνω από το γράφημα της θερμοκρασίας εμφανίζεται η εκάστοτε ημερομηνία και δίπλα σε αυτήν το όνομα της περιοχής που διάλεξε ο χρήστης.

Για την απεικόνιση των δεδομένων σε γραφήματα χρησιμοποιήθηκε η δημοφιλής βιβλιοθήκη η οποία βρίσκεται στον εξής σύνδεσμο: <https://github.com/jjoe64/GraphView>

Μια από τις πολλές δυνατότητες της βιβλιοθήκης γραφημάτων είναι το πάτημα ενός σημείου (data point) και η εμφάνιση της τιμής αυτού.

Αν πατήσουμε πάνω σε ένα data point του γραφήματος θερμοκρασίας θα εμφανιστεί το παρακάτω παράθυρο στο οποίο αναγράφονται:

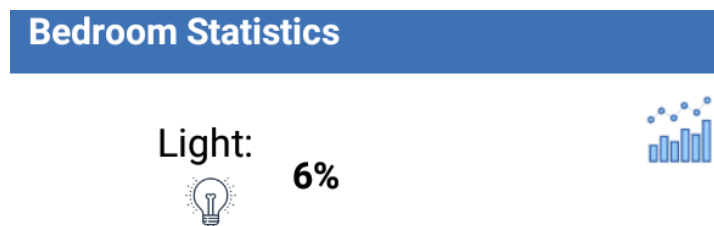
- Η τιμή της θερμοκρασίας
- Το όνομα της περιοχής για την οποία ζήτησε ο χρήστης τη γραφική απεικόνιση



Εικόνα 62: Παράθυρο τιμής θερμοκρασίας data point

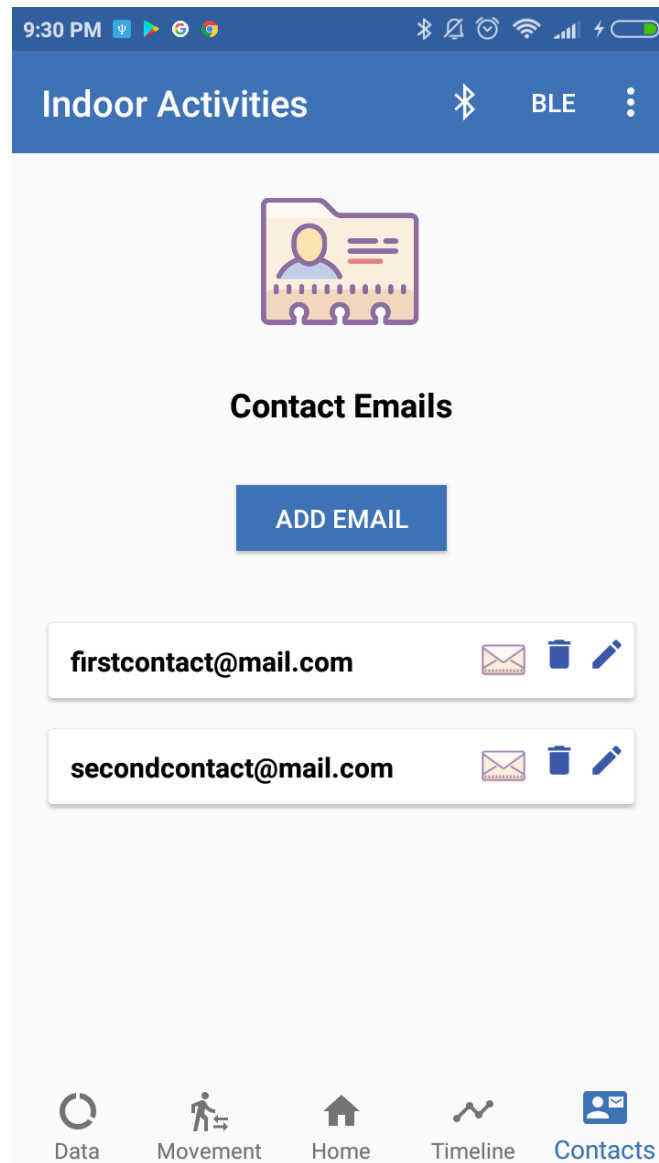
Αν πατήσουμε πάνω σε ένα data point του γραφήματος φωτεινότητας θα εμφανιστεί το παρακάτω παράθυρο στο οποίο αναγράφονται:

- Η τιμή της φωτεινότητας
- Το όνομα της περιοχής για την οποία ζήτησε ο χρήστης τη γραφική απεικόνιση



Εικόνα 63: Παράθυρο τιμής φωτεινότητας data point

6.14 Καρτέλα Χρήστη Λίστας Ηλεκτρονικών Διευθύνσεων



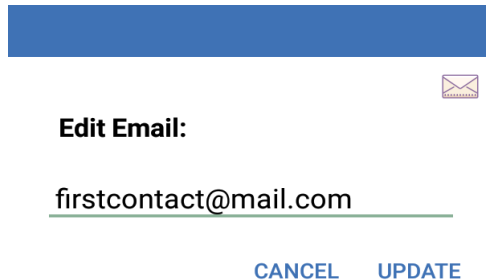
Εικόνα 64: Καρτέλα Χρήστη Λίστας Ηλεκτρονικών Διευθύνσεων

Όπως αναφέρθηκε και πρωτίστως η εφαρμογή παρέχει τη δυνατότητα στο χρήστη μέσω της συσκευής Puck να πατήσει το κουμπί της παρατεταμένα και να δηλώσει με αυτόν τον τρόπο πως βρίσκεται σε κατάσταση έκτακτης ανάγκης. Το κόκκινο κουμπί της συσκευής θα αναβοσβήσει έτσι ώστε βλέποντας το ο χρήστης να καταλάβει πως το σήμα εξέπεμψε κανονικά.

Στην καρτέλα που εμφανίζεται στην φωτογραφία αναγράφονται όλες οι ηλεκτρονικές διευθύνσεις που έχει καταχωρήσει ο χρήστης. Δίνεται η δυνατότητα σε αυτόν να αλλάξει κάποια από αυτές ή να την/τις διαγράψει.



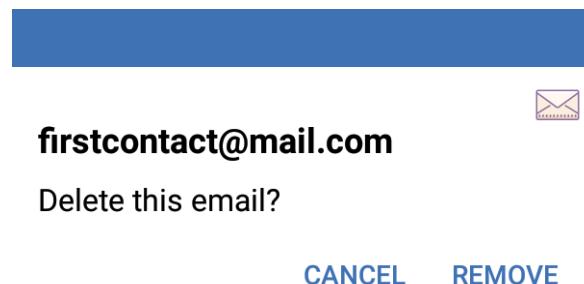
Αν ο χρήστης πατήσει το εικονίδιο επεξεργασίας θα εμφανιστεί στην οθόνη του το εξής παρακάτω παράθυρο. Μέσα στο παράθυρο θα υπάρχει ένα πεδίο στο οποίο θα αναγράφεται το e-mail το οποίο θέλει να επεξεργαστεί και στο οποίο θα πρέπει να εισάγει την νέα ηλεκτρονική διεύθυνση. Εφόσον πατήσει το κουμπί με την επιγραφή “Update” η αλλαγή θα αποθηκευτεί στη βάση δεδομένων και συγκεκριμένα στο έγγραφο με το ID του συγκεκριμένου e-mail. Αν πατήσει την επιλογή “Cancel” θα επιστρέψει πίσω στην καρτέλα με τη λίστα των αποθηκευμένων ηλεκτρονικών διευθύνσεων.



Εικόνα 65: Παράθυρο επεξεργασίας email

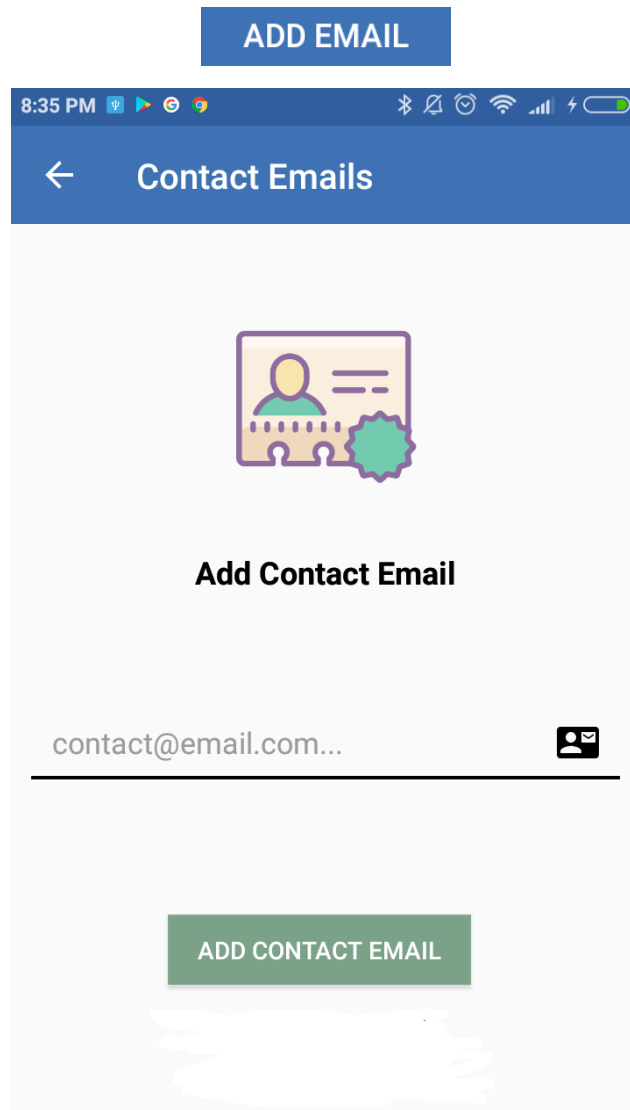


Αν ο χρήστης πατήσει το εικονίδιο διαγραφής θα εμφανιστεί στην οθόνη του το εξής παρακάτω παράθυρο. Πατώντας την επιλογή “Remove” το e-mail θα διαγραφεί, ενώ με την επιλογή “Cancel” θα επιστρέψει πίσω στην καρτέλα με τη λίστα των αποθηκευμένων ηλεκτρονικών διευθύνσεων.



Εικόνα 66: Παράθυρο διαγραφής email

Αν επιθυμεί να καταχωρήσει ένα νέο e-mail μπορεί να πατήσει το κουμπί με την επιγραφή “Add Email” και τότε θα οδηγηθεί στην διεπαφή καταχώρησης νέας ηλεκτρονικής διεύθυνσης όπως φαίνεται στην παρακάτω Εικόνα.



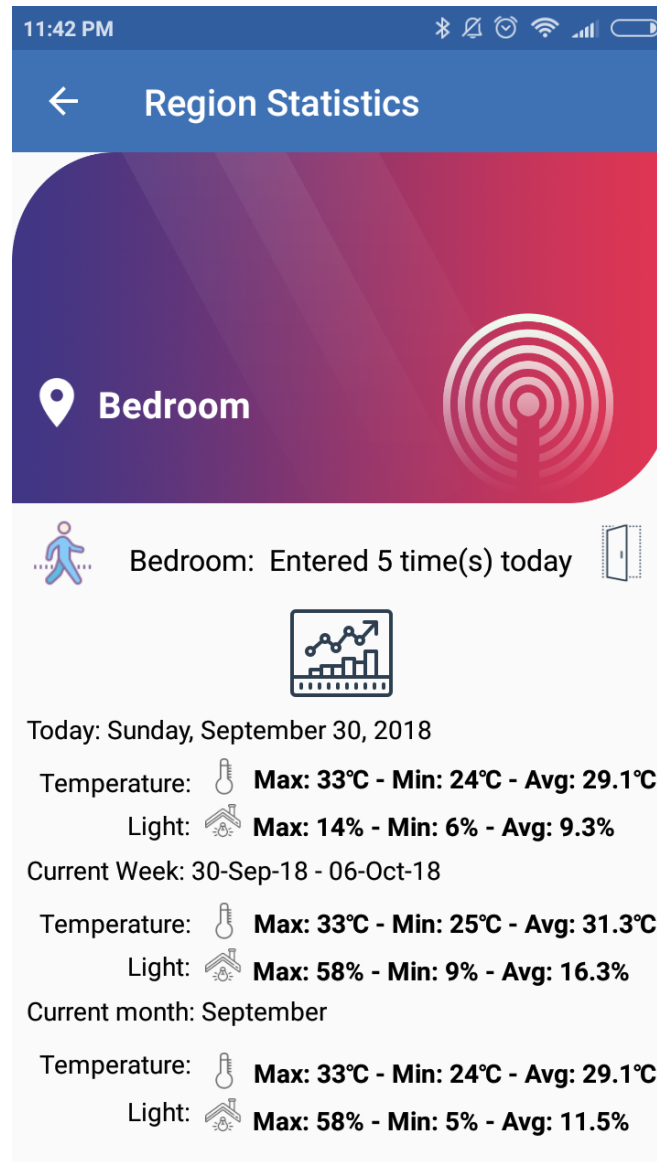
Εικόνα 67: Διεπαφή Χρήστη Καταχώρησης E-mail

Σε αυτό το σημείο στο πεδίο με την επιγραφή «contact@email.com...» θα πρέπει να πληκτρολογήσει την ηλεκτρονική διεύθυνση που θέλει.

Αφού συμπληρώσει το πεδίο θα πρέπει να πατήσει το κουμπί στο κάτω μέρος της οθόνης με την επιγραφή "Add Contact Email". Εφόσον η ηλεκτρονική διεύθυνση πληκτρολογηθεί σε έγκυρη μορφή τότε θα δει στο κάτω μέρος της οθόνης του το μήνυμα.

Contact Email Added

6.15 Διεπαφή Χρήστη Στατιστικών Περιοχής - RegionStatisticsActivity



Εικόνα 68: Διεπαφή Αριθμητικών Μεγεθών Περιοχής



Σημαντική διεπαφή αποτελεί αυτή της εμφάνισης των αριθμητικών μεγεθών που προκύπτουν από την ανάλυση των δεδομένων θερμοκρασίας και φωτεινότητας. Τα μεγέθη που υπολογίζονται και εμφανίζονται είναι η μέση θερμοκρασία και φωτεινότητα, όπως και οι μέγιστες και ελάχιστες τιμές ανά μήνα, εβδομάδα και ημέρα. Το κουμπι με την επιγραφή **“Statistics”** εμφανίζεται



στις καρτέλες “Data” και “Movement” που έχουν παρουσιαστεί και νωρίτερα.



Bedroom: Entered 5 time(s) today



Επίσης πάνω από τα αριθμητικά μέγεθη που προαναφέραμε εμφανίζεται και το σύνολο των φορών που ο χρήστης εισήλθε στην επιλεγμένη περιοχή την εκάστοτε ημέρα.

6.16 Διαδικτυακή Εφαρμογή

Όπως αναφέρθηκε και παραπάνω δημιουργήθηκε και μια διαδικτυακή εφαρμογή για την απεικόνιση των δεδομένων που λαμβάνουμε από τις συσκευές BLE (beacons & Puck.js) και την μελέτη αυτών. Η σελίδα φιλοξενείται μέσω της πλατφόρμας Github, όπου για να επιτευχθεί αυτό χρειάζεται να έχουμε λογαριασμό στην επίσημη σελίδα της. Στον παρακάτω σύνδεσμο εμφανίζονται ένα προς ένα τα βήματα τα οποία και ακολουθήσαμε για να δημιουργήσουμε τη σελίδα της διαδικτυακής εφαρμογής.

<https://pages.github.com/>

Η διεύθυνση είναι η: <https://diam0ndr.github.io/exemplerepo/index.html>

Secure | <https://diam0ndr.github.io/exemplerepo/index.html> ☆

Login Page

Login to Account

Εικόνα 69: Web App Login Page

Στη σελίδα σύνδεσης με την διαδικτυακή εφαρμογή ο χρήστης καλείται να συμπληρώσει τα δύο (2) υποχρεωτικά πεδία που εμφανίζονται, δηλαδή την ηλεκτρονική του διεύθυνση(email) και τον κωδικό πρόσβασης του (password) που εισήγαγε κατά την εγγραφή του στην κινητή (mobile) εφαρμογή.

Υπάρχουν δύο πιθανά μηνύματα που εμφανίζονται σε περίπτωση που ο χρήστης εισάγει λάθος στοιχεία σε κάποιο από τα πεδία. Το πρώτο μήνυμα αφορά στην πληκτρολόγηση λανθασμένο κωδικού πρόσβασης.

diam0ndr.github.io says

Error : The password is invalid or the user does not have a password.

OK

Εικόνα 70: Μήνυμα Απόρριψης Εισόδου/Λανθασμένος Κωδικός Πρόσβασης

diam0ndr.github.io says

Error : There is no user record corresponding to this identifier. The user may have been deleted.

OK

Εικόνα 71: Μήνυμα Απόρριψης Εισόδου/Λανθασμένη Ηλεκτρονική Διεύθυνση

Το δεύτερο μήνυμα αφορά στην πληκτρολόγηση μη υπάρχουσας ή λανθασμένης ηλεκτρονικής διεύθυνσης. Στην περίπτωση που θα συμβεί αυτό θα εμφανιστεί το αυτόματο μήνυμα από τη βάση Cloud Firestore, το οποίο αναφέρει πως δεν υπάρχει χρήστης με αυτό το αναγνωριστικό ή πως υπάρχει περίπτωση να έχει διαγραφεί το προφίλ του.

6.17 Σελίδα Δεδομένων Συσκευής Puck.js

Puck.js Project

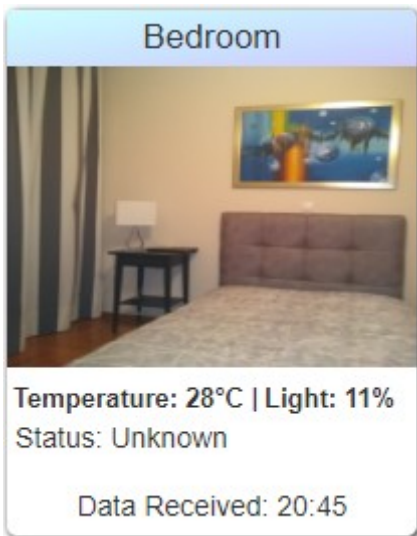
Puck Data

Στη σελίδα αυτή εμφανίζονται σε καρτέλες όλα τα έγγραφα της συλλογής **PuckData**, που αποθηκεύτηκαν την εκάστοτε ημέρα. Στο επάνω και δεξιά μέρος της εμφανίζεται η ημερομηνία της εκάστοτε ημέρας. Στην καρτέλα εμφανίζονται τα εξής δεδομένα: Το όνομα της περιοχής/χώρου που υπάρχει μια συσκευή beacon, η τιμή της θερμοκρασίας, η τιμή της φωτεινότητας, η κατάσταση του χρήστη (user feels good, user needs help ή unknown) και τέλος στο κάτω μέρος εμφανίζεται η ακριβής ώρα της ημέρας, όπου ελήφθησαν τα δεδομένα.

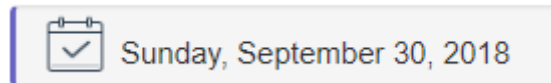
The screenshot shows the 'Puck Data' page for the 'Puck.js Project'. The page includes a navigation bar with 'Events' and 'Puck Data' tabs, a 'Live Region' indicator, and a user profile 'diamond@gmail.com' with a 'Log Out' button. The main content area displays a grid of 12 data cards for Sunday, September 30, 2018. Each card shows a room name, a small image of the room, and sensor data including temperature, light level, and user status.

Room	Temperature (°C)	Light (%)	Status	Data Received
Bedroom	30	6	User needs help	20:49
Bedroom	28	6	User needs help	20:45
Bedroom	28	11	Unknown	20:45
Bathroom	29	10	Unknown	18:34
Bedroom	33	9	Unknown	17:40
Bedroom	33	9	Unknown	17:39
Bathroom	33	10		
Bedroom	31	9		
Bedroom	33	10		
Bathroom	33	10		
Bedroom	33	6		
Bedroom	33	11		

Εικόνα 72: Σελίδα Δεδομένων Συσκευής Puck.js



Εικόνα 73: Καρτέλα Δεδομένων Puck.js






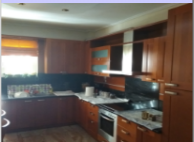


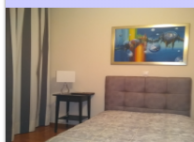
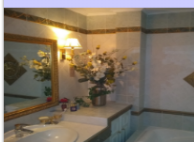
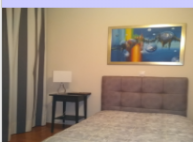
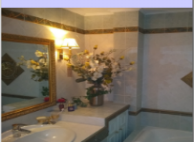


Εικόνα 74: Ημερομηνία της εκάστοτε ημέρας

6.18 Σελίδα Δεδομένων Συσκευών Beacons

Puck.js Project Events

Puck.js Project Events Puck Data
Live Region
diamond@gmail.com Log Out

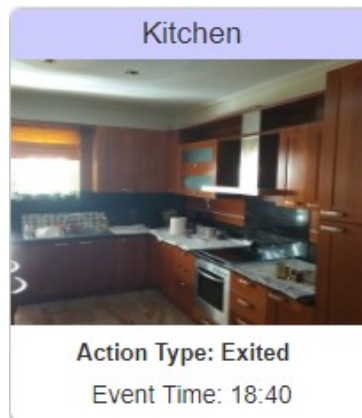
Sunday, September 30, 2018

Bathroom  Action Type: Exited Event Time: 20:54	Bathroom  Action Type: Exited Event Time: 20:53	Bathroom  Action Type: Entered Event Time: 20:44	Kitchen  Action Type: Exited Event Time: 19:21	Bathroom  Action Type: Entered Event Time: 19:16	Kitchen  Action Type: Entered Event Time: 19:16
Bedroom  Action Type: Exited Event Time: 19:15	Bathroom  Action Type: Exited Event Time: 19:14	Bedroom  Action Type: Entered Event Time: 19:09	Bathroom  Action Type: Entered Event Time: 19:09	Kitchen  Action Type: Entered Event Time: 19:09	Bathroom  Action Type: Entered Event Time: 19:07

Εικόνα 75: Σελίδα Δεδομένων Δεδομένων Συσκευών Beacons

Στη σελίδα αυτή εμφανίζονται σε καρτέλες όλα τα έγγραφα της συλλογής **Region-Events**, που αποθηκεύτηκαν την εκάστοτε ημέρα. Στο επάνω και δεξιά μέρος της εμφανίζεται η ημερομηνία της εκάστοτε ημέρας. Στην καρτέλα εμφανίζονται τα εξής δεδομένα:

Το όνομα της περιοχής/χώρου που υπάρχει μια συσκευή beacon, το είδος της ενέργειας δραστηριότητας του χρήστη, δηλαδή για το αν εισήλθε ή εξήλθε από την εκάστοτε περιοχή και τέλος στο κάτω μέρος εμφανίζεται η ακριβής ώρα της ημέρας, όπου πραγματοποιήθηκε η ενέργεια αυτή.



Εικόνα 76: Καρτέλα Δεδομένων Beacon

6.19 Σελίδα Παρακολούθησης Ζωντανής Περιοχής

Puck.js Project [Live Region](#)

The screenshot shows the Puck.js Project web interface. At the top, there are navigation links for 'Events', 'Puck Data', and 'Live Region'. The user's email 'diamond@gmail.com' and a 'Log Out' button are visible. The main content area features a user profile card for 'Diamantis Rodopoulos' with a 'Current Location: Unknown' status. Below the profile are three room cards: 'Bathroom' (Beacon Id: 0IEx), 'Kitchen' (Beacon Id: HBx9), and 'Bedroom' (Beacon Id: yEB9). Each room card includes a photo and a description.

Εικόνα 77: Σελίδα Παρακολούθησης Ζωντανής Περιοχής

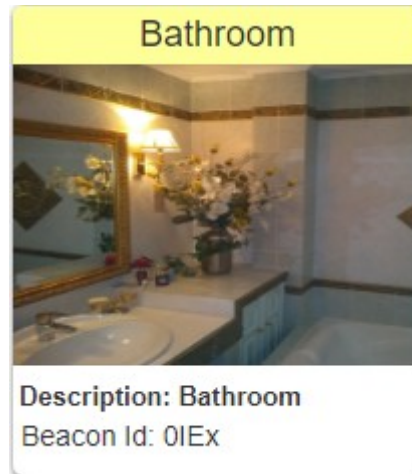
Σε αυτή τη σελίδα εμφανίζονται οι τρεις περιοχές τις οποίες έχει δημιουργήσει ο χρήστης μέσω της Android εφαρμογής. Κάτω από το μενού εμφανίζεται η καρτέλα με τη φωτογραφία και το ονοματεπώνυμο του χρήστη, καθώς και της τωρινής τοποθεσίας όπου βρίσκεται. Αν δε βρίσκεται σε κάποια από τις περιοχές που υπάρχει κάποια συσκευή beacon τότε η τοποθεσία θα είναι άγνωστη - “Current Location: Unknown”

A close-up of the user profile card showing a photo of a man with sunglasses, his full name 'Diamantis Rodopoulos', and his current location 'Unknown'.

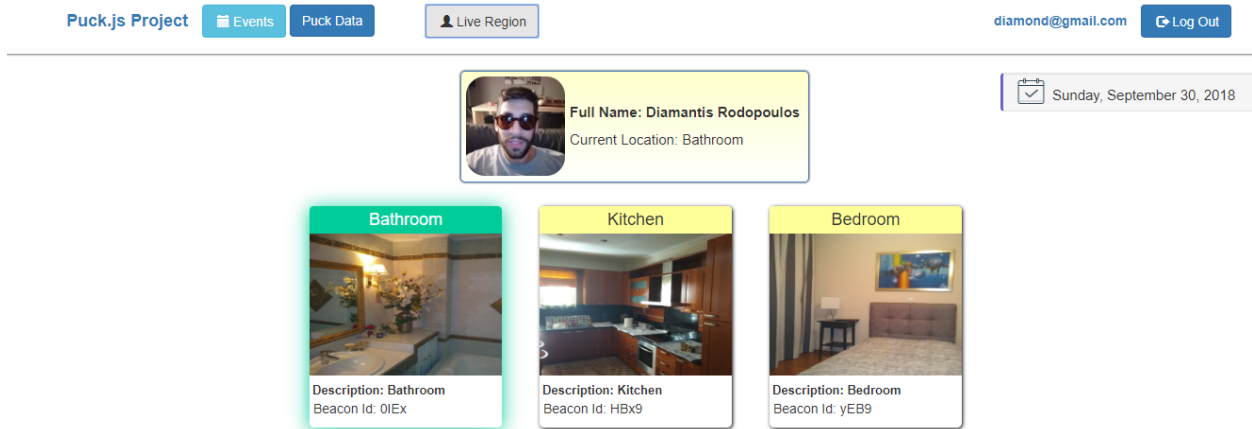
Εικόνα 78: Καρτέλα Στοιχείων χρήστη και τωρινή περιοχή

Στην καρτέλα της περιοχής εμφανίζονται τα εξής δεδομένα:

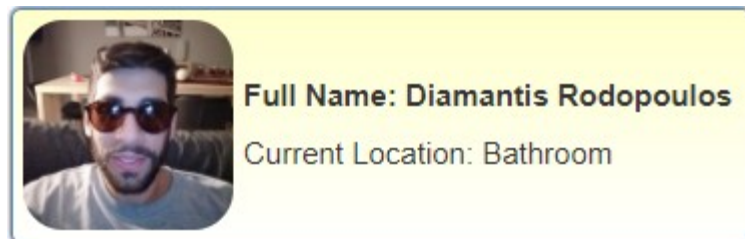
Το όνομα της περιοχής/χώρου, η περιγραφή του το είδος της ενέργειας δραστηριότητας του χρήστη, δηλαδή για το αν εισήλθε ή εξήλθε από την εκάστοε περιοχή και τέλος στο κάτω μέρος εμφανίζεται η ακριβής ώρα της ημέρας, όπου πραγματοποιήθηκε η ενέργεια αυτή.



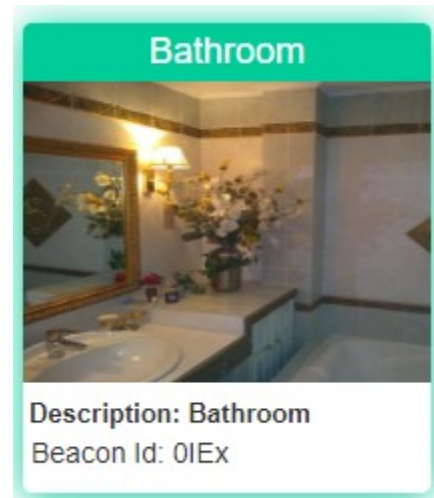
Εικόνα 79: Καρτέλα Στοιχείων περιοχής/χώρου



Εικόνα 80: Παρουσίαση κατάστασης ζωντανής τοποθεσίας



Εικόνα: 80: Καρτέλα στοιχείων χρήστη - Live Region



Εικόνα: 81: Live Region

Όταν η τιμή του πεδίου κατάστασης(status) αλλάξει από “default” σε **“access”**, δηλαδή `status: "access"` όταν ο χρήστης εισέλθει στην εκάστοτε περιοχή τότε η αντίστοιχη καρτέλα της περιοχής θα εμφανίζεται με πράσινο χρώμα και το όνομα της περιοχής θα εμφανίζεται και στην καρτέλα με τα στοιχεία του χρήστη - “Current Location: Bathroom”

ΚΕΦΑΛΑΙΟ 7. ΕΠΙΛΟΓΟΣ

7.1 ΣΥΝΟΨΗ

Στα πλαίσια της διπλωματικής εργασίας αναπτύχθηκε η εφαρμογή ADL App - Activities of Daily Living με την οποία γίνεται η καταγραφή καθημερινών στοιχείων του χρήστη που αφορούν στην κίνηση του στο χώρο όπου κατοικεί. Με πιο απλά λόγια καταγράφονται και αποθηκεύονται δεδομένα που έχουν να κάνουν με τη καθημερινή δραστηριότητα του χρήστη, όπως θερμοκρασία, φωτεινότητα στο χώρο στον οποίο βρίσκεται και τη συγκεκριμένη χρονική στιγμή που εκείνος εισέρχεται/εξέρχεται στον/ή από τον εκάστοτε χώρο. Βασική προϋπόθεση είναι ο χρήστης να έχει τις περισσότερες ώρες της ημέρας τη συσκευή Puck στην τσέπη του ή σε κάποια κοντινή απόσταση από εκείνος, ώστε να μπορεί πατώντας το κουμπί να δηλώσει την κατάσταση του (-Ο χρήστης νιώθει καλά ή -ο χρήστης βρίσκεται σε κατάσταση ανάγκης και χρειάζεται άμεση βοήθεια). Βασικά χαρακτηριστικά της εφαρμογής αποτελούν τα εξής παρακάτω:

- Η εφαρμογή παρέχει τη δυνατότητα καταγραφής σημαντικών δεδομένων που προέρχονται από τις συσκευές Bluetooth χαμηλής ενέργειας, όπως θερμοκρασία.
- Ο χρήστης μπορεί προσθέσει σημειώσεις για κάθε χώρο που έχει δηλώσει κατά την εγγραφή του στην εφαρμογή, έτσι ώστε να μπορεί να διαχειρίζεται καλύτερα τις καθημερινές του ενέργειες/δραστηριότητες που αφορούν σε κάποιες προγραμματισμένες δράσεις του στον εκάστοτε χώρο.
- Η εφαρμογή παρέχει επίσης τη δυνατότητα στο χρήστή να μπορεί να εκπέμψει σήμα έκτακτης ανάγκης με το απλό πάτημα ενός κουμπιού. Μπορεί να αποθηκεύσει όσες ηλεκτρονικές διεθύνσεις αγαπημένων του προσώπων επιθυμεί, ώστε να λάβουν όλοι e-mail με το μήνυμα του κινδύνου που διατρέχει ο χρήστης και σε ποιο χώρο έγινε αυτό.
- Μπορεί εύκολα να τοποθετήσει τις συσκευές beacons σε οποιοδήποτε χώρο επιθυμεί και να ορίσει ο ίδιος τις τοποθεσίες του και το όνομα αυτών.
- Η εφαρμογή με χρήση διαγραμμάτων απεικονίζει τις αλλαγές στα επίπεδα θερμοκρασίας και φωτεινότητας σύμφωνα με την καθημερινή δραστηριότητα του χρήστη σε κάθε χώρο του τόπου όπου κατοικεί.
- Ο χρήστης μπορεί να δηλώσει την κατάστασή στην οποία βρίσκεται πατώντας το ενσωματωμένο κουμπί της συσκευής Puck.
- Η καταχώρηση και αποθήκευση των εσωτερικών τοποθεσιών του χρήστη μπορεί να αλλάξει ανα πάσα στιγμή, αρκεί αυτός να ακολουθήσει τη διαδικασία καταχώρησης μιας περιοχής η οποία διαρκεί μερικά δευτερόλεπτα. Η κάθε συσκευή beacon αντιστοιχεί αυστηρά σε μόνο μια περιοχή.

- Μπορεί επίσης να αλλάξει τα στοιχεία των ηλεκτρονικών διευθύνσεων των συγγενικών του προσώπων, να προσθέσει ή να διαγράψει κάποια από αυτές που έχει ήδη δηλώσει.
- Στην εφαρμογή παρουσιάζονται στατιστικά στοιχεία, όπως: ο αριθμός των φορών που ο χρήστης εισήλθε σε κάποια από τις περιοχές, ημερήσια μέγιστη και ελάχιστη θερμοκρασία και φωτεινότητα, εβδομαδιαία και μηνιαία. Επιπρόσθετα υπολογίζεται και εμφανίζεται και ο μέσος όρος αυτών των δύο τιμών.
- Η συσκευή Puck είναι προγραμματισμένη ώστε ανα πέντε (5) λεπτά και για 5 δευτερόλεπτα ανα φορά, να μπαίνει σε λειτουργία ανίχνευσης για συσκευές beacons έτσι ώστε να στέλνει τα δεδομένα και αυτά να καταχωρούνται και να αποθηκεύονται στη βάση δεδομένων.
- Σημαντικό στοιχείο της εφαρμογής για την αναγνώριση της καθημερινής δραστηριότητας του χρήστη είναι η καταχώρηση και αποθήκευση της ώρας κατά την οποία εκείνος εισήλθε ή εξήλθε σε/από έναν από τους χώρους στους οποίους είναι εγκατεστημένη μία συσκευή beacon.

7.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Εν κατακλείδι, θα γίνει μια αναφορά στις πιθανές νέες προσθήκες διάφορων λειτουργιών που μπορούν να ενσωματωθούν στην εφαρμογή, όπως και οι περαιτέρω κατευθύνσεις που μπορεί να ακολουθήσει η εφαρμογή στον μέλλον.

Η πρώτη σημαντική λειτουργία που θα μπορούσε να ενσωματωθεί αφορά στις επιπλέον έξυπνες συσκευές που θα μπορούσαν να συμπεριληφθούν, ώστε οι ροές δεδομένων να προκύπτουν από διαφορετικές και τους τύπους δεδομένων που συλλέγει η εφαρμογή.

Σημαντική λειτουργία που έχει μελετηθεί και μπορεί να εφαρμοστεί μελλοντικά, είναι εκείνη της συνεχούς ανίχνευσης για τυχόν μακροχρόνια παύση της κίνησης του χρήστη σε χώρους-κλειδιά εντός της εσωτερικής περιοχής όπου διαμένει. Για παράδειγμα θα μπορούσε να ανιχνευθεί παύση στην κίνηση για διάστημα μεγαλύτερο των 10-15 λεπτών στο μπάνιο του χρήστη ή υψηλές θερμοκρασίες στην κουζίνα του, με αποτέλεσμα την άμεση ενημέρωση για τις ενέργειες αυτές.

Ταυτόχρονα, υπάρχουν σημαντικά περιθώρια επέκτασης της στατιστικής ανάλυσης που επιτελεί η εφαρμογή. Στη mobile εφαρμογή υπολογίζονται και εμφανίζονται αριθμητικά μεγέθη όπως η μέση τιμή, η μέγιστη και ελάχιστη τιμή θερμοκρασίας και φωτεινότητας, καθώς και το σύνολο των φορών που εισήχθη ο χρήστης σε κάθε περιοχή του χώρου διαμονής του. Σίγουρα οι παραπάνω τιμές αποτελούν χρήσιμες ενδείξεις για τον χρήστη, όμως η ανάλυση μπορεί να προχωρήσει παρακάτω, πόσο μάλλον, αν ληφθεί υπόψη και η πιθανή συλλογή περισσότερων και διαφορετικών δεδομένων, όπως αναφέρθηκε παραπάνω. Μάλιστα, η εφαρμογή θα μπορούσε να αξιοποιήσει μεθόδους και εργαλεία, που έχουν προκύψει από το πεδίο της Μηχανικής Μάθησης (Machine Learning), ώστε η ανάλυση των δεδομένων να πραγματοποιηθεί βαθύτερα.

ΠΗΓΕΣ - ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] «What is the Internet of Things, and how does it work?». *Internet of Things blog*. 2016-11-17
- [2] Alessandro Bassi et al. «Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model» Springer Open. 2013
- [3] IoT. Wikipedia 2015 https://en.wikipedia.org/wiki/Internet_of_Things
- [4] Joe Decuir, “Bluetooth 4.0: Low Energy”, CSR plc, 2010
- [5] <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>
- [6] <https://www.bluetooth.com/specifications/gatt/generic-attributes-overview>
- [7] <https://www.bluetooth.com/specifications/gatt/services>
- [8] *"Bluetooth Tutorial - Profiles"*. palowireless Pty Ltd. Archived from *the original* on 2013-10-05. Retrieved 2007-01-05.
- [9] *"Adopted Specifications | Bluetooth Technology Website"*. www.bluetooth.com. Retrieved 2016-01-15.
- [10] <https://firebase.google.com/docs/firestore/>
- [11] Foss, Clive (1991). "Beacon". In Kazhdan, Alexander. *The Oxford Dictionary of Byzantium*. Oxford University Press. pp. 273–274.
- [12] Ritchie, Leitch (1835). *Scott and Scotland*. London: Longman, Rees, Orme, Brown, Green, and Longman, p. 53
- [13] Els almogàvers a la frontera amb el sarraïns en el segle XIV. Maria Teresa Ferrer
- [14] Bullough, John; Nicholas P Skinner (December 2009). "Evaluation of Light-Emitting Diode Beacon Light Fixtures" (PDF). *Lighting Research Center – Rensselaer Polytechnic Institute*. Archived from the original (PDF) on 2013-07-29. Retrieved 2010-06-05.

