

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Διπλωματική Εργασία

«Σχεδιασμός (κινητής) εφαρμογής για έξυπνες πόλεις
αξιοποιώντας τεχνολογίες IoT»

Ανδρέας Καψάλης

ΑΜ: ΠΚΜ 202012

Επιβλέπουσα καθηγήτρια: Ελένη Χριστοπούλου

Κέρκυρα, Φεβρουάριος, 2022



«Σχεδιασμός (κινητής) εφαρμογής για έξυπνες πόλεις
αξιοποιώντας τεχνολογίες IoT»

Ανδρέας Καψάλης

ΑΜ ΠΚΜ202012

Τριμελής επιτροπή

Ανδρόνικος Θεόδωρος

Ρίγγας Δημήτριος

Ελένη Χριστοπούλου

Αναπληρωτής Καθηγητής

ΕΔΙΠ

ΕΔΙΠ

Τμήμα Πληροφορικής

Τμήμα Πληροφορικής

Τμήμα Πληροφορικής

Ιόνιο Πανεπιστήμιο

Ιόνιο Πανεπιστήμιο

Ιόνιο Πανεπιστήμιο

Κέρκυρα, Φεβρουάριος, 2022

Θα ήθελα να εκφράσω τις ευχαριστίες μου στην κα. Χριστοπούλου , η οποία με καθοδηγούσε και με συμβούλευε καθ'όλη την διάρκεια της περιόδου που εκτελούσα την διπλωματική μου εργασία. Ακόμη θα ήθελα να εκφράσω την αγάπη μου στους ανθρώπους που γνώρισα κατά την διάρκεια των μεταπτυχιακού κύκλου σπουδών και να τους ευχαριστήσω για τις όμορφες αλλά και δύσκολες στιγμές που μοιραστήκαμε.

Τέλος, θέλω ολόψυχα, μέσα από την καρδιά μου να πω ένα τεράστιο ευχαριστώ στην οικογένεια μου , για την υποστήριξη τους όλο αυτό το διάστημα των μεταπτυχιακών μου σπουδών ώστε να πετύχω τους στόχους μου.

Περίληψη

Η παρούσα εργασία εκπονήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος «Ψηφιακές Εφαρμογές και Καινοτομία» του τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου.

Στόχος της εργασίας είναι η δημιουργία μιας εφαρμογής για έξυπνες πόλεις αξιοποιώντας τεχνολογίες IoT. Για την ολοκληρωμένη επίτευξη του στόχου αναπτύχθηκε η mobile έκδοση της εφαρμογής, η web έκδοση της εφαρμογής και συσκευές IoT.

Μέσα από τη mobile εφαρμογή που υλοποιήθηκε, οι χρήστες μπορούν να καταγράφουν σε χάρτη διάφορα καθημερινά πρόβλημα στην πόλη ώστε να ενημερωθεί η αρμόδια υπηρεσία του Δήμου και να επιληφθεί για την επίλυσή τους, ενώ από την web εφαρμογή μπορούν να ενημερωθούν για τα προβλήματα που έχουν ήδη αναφερθεί.

Αξιοποιώντας τις τεχνολογίες IoT οι χρήστες μπορούν να παρακολουθούν σε πραγματικό χρόνο τις περιβαλλοντικές συνθήκες στην πόλη. Και πάλι με την βοήθεια της τεχνολογίας IoT οι χρήστες θα μπορούν να βλέπουν σε χάρτη τους διαθέσιμους σταθμούς στάθμευσης καθώς και τις διαθέσιμες σε αυτούς θέσεις. Μπορούν ακόμη να ενημερωθούν για Νέα, Ανακοινώσεις, Εκδηλώσεις που αφορούν την πόλη τους.

Με την τεράστια διείσδυση της χρήσης των έξυπνων τηλεφώνων στην ζωή μας, θεωρώ ότι πλέον μια «έξυπνη πόλη» επιβάλλεται να «υιοθετήσει» μια mobile εφαρμογή με την οποία αφενός οι πολίτες θα συμμετέχουν ενεργά στην αντιμετώπιση των καθημερινών προβλημάτων στην πόλη και αφετέρου θα μπορούν να επικοινωνούν και θα ενημερώνονται άμεσα για διάφορα θέματα της πόλης τους.

Η mobile εφαρμογή αναπτύχθηκε με την χρήση της πλατφόρμα ανάπτυξης Android εφαρμογών KODULAR, μαζί με την χρήση της cloud βάσης δεδομένων Firebase ενώ για την WEB εφαρμογή χρησιμοποιήθηκαν η πλατφόρμα διαχείρισης περιεχομένου WordPress μαζί με κώδικα Html και JavaScript. Για τα τμήματα IoT χρησιμοποιήθηκαν μικροελεγκτές ESP8266 με τον αισθητήρα θερμοκρασίας- υγρασίας DHT22 καθώς και υπέρυθρους αισθητήρες ανίχνευσης κίνησης HC-SR501.

Λέξεις – Κλειδιά

Έξυπνη πόλη, Android, Αισθητήρες, IoT, Kodular, JavaScript

Abstract

This project was carried out during the postgraduate courses: “Digital application and Innovations” of the Department of Informatics of the Ionian University.

The aim of this work is to create a Mobile app and a Web app, for smart cities utilizing IoT technologies.

With mobile application, users can report various daily issues/problems in their city to inform the competent service of the Municipality to manage their solution, while with web app, they can be informed about the problems that have already been mentioned. Utilizing IoT technologies, users can see real-time the environmental conditions in the city. Also, with IoT technology users will be able to see on map the available parking lots as well as the spots available on them. They can also be informed about News, Announcements, Events about their city.

As the huge use of smart phones in our lives, I believe that a "smart city" requires to adopt a mobile application with which citizens will actively participate in dealing with everyday problems in their city and at the same time they will communicate and be informed about various issues. This mobile application was developed using the platform KODULAR for Android application development in combine with Firebase. The WEB application was developed with Cms WordPress, Html and JavaScript. About the IoT section, used the ESP8266 microcontrollers with the DHT22 temperature-humidity sensor and HC-SR501 PIR sensor.

Keywords

Smart City, Android, Sensors, IoT, Kodular, JavaScript.

Περιεχόμενα

Περίληψη.....	ii
Abstract	iii
Περιεχόμενα	iv
Κατάλογος Εικόνων / Σχημάτων.....	vii
Κατάλογος Πινάκων.....	x
1. Εισαγωγή	1
1.1 Σκοπός Διπλωματικής εργασίας.....	1
2. Έξυπνη Πόλη	4
2.1 Ορισμός της έννοιας της «Έξυπνης Πόλης» (Smart City)	4
2.2 Χαρακτηριστικά της «Έξυπνης Πόλης»	5
2.3 Έξυπνες πόλεις στην Ευρώπη.....	9
2.4 Έξυπνες Πόλεις στην Ελλάδα.....	14
3. Το Διαδίκτυο των Πραγμάτων (Internet of Things).....	19
3.1. Η ανάπτυξη του Διαδικτύου (Internet).....	19
3.2. Εισαγωγή στο Διαδίκτυο των πραγμάτων (IoT)	20
3.3. Πλεονεκτήματα του IoT	22
3.4. Μειονεκτήματα του IoT	24
3.5. Αρχιτεκτονική IoT.....	25
3.6. Επικοινωνία IoT.....	25
3.6.1. Device to Device	26
3.6.2. Device to Cloud.....	27
3.6.3. Device to Gateway.....	27
3.6.4. Back-End Data-Sharing	28
3.7. Αξιοποίηση του IoT στις «Έξυπνες Πόλεις».....	28
4. Εφαρμογές για Έξυπνες Πόλεις	31
4.1. Smartphones και εφαρμογές.....	31
4.2. Εφαρμογές «Έξυπνων Πόλεων».....	33
4.2.1. Novonville.....	33
4.2.1.1. Ενδεικτικές οθόνες εφαρμογής.....	35
4.2.1.2. Αξιολόγηση εφαρμογής.....	36
4.2.2. SenceCity	36

4.2.2.1.	Ενδεικτικές οθόνες εφαρμογής.....	38
4.2.2.2.	Αξιολόγηση εφαρμογής.....	39
5.	Ανάλυση και σχεδίαση Mobile εφαρμογής «MySmartCity».....	40
5.1.	Εργαλεία ανάπτυξης της mobile εφαρμογής	42
5.1.1.	Kodular.....	42
5.1.2.	Google Map Api.....	43
5.1.3.	Firebase.....	46
5.1.3.1.	Δημιουργία Firebase Project	46
5.1.3.2.	Firebase Authentication.....	48
5.1.3.3.	Realtime Database	49
5.1.3.4.	Storage.....	50
5.1.4.	OneSignal.....	51
5.2.	Υλοποίηση της mobile εφαρμογής	53
5.2.1.	Screen 1.....	53
5.2.2.	Login Screen	54
5.2.3.	Report Screen.....	58
5.2.3.1.	Κάνε μια αναφορά	61
5.2.3.2.	Λίστα Αναφορών.....	69
5.2.3.3.	Περιβάλλον.....	70
5.2.3.4.	Parking	71
5.2.3.5.	Εκδηλώσεις	72
5.2.3.6.	Νέα-Ανακοινώσεις.....	73
5.2.3.7.	Σχετικά με τον Δήμο	73
6.	Ανάλυση και σχεδίαση της Web εφαρμογής “mysmartcity.online”	76
6.1.	Εργαλεία ανάπτυξης της web εφαρμογής	76
6.2.	Υλοποίηση της web εφαρμογής.....	78
6.2.1.	Home	78
6.2.2.	Χάρτης αναφορών	80
6.2.3.	Θέσεις Στάθμευσης.....	81
6.2.4.	Νέα-Ανακοινώσεις.....	82
6.2.4.1.	Εκδηλώσεις	82
6.2.4.2.	Ο Δήμος μας	83
6.2.5.	Διαχείριση εφαρμογής.....	83

6.2.5.1.	Δημιουργία Posts (Ανακοινώσεις – Νέα, Events, Δήμος).....	83
6.2.5.2.	Διαχείριση Αναφορών	84
6.2.5.2.1.	Αναφορές	85
7.	Επίγνωση πλαισίου αξιοποιώντας το Διαδίκτυο των Πραγμάτων.....	86
7.1.	Μέτρηση περιβαλλοντικών συνθηκών.	86
7.1.1.	Το Υλικό (Hardware).....	87
7.1.1.1.	ESP-82666 (NodeMcu).....	87
7.1.1.2.	DHT22.....	89
7.1.1.3.	Lipo Charger DFR0264.....	89
7.1.1.4.	Solar Panel.....	91
7.1.2.	Το Λογισμικό (Software)	91
7.1.2.1.	Περιβάλλον προγραμματισμού Arduino IDE.....	91
7.1.3.	Υλοποίηση συσκευής IoT	94
7.2.	Θέσεις στάθμευσης	95
7.2.1.	Το Υλικό (Hardware).....	96
7.2.1.1.	HC-SR501	98
7.2.2.	Το Λογισμικό (Software)	99
7.2.2.1.	Περιβάλλον προγραμματισμού Arduino IDE.....	99
8.	Αξιολόγηση.....	101
8.1.	Περιγραφή διαδικασίας αξιολόγησης.....	102
8.2.	Αποτελέσματα αξιολόγησης.....	106
9.	Συμπεράσματα	107
9.1.	Μελλοντικές επεκτάσεις.....	108
	Βιβλιογραφία.....	109
	Παράρτημα Α: «Ο Block Κώδικας της Mobile εφαρμογής».....	114
	Παράρτημα Β: «Κώδικας Web εφαρμογής »	120
	Παράρτημα Γ: «Ο Κώδικας της IoT συσκευής θερμοκρασίας -υγρασίας»	134
	Παράρτημα Δ: «Ο Κώδικας της IoT συσκευής μέτρησης θέσεων στάθμευσης»	135

Κατάλογος Εικόνων / Σχημάτων

Εικόνα 1 Διασύνδεση των επιμέρους τμημάτων.....	2
Εικόνα 2 Το μοντέλο του τροχού για την έξυπνη πόλη	6
Εικόνα 3 Διαστάσεις Έξυπνης Πόλης	7
Εικόνα 4 Έξυπνη Πόλη Βαρκελώνη.....	9
Εικόνα 5 Άμστερνταμ Έξυπνη Πόλη.....	11
Εικόνα 6 Κοπεγχάγη Έξυπνη Πόλη.....	13
Εικόνα 7 Τρίκαλα Έξυπνη Πόλη».....	15
Εικόνα 8 Ιωάννινα «Έξυπνη πόλη».....	17
Εικόνα 9 Internet of Things	21
Εικόνα 10 Πρωτόκολλα επικοινωνίας IoT	26
Εικόνα 11 Κατανομή των (IoT) σε όλο τον κόσμο	29
Εικόνα 12 Αισθητήρες στα Smartphone	32
Εικόνα 13 Η εφαρμογή Novonville	33
Εικόνα 14 Ενδεικτικές Οθόνες Novonville (a).....	35
Εικόνα 15 Ενδεικτικές Οθόνες Novonville (b)	35
Εικόνα 16 Αξιολόγηση εφαρμογής Novonville	36
Εικόνα 17 Η εφαρμογή SenceCity.....	36
Εικόνα 18 Ενδεικτικές Οθόνες SenceCity (a)	38
Εικόνα 19 Ενδεικτικές Οθόνες SenceCity (b)	38
Εικόνα 20 Αξιολόγηση εφαρμογής SenceCity.....	39
Εικόνα 21 Η πλατφόρμα Kodular	42
Εικόνα 22 Παραμετροποίηση Kodular	43
Εικόνα 23 Google Cloud Console.....	43
Εικόνα 24 Δημιουργία Project.....	44
Εικόνα 25 Ενεργοποίηση Maps JavaScript Api.....	44
Εικόνα 26 Δημιουργία API Key	45
Εικόνα 27 Δικαιώματα εφαρμογών στο API.....	45
Εικόνα 28 Δημιουργία Firebase Project.....	46
Εικόνα 29 Δημιουργία Credentials.....	47
Εικόνα 30 Web app registration.....	47
Εικόνα 31 Firebase Authentication.....	48

Εικόνα 32 Παραμετροποίηση Authenticator	49
Εικόνα 33 Authenticated Users.....	49
Εικόνα 34 Realtime Database	50
Εικόνα 35 Storage.....	51
Εικόνα 36 OneSignal.....	51
Εικόνα 37 Παραμετροποίηση OneSignal.....	52
Εικόνα 38 Αποστολή (Push) μηνυμάτων	52
Εικόνα 39 Η οθόνη Screen 1	53
Εικόνα 40 Ενδεικτικές οθόνες της Screen 1	54
Εικόνα 41 Login Screen.....	54
Εικόνα 42 Ενδεικτικές οθόνες Login Screen (a)	55
Εικόνα 43 Ενδεικτικές οθόνες Login Screen (b).....	56
Εικόνα 44 Έλεγχος αυθεντικοποίησης	56
Εικόνα 45 Λήψη συντεταγμένων	57
Εικόνα 46 Υπολογισμός της θέσης των Image Sprite.....	57
Εικόνα 47 Περιστροφή των στοιχείων του Μενού	57
Εικόνα 48 Μετάβαση στη οθόνη Report.....	58
Εικόνα 49 Report Screen.....	58
Εικόνα 50 Ενδεικτικές οθόνες (a).....	59
Εικόνα 51 Ενδεικτικές οθόνες (b)	60
Εικόνα 52 Ενδεικτικές οθόνες (c)	60
Εικόνα 53 Επιλογή «Κάνε μια Αναφορά»	61
Εικόνα 54 Συμπίεση φωτογραφίας	62
Εικόνα 55 Μεταφόρτωση στο Storage.....	62
Εικόνα 56 Αποθήκευση στην Realtime Database	63
Εικόνα 57 Δομή Realtime Database	63
Εικόνα 58 Δημιουργία Unique ID.....	63
Εικόνα 59 Address Json object	64
Εικόνα 60 Αποκωδικοποίηση διεύθυνσης.....	65
Εικόνα 61 Δημιουργία Marker.....	65
Εικόνα 62 Έλεγχος Γεωγραφικά ορίων Δήμου (Geofence)	66
Εικόνα 63 Γεωγραφικά όρια Δήμου	66
Εικόνα 64 Αποστολή Email.....	67

Εικόνα 65 Λήψη των Tags από την Realtime Database	67
Εικόνα 66 Λήψη των συντεταγμένων των Marker	68
Εικόνα 67 Δημιουργία των Marker στον χάρτη	68
Εικόνα 68 Επιλογή «Λίστα αναφορών».....	69
Εικόνα 69 Υπολογισμός των Marker ανά κατηγορία	69
Εικόνα 70 Δυναμική δημιουργία καρτών αναφοράς	70
Εικόνα 71 Επιλογή «Περιβάλλον»	70
Εικόνα 72 Θερμοκρασία και Υγρασία στην Realtime Database.....	71
Εικόνα 73 Προβολή της Θερμοκρασίας και Υγρασίας.....	71
Εικόνα 74 Στοιχεία Parking στην Realtime Database	72
Εικόνα 75 Προβολή χάρτη Parking.....	72
Εικόνα 76 Επιλογή «Εκδηλώσεις».....	72
Εικόνα 77 Επιλογή «Νέα-Ανακοινώσεις»	73
Εικόνα 78 Επιλογή «Σχετικά με τον Δήμο»	73
Εικόνα 79 Web Κλήση	73
Εικόνα 80 Json Object	74
Εικόνα 81 Δυναμική δημιουργία κάρτας «Δήμος -Δημαρχείο».....	74
Εικόνα 82 Στοιχεία Δήμος -Δημαρχείο στην Realtime Database.....	75
Εικόνα 83 Home Page	78
Εικόνα 84 Τελευταίες αναφορές.....	79
Εικόνα 85 Last_10_ Issues	79
Εικόνα 86 Αναφορές στην Realtime Database	80
Εικόνα 87 Χάρτης αναφορών	80
Εικόνα 88 Θέσεις Στάθμευσης.....	81
Εικόνα 89 Θέσεις Στάθμευσης στην Realtime Database	81
Εικόνα 90 Νέα - Ανακοινώσεις.....	82
Εικόνα 91 Εκδηλώσεις	82
Εικόνα 92 Ο Δήμος μας.....	83
Εικόνα 93 Δημιουργία Post 's.....	84
Εικόνα 94 Αναφορές	84
Εικόνα 95 Επεξεργασία αναφορών	85
Εικόνα 96 Σχηματικό διάγραμμα ηλ. κυκλώματος IoT	86
Εικόνα 97 NodeMcu.....	88

Εικόνα 98 DHT22 Sensor	89
Εικόνα 99 Lipo Charger DFR0264	90
Εικόνα 100 Ηλιακός συλλέκτης	91
Εικόνα 101 Arduino IDE	92
Εικόνα 102 Arduino IDE Port Settings	93
Εικόνα 103 Arduino IDE Board Select	93
Εικόνα 104 Άνω όψη συσκευής IoT	94
Εικόνα 105 Εσωτερική εικόνα συσκευής IoT	94
Εικόνα 106 Κάτω όψη συσκευής IoT	95
Εικόνα 107 Έλεγχος εισόδου εξόδου αυτοκινήτων	96
Εικόνα 108 Κύκλωμα IoT μέτρησης αυτοκινήτων	97
Εικόνα 109 Ο Αισθητήρας HC-SR501	98
Εικόνα 110 HC-SR501 Ανίχνευση κίνησης	99
Εικόνα 111 Ερώτηση 1	103
Εικόνα 112 Ερώτηση 2	104
Εικόνα 113 Ερώτηση 3	104
Εικόνα 114 Ερώτηση 4	104
Εικόνα 115 Ερώτηση 5	104
Εικόνα 116 Ερώτηση 6	105
Εικόνα 117 Ερώτηση 7	105
Εικόνα 118 Ερώτηση 8	105
Εικόνα 119 Ερώτηση 9	105
Εικόνα 120 Ερώτηση 10	106

Κατάλογος Πινάκων

Πίνακας 1 Διαστάσεις της “Εξυπνης Πόλης”	6
Πίνακας 2 Τεχνικά χαρακτηριστικά NodeMcu	88
Πίνακας 3 HC-SR501 Τεχνικά χαρακτηριστικά	99
Πίνακας 4 Βαθμολογία Ευχρηστίας Συστήματος	106

1. Εισαγωγή

Ο όρος έξυπνη πόλη (έξυπνες πόλεις) έχει μπει πλέον στην καθημερινή μας ζωή. Με τον όρο αυτό εννοούμε μια πόλη που χρησιμοποιεί το δυναμικό της τεχνολογίας και της καινοτομίας, μαζί με άλλους πόρους, για την αποτελεσματικότερη προώθηση της αειφόρου ανάπτυξης και, τελικά, τη βελτίωση της ποιότητας ζωής των πολιτών της. Οι σύγχρονες πόλεις πλέον αντιμετωπίζουν τεράστιες προκλήσεις, όπως την αύξηση του πληθυσμού, τη μόλυνση του περιβάλλοντος, την αύξηση της κυκλοφοριακής κίνησης, τη μείωση των διαθέσιμων προϋπολογισμών. Το Διαδίκτυο των Πραγμάτων (IoT), τα μεγάλα δεδομένα, οι εφαρμογές για κινητά, μεταξύ άλλων καταφέρνουν να βελτιώσουν την αποτελεσματικότητα των πόλεων, εάν γνωρίζουμε πώς να τα χρησιμοποιούμε έξυπνα προσφέροντας νέες δυνατότητες που συμβάλλουν στην ενίσχυση της βιωσιμότητας, αλλά και αποδοτικότητας του αστικού περιβάλλοντος, μέσα από τη συμμετοχή των πολιτών, των δημοτικών αρχών και των τοπικών επιχειρήσεων στην καθημερινή ζωή στην πόλη.

1.1 Σκοπός Διπλωματικής εργασίας

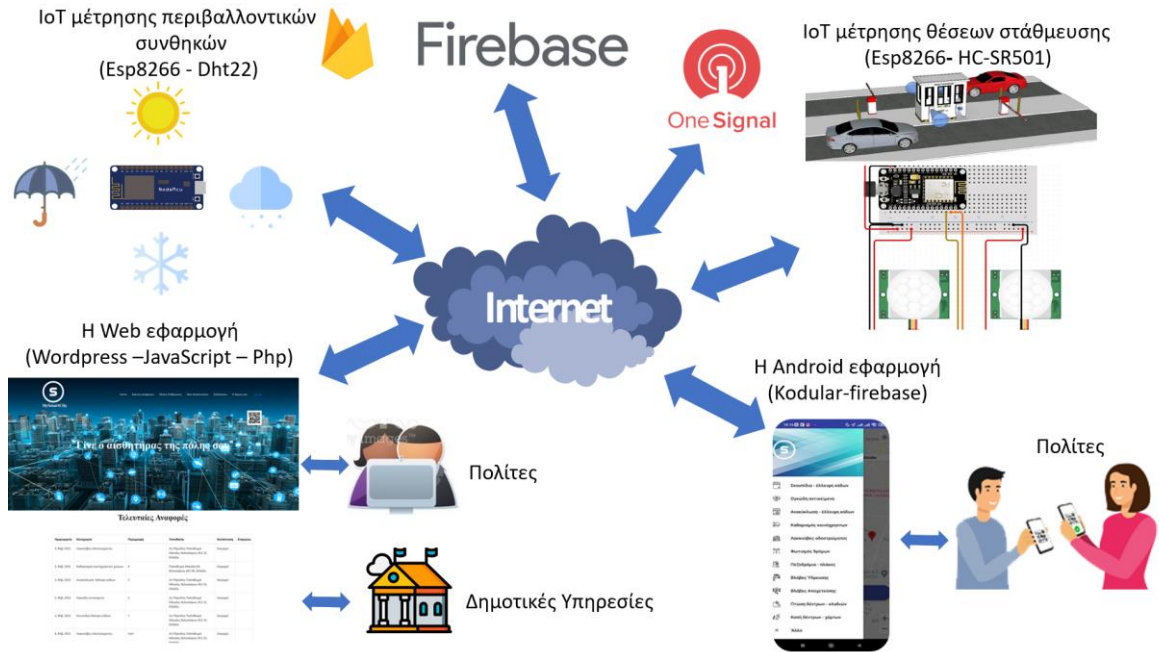
Η παρούσα διπλωματική εργασία έχει ως σκοπό τον σχεδιασμό και την υλοποίηση εφαρμογής για έξυπνες πόλεις αξιοποιώντας τεχνολογίες IoT.

Η εφαρμογή μπορεί να διαχωριστεί σε τρία επιμέρους τμήματα:

- Την mobile εφαρμογή μέσω της οποίας οι χρήστες έξυπνων συσκευών (smartphones, tablet) συνεισφέρουν, στην βελτίωση της ποιότητας ζωής των πολιτών αφού οι ίδιοι οι πολίτες «γίνονται οι αισθητήρες της πόλης» αναφέροντας μέσα από την εφαρμογή τα προβλήματα της καθημερινότητας.
- Την web έκδοση της εφαρμογής μέσω της οποίας αφενός οι πολίτες μπορούν να ενημερώνονται για καταγεγραμμένα προβλήματα, Νέα - Ανακοινώσεις και Εκδηλώσεις και αφετέρου οι διαχειριστές μπορούν να διαχειρίζονται και να ενημερώνουν της κατάσταση των αιτημάτων όπως αυτές διεκπεραιώνονται από τις αρμόδιες υπηρεσίες του Δήμου.
- Το σύστημα IoT το οποίο συλλέγει και ενημερώνει τις εφαρμογές σε τακτά χρονικά με:
 - ✓ πληροφορίες για τις περιβαλλοντικές συνθήκες (υγρασία - θερμοκρασία)

✓ πληροφορίες για τα σημεία και τις διαθέσιμες θέσεις στάθμευσης.

Οι πληροφορίες αυτές αποθηκεύονται σε μια κεντρική Cloud βάση δεδομένων και μέσω της mobile αλλά και της web έκδοση της εφαρμογής είναι διαθέσιμες στους χρήστες, συμβάλλοντας έτσι στην μείωση της κυκλοφοριακής συμφόρησης αλλά και στην ενημέρωση για τις περιβαλλοντικές συνθήκες.



Εικόνα 1 Διασύνδεση των επιμέρους τμημάτων

Η διπλωματική εργασία είναι οργανωμένη σε 9 κεφάλαια.

Το 1ο κεφάλαιο αποτελεί την παρούσα εισαγωγή, ενώ ακολουθεί επισκόπηση των κεφαλαίων που ακολουθούν:

- Στο κεφάλαιο 2 περιγράφεται η έννοια της «Έξυπνης Πόλης» και παρουσιάζονται χαρακτηριστικά παραδείγματα – υλοποιήσεις της ,τόσο στην Ευρώπη όσο και στην Ελλάδα.
- Στο κεφάλαιο 3 παρουσιάζεται η τεχνολογία IoT που χρησιμοποιήθηκε στην παρούσα εργασία ή αρχιτεκτονική της καθώς και οι εφαρμογές της στις έξυπνες πόλεις.
- Στο κεφάλαιο 4 γίνεται αναφορά για τον ρόλο των smartphones στις έξυπνες πόλεις και παρουσιάζονται ενδεικτικές mobile εφαρμογές για έξυπνες πόλεις.

- Στο κεφάλαιο 5 γίνεται η ανάλυση καθώς και η παρουσίαση της mobile εφαρμογής που έχουμε αναπτύξει .
 - Στο κεφάλαιο 6 γίνεται η ανάλυση και η παρουσίαση της Web εφαρμογής που αναπτύξαμε .
 - Στο κεφάλαιο 7 παρουσιάζουμε τα κυκλώματα (Things) IoT που έχουμε σχεδιάσει και υλοποιήσει για την μέτρηση των περιβαλλοντικών συνθηκών καθώς και την εμφάνιση των διαθέσιμων θέσεων στάθμευσης.
 - Στο κεφάλαιο 8 γίνεται μια ανασκόπηση της εργασίας ,και παρουσιάζεται η αξιολόγηση από τους χρήστες που την έχουν δοκιμάσει την εφαρμογή.
- Στο κεφάλαιο 9 περιγράφουμε τα συμπεράσματα από την υλοποίηση της εφαρμογής και προτείνουμε μελλοντικές επεκτάσεις – βελτιώσεις της.

2. Έξυπνη Πόλη

2.1 Ορισμός της έννοιας της «Έξυπνης Πόλης» (Smart City)

Διάφορες πόλεις σε όλο τον κόσμο προσεγγίζουν την έννοια των «έξυπνων πόλεων». Καθεμιά από αυτές συνήθως αντιμετωπίζει διαφορετικές προκλήσεις και δίνει προτεραιότητα σε εναλλακτικούς στόχους, όπως η βελτίωση της τοπικής καθημερινής ζωής, η ανάπτυξη κοινωνιών που βασίζονται στη γνώση, η γεφύρωση του ψηφιακού χάσματος και η προώθηση της ηλεκτρονικής διακυβέρνησης σε τοπικό επίπεδο (Anthopoulos and Vakali, 2012).

Σε γενικές γραμμές, μια έξυπνη πόλη είναι ένα έργο που βασίζεται σε μια στρατηγική από πάνω προς τα κάτω ή από κάτω προς τα πάνω που χρησιμοποιεί τεχνολογία πληροφοριών και επικοινωνιών (ΤΠΕ) για τη βελτίωση της παραγωγικότητας των σύγχρονων πόλεων. Οι έξυπνες πόλεις επικεντρώνονται στην παροχή πόρων για να βοηθήσουν τους πολίτες να συνεισφέρουν στην αστική αλλαγή και να ζουν καλύτερες, πιο παραγωγικές ζωές. Από οικονομική, πολιτική και κοινωνική άποψη, αυτές οι εξελίξεις έχουν φέρει νέες ευκαιρίες, συνεργασία και στρατηγικές επίλυσης προβλημάτων για τις πόλεις. Οι στρατηγικές έξυπνων πόλεων έχουν υιοθετηθεί από πολλούς ακαδημαϊκούς, οργανισμούς, εταιρείες και μεγάλες παγκόσμιες πόλεις. Η Ευρωπαϊκή Ένωση και άλλοι διεθνείς οργανισμοί πιστεύουν σε μια ενσύρματη μορφή που βασίζεται στις ΤΠΕ, η οποία προωθείται με μεγάλες επενδύσεις στην έρευνα και την ανάπτυξη. Η μέθοδος της Έξυπνης Πόλης κερδίζει τη δημοτικότητα λόγω: α) του αντίκτυπου που έχει σε ατομικό επίπεδο (η μεθοδολογία εστιάζει στους πολίτες, φέρνοντας έτσι μια λανθάνουσα μεταμόρφωση των πολιτών σε έξυπνους πολίτες) και β) λόγω της χρήσης της σε πολλούς άλλους κλάδους, η οποία προάγει μια πολυπολιτισμικότητα.

Οι έξυπνες πόλεις είναι αστικές περιοχές που χρησιμοποιούν δίκτυα Internet of Things (IoT) και διάφορους τύπους ηλεκτρονικών αισθητήρων για τη συλλογή δεδομένων, τα οποία επεξεργάζονται και τα χρησιμοποιούν για την αποτελεσματική διαχείριση περιουσιακών στοιχείων και πόρων. Περιλαμβάνει δεδομένα τα οποία τα συλλέγουν από τους πολίτες ή από συσκευές που επεξεργάζονται και αναλύουν δεδομένα που χρησιμοποιούνται για την παρακολούθηση και τη διαχείριση συστημάτων μεταφοράς, σταθμών παραγωγής ενέργειας, δικτύων ύδρευσης, διαχείρισης α-

πορριμμάτων, ανίχνευσης εγκλήματος, συστημάτων πληροφοριών, σχολείων, βιβλιοθηκών, νοσοκομείων και άλλων κοινοτήτων. (McLaren και Ajman, 2015).

Οι Deakin And Al Wear (2011) αναφέρουν τους ακόλουθους παράγοντες που εμπλέκονται στον καθορισμό μιας Έξυπνης Πόλης.

- Ευρεία χρήση ηλεκτρονικών και ψηφιακών τεχνολογιών σε κοινότητες και πόλεις.
- Χρήση ΤΠΕ για αλλαγή των συνθηκών διαβίωσης και εργασίας στην περιοχή.
- Ενοποίηση ΤΠΕ και κυβερνητικών συστημάτων.

Ο όρος «Έξυπνη Πόλη» είναι μάλλον σχετικά χαλαρός και μπορεί να εφαρμοστεί σε:

- υποδομές μεταφορών και ΤΠΕ
- τους ανθρώπους, ειδικά από την άποψη ενός μορφωμένου, ευέλικτου και δημιουργικού εργατικού δυναμικού.
- το περιβάλλον, από άποψη βιωσιμότητας, ασφάλειας, ενεργειακής απόδοσης και ποιότητας ζωής.
- διακυβέρνηση με την ευρύτερη έννοια, που περιλαμβάνει τη συμμετοχή, τη διαφάνεια και ένα μακροπρόθεσμο όραμα.

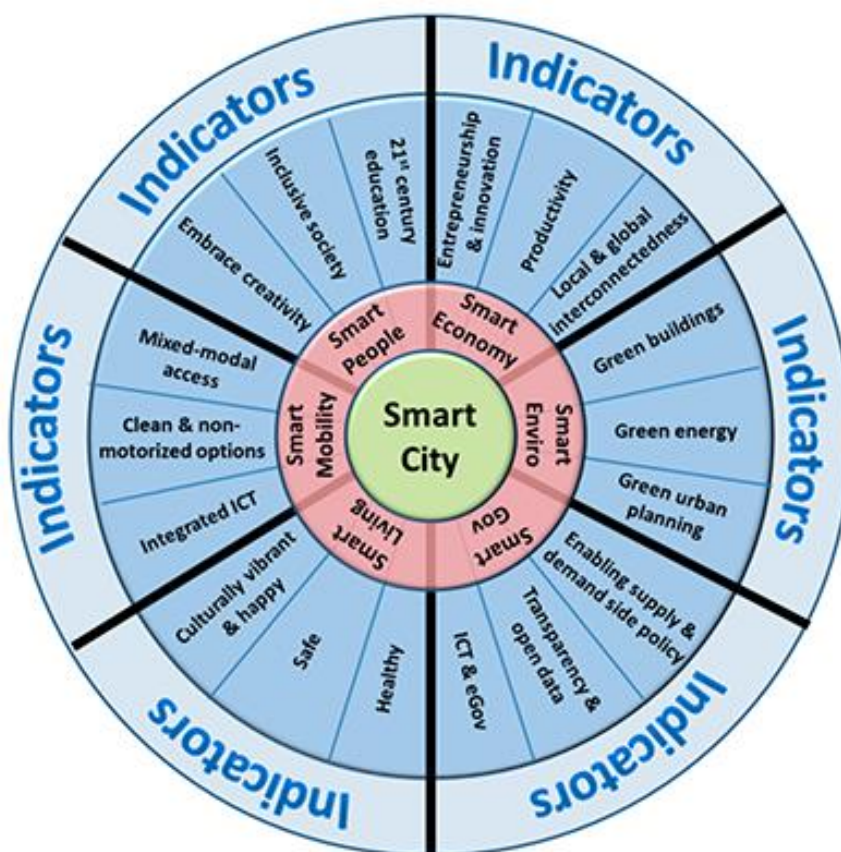
Ως εκ τούτου, οι έξυπνες πόλεις προωθούν την έννοια της αστικής αποτελεσματικότητας, η οποία εξαρτάται όχι μόνο από τις οικονομικές επενδύσεις της πόλης σε υποδομές υλικού, αλλά και από τη διαθεσιμότητα και την ποιότητα της ανταλλαγής γνώσης και της κοινωνικής υποδομής. (Caragliu and Nijkamp, 2008).

Μια κοινωνική έξυπνη πόλη είναι πλούσια σε υποδομές, η οποία συνδυάζει και αξιοποιεί τόσο την ανθρώπινη όσο και την τεχνητή νοημοσύνη, ενδυναμώνει και εμπλέκει τους ανθρώπους σε δραστηριότητες όπου οι αστικές κοινωνικές αλληλεπιδράσεις ευδοκιμούν με στόχο την βελτίωση της ποιότητας ζωής και την ανάπτυξη του πολιτισμού. (Christopoulou et al.)

2.2 Χαρακτηριστικά της «Έξυπνης Πόλης»

Το σύστημα “Έξυπνης Πόλης αποτελείται από έξι βασικά δομικά στοιχεία που είναι στενά συνδεδεμένα μεταξύ τους (Vinod Kumar T,2017):

Ο ερευνητής αστικής ανάπτυξης Boyd Cohen, εισάγει το μοντέλο του τροχού για την έξυπνη πόλη σύμφωνα και με αυτά που αναφέρθηκαν παραπάνω.

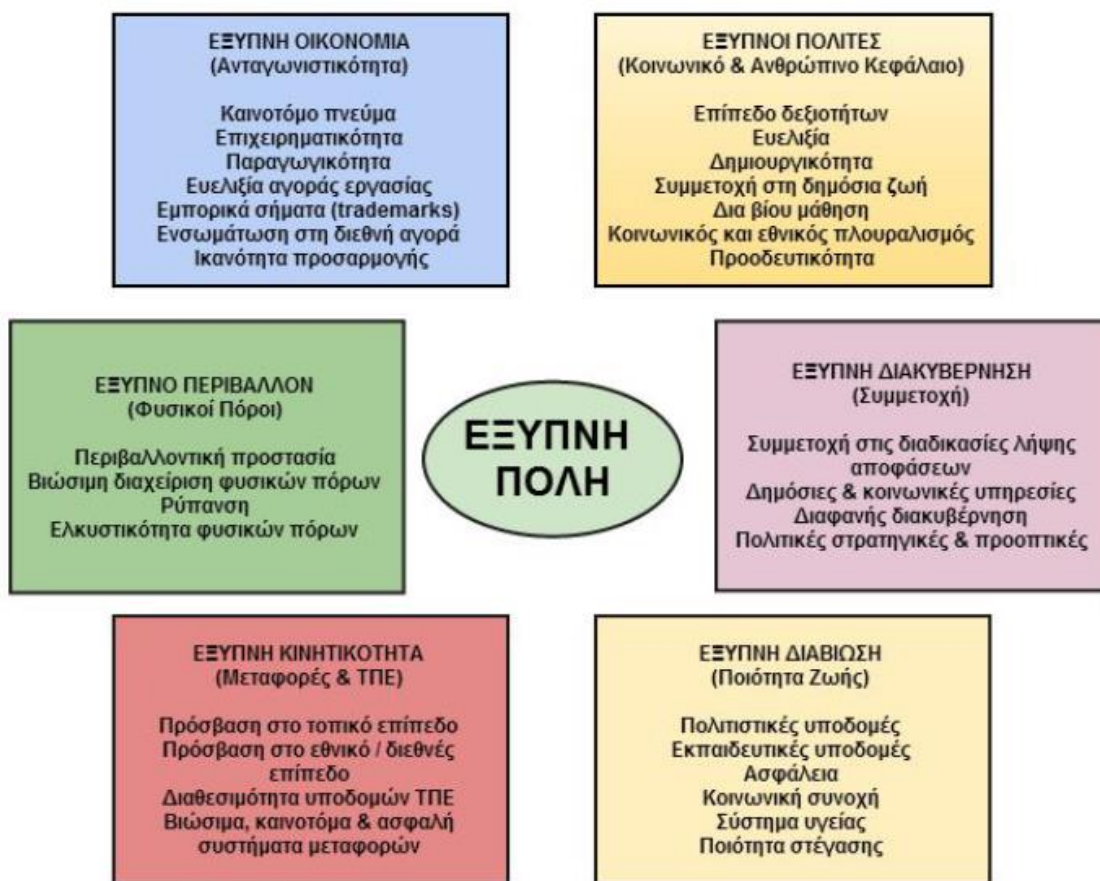


Εικόνα 2 Το μοντέλο του τροχού για την έξυπνη πόλη (Πηγή: Boyd Cochen,2012)

Διαστάσεις της “Έξυπνης Πόλης σχετιζόμενες πτυχές (τομείς) της αστικής ζωής

Έξυπνη οικονομία	Βιομηχανία
Έξυπνοι άνθρωποι	Εκπαίδευση
Έξυπνη Διακυβέρνηση	Ηλεκτρονική Δημοκρατία
Έξυπνη Κινητικότητα	logistics & υποδομές
Έξυπνο Περιβάλλον	αποδοτικότητα και βιωσιμότητα
Έξυπνη Διαβίωση	ασφάλεια και ποιότητα

Πίνακας 1 Διαστάσεις της “Έξυπνης Πόλης (Πηγή: Albino, Berardi, Dangelico, 2015)



Εικόνα 3 Διαστάσεις Έξυπνης Πόλης (Πηγή: Giffinger et al, 2007)

- **Έξυπνη οικονομία**

Με τον όρο έξυπνη οικονομία εννοούμε την οικονομία, που συμμετέχει στην αύξηση της παραγωγικότητας, τη δημιουργία νέων προϊόντων και την μείωση κόστους των υπηρεσιών.

- **Έξυπνοι άνθρωποι**

Ο όρος «έξυπνοι άνθρωποι» χαρακτηρίζει πολίτες, με αυξημένες δεξιότητες πάνω στις ΤΠΕ, οι οποίοι μπορούν να έχουν ενεργό συμμετοχή στη βελτίωση της πόλης και μέσω ειδικών εργαλείων ανάλυσής να λαμβάνουν αποφάσεις και να δημιουργούν νέα προϊόντα και υπηρεσίες.

- **Έξυπνη διακυβέρνηση**

Στην έξυπνη διακυβέρνηση είναι τμήμα της διακυβέρνηση ολόκληρης της πόλης, και σχετίζεται με διοικητικές λειτουργίες και λειτουργίες λήψης αποφάσεων. Το κύριο εργαλείο για την επίτευξη αυτού του στόχου είναι οι ΤΠΕ (υποδομές, υλικό και λογι-

σμικό), που είναι εφοδιασμένες με έξυπνες διαδικασίες , ηλεκτρονικές υπηρεσίες, και ηλεκτρονικές συναλλαγές των πολιτών .

- **Έξυπνη κινητικότητα**

Με την έξυπνη κινητικότητα εννοούμε τα πολυδιάστατα συστήματα μεταφορών και που στηρίζονται τις ΤΠΕ. Τα συστήματα αυτά μπορούν να περιλαμβάνουν τραμ, λεωφορεία, τρένα, μετρό, αυτοκίνητα, ποδήλατα και πεζούς, που χρησιμοποιούν έναν ή περισσότερους τρόπους μεταφοράς. Οι πολίτες μπορούν σε πραγματικό χρόνο να έχουν πρόσβαση τις πληροφορίες πραγματικού χρόνου, προκειμένου να εξοικονομηθεί χρόνος, χρήμα, να βελτιωθεί η αποτελεσματικότητα των μετακινήσεων και να μειωθούν οι εκπομπές CO₂. Οι πολίτες θα μπορούσαν να συμβάλουν παρέχοντας τα δικά τους δεδομένα σε πραγματικό χρόνο.

- **Έξυπνο περιβάλλον**

Το έξυπνο περιβάλλον είναι ο τομέας των έξυπνων πόλεων που με την χρήση των ΤΠΕ μπορούμε να έχουμε την δυνατότητα μέτρησης των ενεργειακών δικτύων, τον έλεγχο και την παρακολούθηση για της ρύπανσης. Ο τομέας αυτός περιλαμβάνει τις υπηρεσίες, όπως ο φωτισμός δρόμων, η διαχείριση αποβλήτων, το αποχετευτικό συστήματα και τα συστήματα ύδρευσης κ.α.

- **Έξυπνη διαβίωση**

Η έξυπνη διαβίωση αφορά τρόπο ζωής των πολιτών και την καταναλωτικής τους συμπεριφοράς και κατανάλωσης χρησιμοποιώντας ΤΠΕ . Ο τομέας αυτός σχετίζεται με την υγιή και ασφαλή ζωή σε μια πολιτιστικά ζωντανή πόλη, με υψηλή ποιότητα ζωής.

Αυτά τα έξι δομικά στοιχεία είναι στενά συνδεδεμένα μεταξύ τους και συμβάλλουν στο "Σύστημα Έξυπνη Πόλη". Ένα έργο Έξυπνης Πόλης έχει τέσσερα κύρια στρώματα που καλύπτουν τους παραπάνω τομείς:

- Αειφορία, εφαρμογές και υπηρεσίες προστιθέμενης αξίας για τους πολίτες.
- Πλατφόρμα παροχής υπηρεσιών (επεξεργασία και αξιοποίηση των Μεγάλων Δεδομένων της περιοχής).
- Αισθητήρες (IoT για τη συλλογή των μεγάλων δεδομένων της πόλης και διαχείριση της υποδομής από απόσταση).

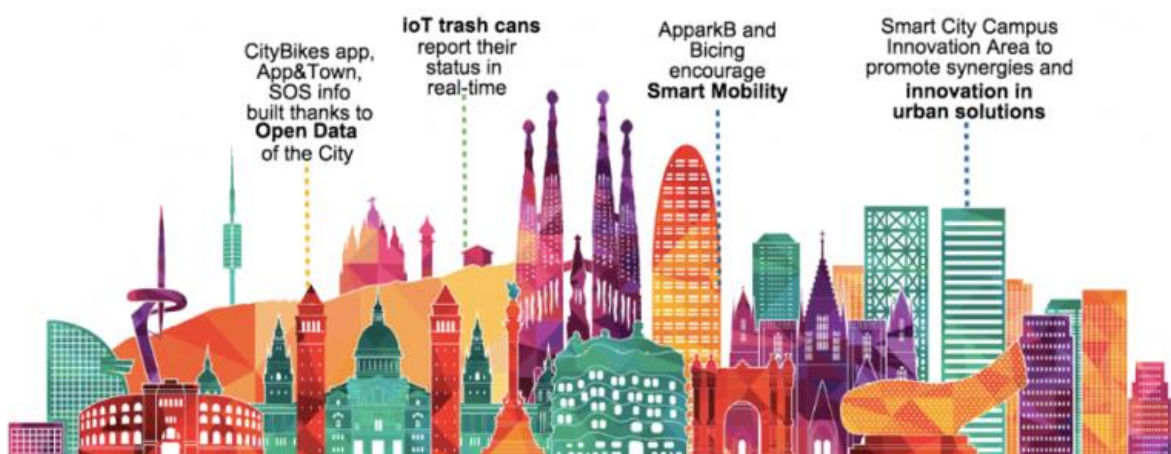
- Υποδομές (δίκτυα, διαθεσιμότητα Wi-Fi και όλος ο τεχνολογικός εξοπλισμός που επιτρέπει την κατασκευή μιας "Έξυπνης Πόλης).

2.3 Έξυπνες πόλεις στην Ευρώπη

Με βάση τα σύγχρονα πρότυπα μια έξυπνη πόλη, δεν είναι απαραίτητο να στηρίζεται μόνο την ανάπτυξη των Τ.Π.Ε αλλά σε και άλλα στοιχεία, όπως τη δυνατότητα για συμμετοχή και πρόσβαση σ' ένα ευρύτερο πλαίσιο κοινωνικής οικονομικής και πολιτιστικής ανάπτυξης. Υπάρχουν αρκετά παραδείγματα που θεωρούνται επιτυχημένα έργα Έξυπνης Πόλης σε όλο τον κόσμο:

➤ Βαρκελώνη

Η Βαρκελώνη θεωρείται μια από τις πιο πρωτοποριακές πόλεις στον κόσμο όσον αφορά την υιοθέτηση της έξυπνης τεχνολογίας και του Διαδικτύου των Πραγμάτων. Η Ευρωπαϊκή Ένωση ανακήρυξε τη Βαρκελώνη την πιο καινοτόμο πόλη της Ευρώπης το 2018. Αυτή είναι μια τυπική περίπτωση της μακροπρόθεσμης πρακτικής μιας πόλης με σημασία "Έξυπνης Πόλης. Το ισχυρό βιομηχανικό και επιχειρηματικό υπόβαθρό της έχει συμβάλει στη διαμόρφωση μιας οικονομίας της γνώσης, βασισμένης στην προώθηση της ανταγωνιστικότητας, της επιχειρηματικότητας, της τεχνολογίας και της καινοτομίας, τα οποία είναι αδιαχώριστα από την οικονομική ανάπτυξη και ανάπτυξη και τις παραγωγικές διαδικασίες (Schaffers et al., 2012).



Εικόνα 4 "Έξυπνη Πόλη Βαρκελώνη

Πηγή : <https://www.e-zigurat.com/blog/en/smart-city-barcelona-experience>

Είναι η πρώτη πόλη που ανέπτυξε μια νέα υποδομή συλλογής και επεξεργασίας δεδομένων που αποτελείται από την πλατφόρμα συλλογής ανοιχτού κώδικα της **Sentilo**, ένα δίκτυο αισθητήρων και την πλατφόρμα **CityOS**, η οποία αναλύει δεδομένα και δημιουργεί εφαρμογές υπηρεσιών σε επίπεδο διεπαφής χρήστη για ευκολότερη πρόσβαση στα δεδομένα. Τα δεδομένα που συλλέγονται ανήκουν στην πόλη και είναι διαθέσιμα σε πολίτες, ιδιωτικές εταιρείες και άλλους ενδιαφερόμενους φορείς, αλλά η πόλη και οι πολίτες της θα διατηρήσουν την πλήρη ιδιοκτησία και θα καθορίσουν τα κατάλληλα δικαιώματα πρόσβασης, διασφαλίζοντας ότι προστατεύονται το απόρρητο, η ζωή τους κλπ. Στη Βαρκελώνη, υπάρχει ένα δίκτυο αισθητήρων που καταγράφουν: - Ηλεκτρισμό - Καιρό - Εξωτερικές περιβαλλοντικές συνθήκες - Στάθμευση - Αποκομιδή σκουπιδιών - Ποιότητα αέρα - Θόρυβος - Ροή πλήθους - Οδική κυκλοφορία.

Η Βαρκελώνη έχει εφαρμόσει τα παρακάτω συστήματα:

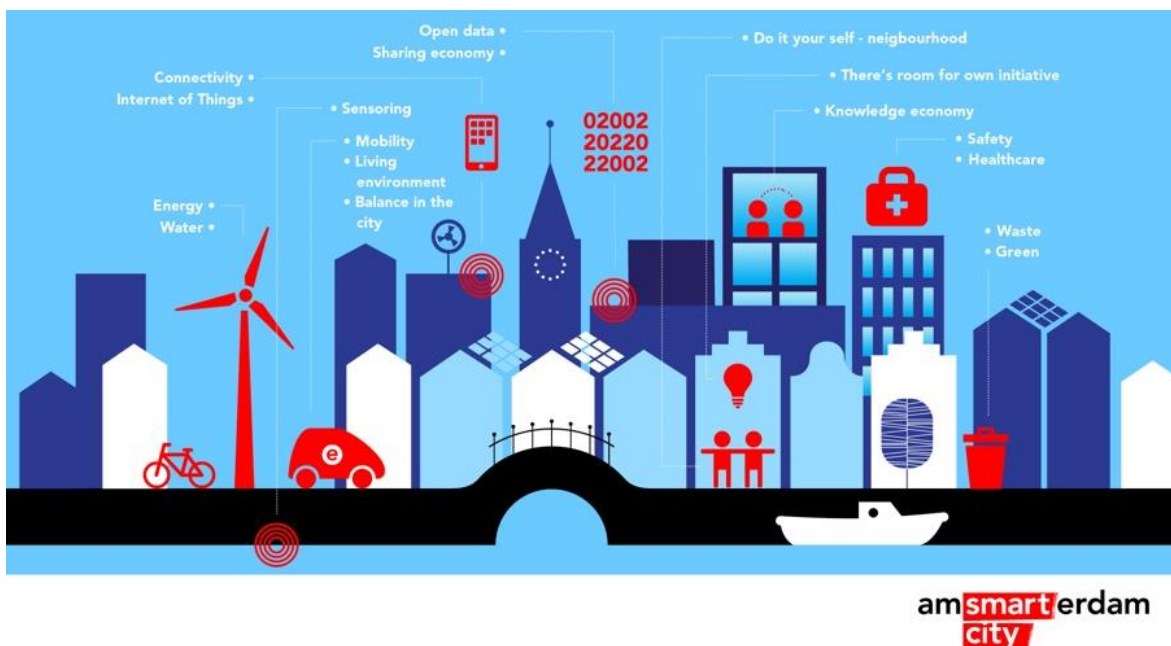
Έξυπνη στάθμευση: ένα σύστημα αισθητήρων για να παρκάρουν οι οδηγοί δωρεάν. Οι αισθητήρες βρίσκονται κάτω από την ασφαλτο και μπορούν να εντοπίσουν τις διαθέσιμες θέσεις στάθμευσης και να ειδοποιήσουν τους οδηγούς. Μέσα σε ένα χρόνο από την εφαρμογή, η πόλη εξέδιδε 4.000 άδειες στάθμευσης την ημέρα.

Έξυπνος φωτισμός: Έχουν εγκατασταθεί συστήματα φωτισμού που βασίζονται σε LED και ένα δίκτυο αισθητήρων έχει αντικαταστήσει τα φώτα του δρόμου σε όλη τη Βαρκελώνη. Αυτό το σύστημα φωτισμού είναι πιο ενεργειακά αποδοτικό και μειώνει τη θερμότητα που παράγεται από παλιούς λαμπτήρες, εξοικονομώντας χρήματα στην πόλη. Το σύστημα φωτισμού λαμβάνει πληροφορίες για το περιβάλλον (ρύπανση, υγρασία, θερμοκρασία, ανθρώπινη παρουσία και θόρυβος) με τη βοήθεια αισθητήρων. Η κεντρική μονάδα του επιτρέπει να επικοινωνεί με τα φώτα και η μονάδα διαχειρίζεται επίσης διάφορες άλλες υπηρεσίες όπως σταθμούς φόρτισης, Wi-Fi και καλώδια οπτικών ινών. Οι αισθητήρες μπορούν να προσαρμόσουν τον φωτισμό με βάση την ανθρώπινη παρουσία και καθυστέρηση.

Έξυπνες στάσεις: Μια οθόνη αφής που τροφοδοτείται από το ήλιο εμφανίζει την ώρα άφιξης και τις διαθέσιμες θέσεις επιβατών του λεωφορείου που φτάνει. Στην οθόνη ακόμη εμφανίζονται τουριστικές πληροφορίες και παρέχεται δωρεάν Wi-Fi. Τα λεωφορεία διαθέτουν GPS, αισθητήρες τοποθεσίας και στέλνουν τις σχετικές πληροφορίες στις στάσεις λεωφορείων.

➤ Amsterdam

Η στρατηγική του Άμστερνταμ ως Έξυπνη Πόλη βασίζεται στη συνεχή ανάπτυξη έργων ΤΠΕ που επιτρέπουν την εισαγωγή νέων εφαρμογών, υπηρεσιών, συσκευών και τεχνολογικών υποδομών εντός της πόλης βραχυπρόθεσμα και μεσοπρόθεσμα. Το **Ίδρυμα Έξυπνης Πόλης του Άμστερνταμ** ιδρύθηκε για να διασφαλίσει τον σωστό συντονισμό και την εφαρμογή. Με αυτόν τον τρόπο, η στρατηγική έχει «μετατοπιστεί από μια ολιστική άποψη [της φάσης σχεδιασμού] σε συγκεκριμένα έργα» (Baron 2012b).



Εικόνα 5 Άμστερνταμ "Έξυπνη Πόλη

Πηγή: <https://smartercommunities.media/truly-smart-city-five-lessons-amsterdam/>

Ένα δίκτυο αισθητήρων καταγράφει την ενέργεια που καταναλώνουν τα καταστήματα της πόλης, καθώς και το διοξείδιο του άνθρακα και τη ρύπανση του περιβάλλοντος. Για τον έλεγχο της ατμοσφαιρικής ρύπανσης έχει εγκατασταθεί το «**TreeWiFi**» ως αισθητήρας και έχει τοποθετηθεί στα δέντρα. Όταν μειώνονται οι μολύνσεις, οι άνθρωποι ανταμείβονται με μια δωρεάν σύνδεση στο διαδίκτυο. Αυτό προϋποθέτει τη συμβολή των πολιτών που θα πρέπει να αποφεύγουν να χρησιμοποιούν αυτοκίνητα και να μετακινούνται με ποδήλατο ή μέσα μαζικής μεταφοράς. Έχουν επίσης εγκατασταθεί «έξυπνοι» ελεγκτές για τον έλεγχο του δημοτικού φωτισμού σβήνοντας ή μειώνοντας τα φώτα όταν δεν χρησιμοποιούνται, καθώς και με-

τηρήτες που μετρούν την κατανάλωση ενέργειας διαφόρων συσκευών, οι οποίοι θα αντικατασταθούν με λιγότερο ενεργοβόρες, εάν χρειαστεί.

Smart mobility: Η διαχείριση της κυκλοφορίας ελέγχεται από ένα έξυπνο σύστημα μεταφορών, το οποίο παρακολουθεί τις συνθήκες κυκλοφορίας σε πραγματικό χρόνο και προτείνει εναλλακτικές διαδρομές στους πολίτες για τη μείωση της κυκλοφοριακής συμφόρησης και τη διευκόλυνση των πολιτών να ταξιδεύουν, μειώνοντας έτσι την περιβαλλοντική ρύπανση. Μέσω της πλατφόρμας **Mobypark** παρέχονται θέσεις στάθμευσης σε ιδιωτικούς χώρους, δημόσιους χώρους, ξενοδοχεία, νοσοκομεία κ.λπ. και μπορούν να ενοικιαστούν βραχυπρόθεσμα ή μακροπρόθεσμα μετά από κράτηση (Parking in Amsterdam - Mobypark, 2016). Όταν χρησιμοποιούν την Digital Road Authority, οι κάτοικοι μπορούν να λαμβάνουν συμβουλές μέσω μιας εφαρμογής για το τηλέφωνό τους, για τους προορισμούς τους και την τρέχουσα κατάσταση των φωτεινών σηματοδοτών. Ως εκ τούτου, ανάλογα με την προσέλευση των πολιτών και την πυκνότητα των φωτεινών σηματοδοτών, η Ψηφιακή Οδική Αρχή μπορεί να προσαρμόζει τον χρόνο διέλευσης των οχημάτων με αντίστοιχα φανάρια κάθε φορά.

Smart Lighting : Το Άμστερνταμ χρησιμοποιεί «ευέλικτο φωτισμό δρόμου» για τον έλεγχο του δημόσιου φωτισμού. Μέσω αυτής της πλατφόρμας, η πόλη και οι συνοικίες της μπορούν να ελέγχουν με ροοστάτες το δημόσιο φωτισμό. Με τα «έξυπνα φώτα» και τα τηλεχειριστήρια ή τους αισθητήρες, ο φωτισμός μπορεί να ρυθμιστεί ανάλογα με τις καιρικές συνθήκες, βελτιώνοντας την ασφάλεια και εξοικονομώντας ενέργεια. Η εξοικονομούμενη ενέργεια μπορεί να χρησιμοποιηθεί για άλλες λειτουργίες, όπως η τροφοδοσία ενός δικτύου Wi-Fi ή τη μέτρηση της ποιότητας του αέρα και την μόλυνση.

IoT Living Lab : Με την εγκατάσταση συσκευών beacon, οι συσκευές smartphone μπορεί να συνδεθούν με άλλες συσκευές χωρίς σύνδεση στο διαδίκτυο. Με αυτόν τον τρόπο, οι πολίτες μπορούν να μάθουν για αξιοθέατα όπως εστιατόρια, μουσεία και αξιοθέατα, και έτσι έχουν την ευκαιρία να λάβουν σημαντικές πληροφορίες για αυτά τα αξιοθέατα. Αυτό επιτρέπει επίσης τη μετάφραση τουριστικών πινακίδων σε διαφορετικές γλώσσες, ανάλογα με τη γλώσσα των τουριστών.

➤ Κοπεγχάγη

Η Κοπεγχάγη ψηφίστηκε το 2017 σαν την πιο έξυπνη πόλη του κόσμου, με βάση την παροχή έξυπνων λύσεων για τους πολίτες της. Η πόλη πρόσφερε στους πολίτες την καλύτερη διαθεσιμότητα σε 4G, πληθώρα σημείων πρόσβασης Wi-Fi, ψηφιακή πλατφόρμα παρακολούθησης αισθητήρων κυκλοφορίας.



Εικόνα 6 Κοπεγχάγη "Έξυπνη Πόλη"

Πηγή : <https://www.smartecocity.com/category/eucities/dn-copenhagen/>

Οι εφαρμογές IoT που την χαρακτηρίζουν ως Έξυπνη Πόλη είναι:

- Smart City infrastructure
- Smart Environment
- Smart Mobility
- Smart waste
- Plant sensing
- City WiFi-Tracking
- Smart Environment
- Smart Water Defense

Smart Water Defense: Με την χρήση ενός έξυπνου σύστημα υδατοπροστασίας γίνεται η ειδοποίηση των πολιτών της πόλης για να αποφύγουν τον κίνδυνο αποκλεισμού σε περίπτωση πλημμύρας.

Smart Waste: Με ένα έξυπνο σύστημα απορριμμάτων στέλνονται ειδοποιήσεις στα φορτηγά καθαρισμού για βελτιστοποίηση των διαδρομών τους χωρίς να χάνουν πολύτιμο χρόνο, καύσιμα και ενέργεια.

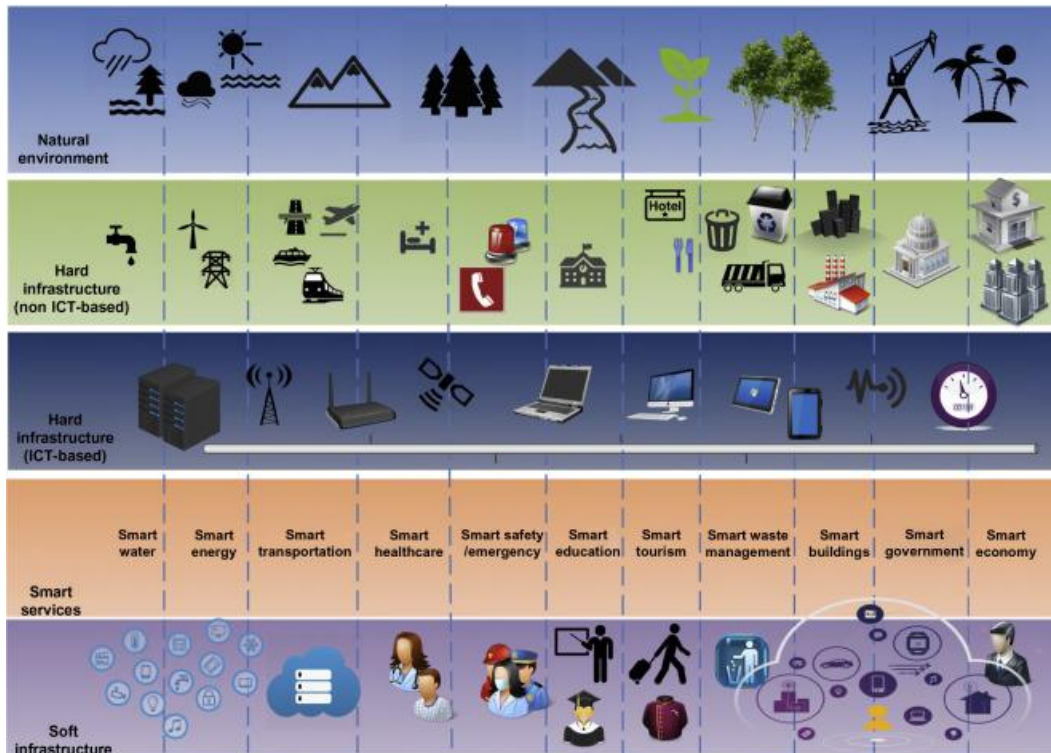
Smart Mobility: Τα έξυπνα συστήματα μεταφορών εξάγουν πληροφορίες σε πραγματικό χρόνο και προσαρμόζουν την κυκλοφορία, τα σήματα και τα σήματα με βάση τις τρέχουσες ανάγκες κυκλοφορίας. Επιπλέον, μέσω του συστήματος έξυπνης στάθμευσης, είναι δυνατή η κράτηση θέσης στάθμευσης. Περισσότερες από 100 βάσεις δεδομένων, όπως χάρτες, χώροι στάθμευσης και προσομοιώσεις ροής κυκλοφορίας, είναι διαθέσιμες μέσω του Open Data Portal. Επίσης, τα περισσότερα στοιχεία στο μέλλον θα είναι στατιστικά στοιχεία που θα δημοσιεύουν στοιχεία αστικής ενέργειας και πληθυσμού. Μια άλλη δυναμική καινοτομία είναι η πλατφόρμα Copenhagen Connect (City Data Exchange) για κοινή χρήση ιδιωτικών και δημόσιων δεδομένων, σχεδιασμένη να καλύπτει τις ανάγκες τους προτείνοντας ιδέες και λύσεις, χρησιμοποιώντας δεδομένα για την επίλυση προβλημάτων και διαμορφώνοντας νέους τρόπους δημόσιας συμμετοχής. Ωστόσο, η χρήση αυτών των συστημάτων θα παραβιάζει πάντα το απόρρητο και θα εκμεταλλεύεται προσωπικά δεδομένα.

City WiFi-Tracking: Η Κοπεγχάγη διαθέτει ένα ολοκληρωμένο δίκτυο Free Wi-Fi, σε ολόκληρη την πόλη με σκοπό την σύνδεση των πολιτών στο Internet.

2.4 Έξυπνες Πόλεις στην Ελλάδα

➤ Τρίκαλα

Τα Τρίκαλα είναι ίσως το πιο χαρακτηριστική περίπτωση Έξυπνης Πόλης στην Ελλάδα και η πρώτη ελληνική πόλη που έχει αξιοποιήσει με επιτυχία IoT τεχνολογίες.



Εικόνα 7 Τρίκαλα "Εξυπνη Πόλη"

Πηγή: Leonidas Anthopoulos, - The Smart city of Trikala 2019

Στην πόλη έχουν εφαρμοστεί τα παρακάτω συστήματα:

Σύστημα Έξυπνης Διαχείρισης Στάθμευσης: Με το Σύστημα Έξυπνης Διαχείρισης Στάθμευσης, επιτυγχάνεται η εύρεση, η απεικόνιση και ο έλεγχος οριοθετημένων θέσεων στάθμευσης στο κέντρο της πόλης. Έχει γίνει εγκατάσταση δικτύου εξειδικευμένων αισθητήρων στο οδόστρωμα συγκεκριμένων οδών της πόλης, έτσι ώστε να αντιστοιχεί ένας αισθητήρας για κάθε διακριτή, διαγραμμισμένη θέση στάθμευσης. Ο αισθητήρας τροφοδοτεί τα σημεία ελέγχου του δικτύου (controllers) στέλνοντας τα ανάλογα σήματα, όταν η θέση είναι ή δεν είναι κατειλημμένη. Επιπλέον οι πολίτες ενημερώνονται σε πραγματικό χρόνο για τη διαθεσιμότητα θέσεων στην επιλεγμένη περιοχή, τόσο μέσω της εφαρμογής στάθμευσης (mobile app) για κινητά τηλέφωνα, όσο και από πινακίδες που μπορούν να εγκατασταθούν σε κομβικά σημεία της πόλης. Επίσης παρέχεται και στα όργανα ελέγχου της στάθμευσης, ενημέρωση σε πραγματικό χρόνο για περιπτώσεις παράνομου παρκαρίσματος. Μέσω της εφαρμογής παρέχεται και δυνατότητα αυτόματης πληρωμής του τιμήματος στάθμευσης. (<https://trikalacity.gr/systima-exypnis-stathmeysis/>)

Σύστημα παρακολούθησης περιβαλλοντικών συνθηκών: Με τη χρήση ειδικών συσκευών περιβαλλοντικών μετρήσεων (όπως για συγκέντρωση αέριων ρύπων, αιωρούμενων σωματιδίων και θορύβου), μπορεί να εκτιμηθεί η ποιότητα της ατμόσφαιρας και να αξιολογηθεί πιθανός αντίκτυπος στη δημόσια υγεία. Επίσης, απεικονίζονται σε πραγματικό χρόνο τυποποιημένοι δείκτες ποιότητας του περιβάλλοντος που επιτρέπουν συγκριτική αξιολόγηση (benchmarking), επισημάνσεις (alerts) και την αναγνώριση τάσεων που θα μπορούσαν να οδηγήσουν στη λήψη μέτρων. (<https://trikalacity.gr/systima-parakoloythisis-perivallontikon-synthikon/>)

Έξυπνη και Διασυνδεδεμένη Ψηφιακή Πλατφόρμα: Έχει εγκατασταθεί η πλατφόρμα Έξυπνης Πόλης Cisco Smart+Connected Digital Platform – CDP. Είναι ένα ολοκληρωμένο πληροφοριακό σύστημα που αξιοποιεί το Διαδίκτυο των Πραγμάτων (IoT) και διαχειρίζεται εφαρμογές προσωπικής παρακολούθησης και πληροφοριών, ενώ τροφοδοτεί συστήματα τρίτων μέσω πρωτοκόλλων ανοιχτής διασύνδεσης (API). Η πλατφόρμα συλλέγει, αποθηκεύει, κανονικοποιεί και οπτικοποιεί τα δεδομένα που δημιουργούνται από τις προαναφερθείσες υποδομές και εφαρμογές και τα καθιστά διαθέσιμα για ανάλυση από όσους ενδιαφέρονται να τα χρησιμοποιήσουν προς όφελος των πολιτών και των επιχειρήσεων.

Σύστημα Έξυπνου Φωτισμού: Έχει εφαρμοστεί ένα έξυπνο σύστημα φωτισμού, μέσω του οποίου γίνεται η διαχείριση του δημοτικού ηλεκτροφωτισμού, εξοικονομώντας περισσότερο από 60% ενέργεια σε σύγκριση με τα παραδοσιακά συστήματα φωτισμού. Πιο συγκεκριμένα, τα υπάρχοντα συστήματα φωτισμού συμβατικής τεχνολογίας αντικαθίστανται από τα νέας τεχνολογίας συστήματα φωτισμού LED, που είναι ο τρόπος που αναπαρίσταται το υπεραστικό οδικό δίκτυο. Επιπλέον, έχει εγκατασταθεί ένα ασύρματο σύστημα διαχείρισης, το οποίο προσφέρει τη δυνατότητα έγκαιρης ανίχνευσης βλαβών, σχεδιασμού «έξυπνων» παρεμβάσεων, δυναμικής προσαρμογής του φωτισμού ανάλογα με τις ανάγκες για μεγιστοποίηση της εξοικονόμησης ενέργειας και βελτίωση της ορατότητας για οδηγούς, ποδηλάτες και πεζούς.

Κέντρο διαχείρισης της «Έξυπνης Πόλης»: Δημιουργήθηκε ένα κέντρο ελέγχου για όλες τις υπηρεσίες που παρέχονται. Συγκεκριμένα, εγκαθίστανται οθόνες παρακολούθησης για τα ακόλουθα συστήματα:

Σύστημα γεωγραφικών πληροφοριών που εμφανίζει τα χωροχωρικά δεδομένα και τα σημεία ενδιαφέροντος της πόλης.

Σύστημα παρακολούθησης φωτεινών σηματοδοτών. Παρέχει διαδικτυακή παρακολούθηση βλαβών σε διασταυρώσεις αστικής κυκλοφορίας που ελέγχονται από φανάρια.

Σύστημα καταγραφής της κυκλοφορίας των οχημάτων του δήμου.

Σύστημα παρακολούθησης της λειτουργία των ασύρματων κόμβων δικτύου παροχής πρόσβασης στο Διαδίκτυο.

Ηλεκτρομαγνητικό σύστημα παρακολούθησης και ρύθμισης δικτύου σωληνώσεων ύδρευσης.

Καταγραφή και παρακολούθηση της προόδου επίλυση αιτημάτων των πολιτών.

➤ **Ιωάννινα**

Οι στρατηγικές του Δήμου Ιωαννιτών για μια έξυπνη πόλη αποτελούν μια σύνθετη διαδικασία με επιθυμητές πρωτοβουλίες στο πλαίσιο:

- της έξυπνης διακυβέρνησης,
- των έξυπνων πολιτών,
- της έξυπνης διαβίωσης,
- της έξυπνης κινητικότητας,
- της έξυπνης οικονομίας και
- του έξυπνου περιβάλλοντος.



Εικόνα 8 Ιωάννινα «Έξυπνη πόλη»

Πηγή: <http://www.diavouleusi.eu/diabloleyseis/ioannnina-smart-city>

Γίνονται προσπάθειες για την ανάπτυξη των ακόλουθων υποδομών/εφαρμογών (www.ioannina.gr):

- Μητροπολιτικό Δίκτυο οπτικών ινών και ασύρματο δίκτυο Wi-Fi.
- Ηλεκτρονικές Υπηρεσίες Ιθαγένειας: Ηλεκτρονική υποβολή αιτήσεων, παραπόνων και λοιπών αιτημάτων πληροφοριών για την ανάπτυξή του.
- Εφαρμογή (Οδηγός) για την υποστήριξη των αναγκών των επισκεπτών της πόλης.
- Σύστημα μέτρησης ηλεκτρομαγνητικής ακτινοβολίας «ΕΡΜΗΣ»: Μέτρηση/Έλεγχος Εκπομπών.
- Ιστορικά Αρχεία Δημοτικών Αποφάσεων: Ψηφιοποίηση/Έκδοση Αποφάσεων την Περίοδο 1911-1980.
- Ψηφιακή Βιβλιοθήκη: Ψηφιοποίηση/διανομή υλικού Ζωσιμαίας Δημόσιας Κεντρικής Ιστορικής Βιβλιοθήκης.
- Καλλιτεχνική Δημιουργία: Καταγραφή, Ψηφιοποίηση/δημοσίευση δημιουργιών ντόπιων καλλιτεχνών.
- Πολιτισμός και Φυσικοί Πόροι: Ψηφιοποίηση συλλογών δημοτικών πινακοθηκών, εικονικές ξεναγήσεις σε δημοτικά μουσεία και ψηφιοποίηση του δρομολογίου περιήγησης στο Σπήλαιο Περάματος.

3. Το Διαδίκτυο των Πραγμάτων (Internet of Things)

3.1. Η ανάπτυξη του Διαδικτύου (Internet)

Ο ψηφιακός κόσμος υπάρχει εδώ και πολλά χρόνια, ειδικά από τους πρώτους υπολογιστές τη δεκαετία του 1950. Οι μηχανές, το λογισμικό, οι επικοινωνίες, τα δίκτυα υπάρχουν από τότε, αλλά μόνο με περιορισμένο κοινό, και στα χρόνια που ακολούθησαν, με την εμφάνιση των προσωπικών υπολογιστών, οι περισσότερες λειτουργίες τους αφορούσαν συγκεκριμένες εφαρμογές και συγκεκριμένες πληροφορίες. Η μεγαλύτερη διαφορά και αυτό που βοήθησε στη διάδοση της είδησης ήταν το Διαδίκτυο. Το Διαδίκτυο φαίνεται να είναι ίσως η μεγαλύτερη τεχνολογική επανάσταση στην ιστορία της ανθρωπότητας. Η γενική κυριαρχία του Διαδικτύου δεν ήταν βέβαια εύκολη ή άμεση. Η επίτευξη της σημερινής ταχύτητας και ευκολίας σύνδεσης στο Διαδίκτυο απαιτούσε έρευνα, νέες εφευρέσεις, βελτιώσεις στις υπάρχουσες τεχνολογίες και γνώσεις από πολλούς ανθρώπους και επιστήμονες σε πολλούς τομείς για να γίνουν εφικτά τα σημερινά αποτελέσματα. Το Διαδίκτυο μπορεί απλώς να περιγραφεί ως μια παγκόσμια σύνδεση μεταξύ υπολογιστών και δικτύων υπολογιστών το οποίο επιτρέπει την ανταλλαγή και τη μετάδοση μεγάλων ποσοτήτων δεδομένων και πληροφοριών. (Ryan, 2010). Το δίκτυο ARPANET γεννήθηκε το 1969 με πόρους από το Advanced Research Projects Agency (ARPA) (τώρα γνωστό ως DARPA). Στην αρχική του μορφή, το πρόγραμμα σχεδιάστηκε για να πειραματίζεται με την εναλλαγή πακέτων, σύμφωνα με την οποία τα δεδομένα που θα μεταδοθούν χωρίζονται σε πακέτα και πολλοί χρήστες μπορούν να μοιράζονται την ίδια γραμμή επικοινωνίας. Δεν είχε τη δομή και τη λειτουργία που έχει σήμερα και δεν μπορούσε να χρησιμοποιηθεί από απλούς ανθρώπους, αλλά η αρχή είναι η ίδια, ακόμα κι αν δεν είχε πολλές δυνατότητες γραφικών και μεταφοράς βίντεο. Μέχρι τη δημιουργία του Παγκόσμιου Ιστού, το Διαδίκτυο εμφανίστηκε με τη μορφή καταναλωτικού προϊόντος και χρησιμοποιείται με τη μορφή που το γνωρίζουμε σήμερα. Το επόμενο βήμα ήταν η ανάπτυξη του Πρωτοκόλλου Ελέγχου Μετάδοσης TCP/IP, ενός μοντέλου επικοινωνίας που ορίζει ένα πρότυπο για τον τρόπο μετάδοσης δεδομένων σε πολλαπλά δίκτυα. Το 1985, το Εθνικό Ίδρυμα Επιστημών (NSF) δημιούργησε το δικό του γρήγορο δίκτυο, το NSFNET, χρησιμοποιώντας αυτό το πρωτόκολλο TCP/IP για να συνδέ-

σει πέντε κέντρα υπερυπολογιστών μεταξύ τους αλλά και με την υπόλοιπη επιστημονική κοινότητα. Στα τέλη της δεκαετίας του 1980, όλο και περισσότερες χώρες συνδέονται με το NSFNET. Χιλιάδες πανεπιστήμια και οργανισμοί χτίζουν τα δικά τους ανεξάρτητα δίκτυα και τα συνδέουν με αυτό το νέο παγκόσμιο δίκτυο που ονομάζεται INTERNET, το οποίο εξαπλώνεται σε όλο τον κόσμο με ιλιγγιώδη ταχύτητα. Το 1990, το ARPANET καταργήθηκε. (Ryan, 2010) και το 1993 τα εργαστήρια του CERN στην Ελβετία κυκλοφόρησαν τον Παγκόσμιο Ιστό (WWW).

3.2. Εισαγωγή στο Διαδίκτυο των πραγμάτων (IoT)

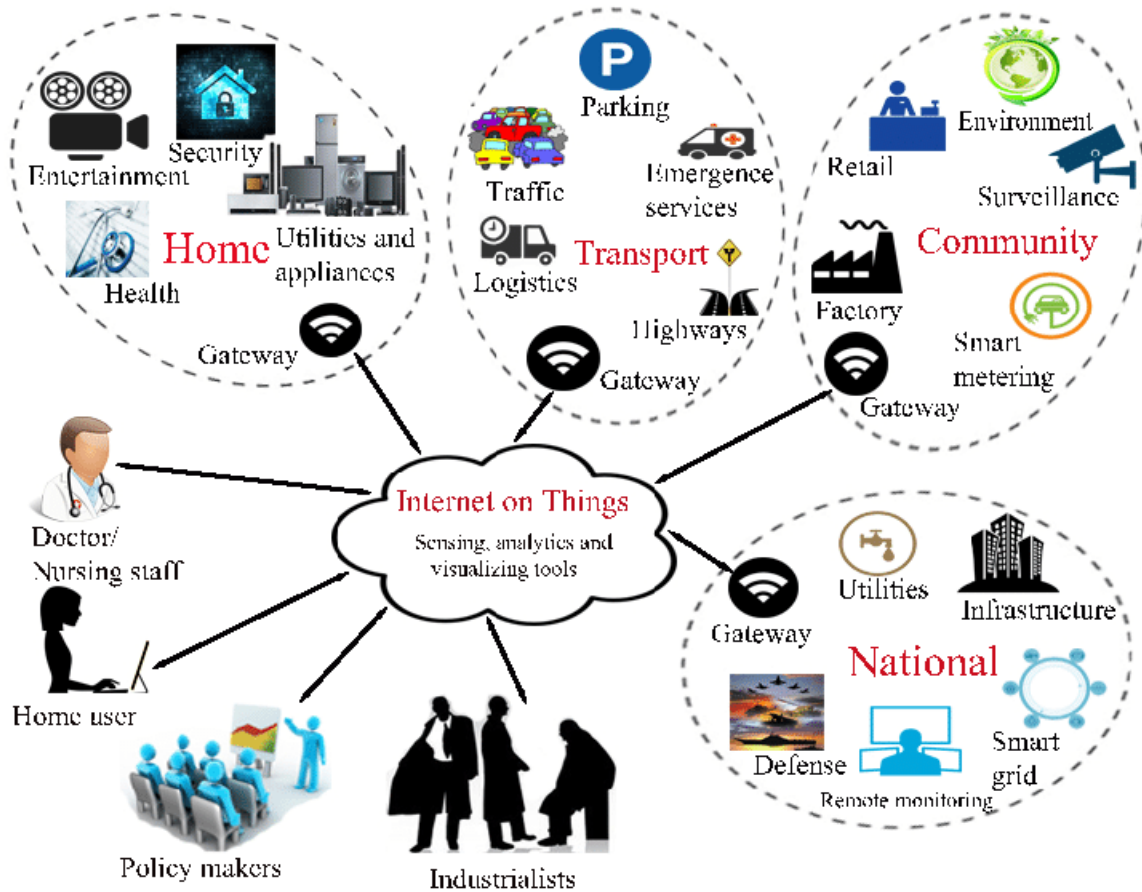
Με την ταχεία διείσδυση του διαδικτύου στην ζωή του σύγχρονου, ήταν φυσικό επακόλουθο να συνδέονται σε αυτό όλο και περισσότερες ψηφιακές συσκευές. Πέραν λοιπόν των ηλ υπολογιστών σήμερα υπάρχουν και άλλες συσκευές όπως οι λεγόμενες IoT που με την βοήθεια του κατάλληλου λογισμικού, εκτελούν υπολογιστικές εργασίες.

Το Διαδίκτυο των Πραγμάτων (IoT) είναι ένα δίκτυο επικοινωνίας από διάφορες συσκευές, οικιακές συσκευές, αυτοκίνητα και οποιαδήποτε αντικείμενα που ενσωματώνουν ηλεκτρονικά μέσα, λογισμικό, αισθητήρες και συνδεσιμότητα δικτύου για να επιτρέπουν τη σύνδεση και την ανταλλαγή δεδομένων. Με απλά λόγια, η ιδέα του IoT είναι να συνδέσει όλες τις ηλεκτρονικές συσκευές μεταξύ τους (Local Area Network) ή να μπορεί να συνδεθεί στο Internet (World Wide Web).

Η λέξη “Things” (πράγματα) δεν συνδέεται αυστηρά με ορισμένα προϊόντα. Αναφέρεται σε μια ποικιλία συσκευών που είναι εντελώς διαφορετικές μεταξύ τους, όπως αυτοκίνητα με ενσωματωμένους αισθητήρες, κάμερες, κλιματισμό, φώτα, συστήματα ασφαλείας, έξυπνα ρολόγια, ακόμη και αυτοκίνητα με πολύπλοκους αισθητήρες για την ανίχνευση αντικειμένων στο πέρασμά τους. Αποτελούν μέρος πολλών τεχνολογικών προϊόντων.

Βασικό χαρακτηριστικό όλων αυτών είναι η μεταξύ τους σύνδεση, με απώτερο στόχο να μπορεί ο χρήστης να τα ελέγχει από υπολογιστή ή φορητή συσκευή. Το Διαδίκτυο των Πραγμάτων (Internet of Things) είναι μία από τις τρεις κορυφαίες τεχνολογικές εξελίξεις της επόμενης δεκαετίας (μαζί με το mobileInternet και την αυτοματοποίηση του knowledgework) και είναι το επόμενο μεγάλο βήμα στην τεχνολογία. Ο όρος Internet of Things επινοήθηκε από τον Kevin Ashton τη δεκαετία του 1990.

Ο Ashton, ένας από τους ιδρυτές του Κέντρου Auto-ID του MIT, ήταν μέλος μιας ομάδας που ανακάλυψε πώς να συνδέει αντικείμενα στο Διαδίκτυο μέσω ετικετών RFID.



Εικόνα 9 Internet of Things

Πηγή: https://www.researchgate.net/figure/An-architecture-for-Internet-of-Things-applications-Source-4_fig1_317555651/download

Αυτή είναι και η βασική ιδέα του Διαδικτύου των πραγμάτων, η σύνδεση δηλαδή οποιασδήποτε συσκευής με το Internet και η ικανότητα ελέγχου αυτής της συσκευής απομακρυσμένα. Αυτό περιλαμβάνει έξυπνες συσκευές, όπως κινητά τηλέφωνα, ακουστικά, λάμπες, φορητές συσκευές, μέχρι εξειδικευμένα ιατρικά μηχανήματα διατήρησης ζωής και σχεδόν οτιδήποτε άλλο μπορεί να σκεφτεί ένας χρήστης. Η εταιρεία αναλυτών Gartner αναφέρει ότι μέχρι το 2020 θα υπάρχουν πάνω από 26 δισεκατομμύρια συνδεδεμένες συσκευές. Αυτό λοιπόν έχει σαν αποτέλεσμα να θεωρούμε το IoT είναι ένα γιγαντιαίο δίκτυο συνδεδεμένων «πραγμάτων».

Κατά βάση χρησιμοποιώντας τον όρο IoT, εννοούμε βασικά καθημερινές συσκευές, αισθητήρες και πράγματα, που όταν δεν είναι συνδεδεμένα στο Διαδίκτυο (Internet) δεν έχουν καμία υπολογιστική ισχύ, αλλά οι παραπάνω δυνατότητες μπορούν να αποκτηθούν μέσω της τεχνολογίας IoT. Έτσι, αυτά τα καθημερινά αντικείμενα-συσκευές μπορούν αρχικά να δημιουργήσουν πληροφορίες και στην συνέχεια να μοιραστούν αυτές τις πληροφορίες μεταξύ τους με ελάχιστη ή καθόλου ανθρώπινη παρέμβαση.

Δίκτυα πολλών διασυνδεδεμένων αντικειμένων όπως μηχανές στην βιομηχανία, αυτοκίνητα, ηλεκτρικές οικιακές συσκευές απαρτίζουν τα συστήματα IoT. Κάποιες συσκευές θα μπορούν να έχουν διάφορους ενσωματωμένους αισθητήρες (Sensors), με τους οποίους μπορούν να συλλέγουν πληροφορίες από το περιβάλλον, οι από τους χρήστες και στην συνέχεια να τις διαμοιράζονται μεταξύ τους.

Τα κύρια μέρη ενός συστήματος είναι:

- τα «πράγματα» (αντικείμενα),
- τα δίκτυα που τα συνδέουν και επικοινωνούν μεταξύ τους,
- τα συστήματα υπολογιστών καθώς και τις εφαρμογές που επεξεργάζονται τα δεδομένα που ανταλλάσσουν μεταξύ τους τα αντικείμενα.

Μερικά από τα χαρακτηριστικά του IoT είναι:

"Μέγεθος": Ο αριθμός των συνδεδεμένων συσκευών έχει αυξηθεί, αλλά το μέγεθός τους έχει περιοριστεί κάτω από το ορατό εύρος του ανθρώπινου ματιού.

"Mobility": Πληθαίνουν συνεχώς τα αντικείμενα που συνδέονται ασύρματα έτσι ώστε να είναι κινητά και ανιχνεύσιμα.

"Ετερογένεια" και "Πολυπλοκότητα": Ο αριθμός των εφαρμογών που θα εκτελούν διασυνδεδεμένα αντικείμενα δημιουργεί πρόσθετες απαιτήσεις για τη διασφάλιση της λειτουργικότητας.

3.3. Πλεονεκτήματα του IoT

Το Διαδίκτυο των πραγμάτων διευκολύνει πολλά πλεονεκτήματα στην καθημερινή μας ζωή. Μερικά από τα πλεονεκτήματά του δίνονται παρακάτω :

- **Ελαχιστοποιεί την ανθρώπινη προσπάθεια:** Καθώς οι συσκευές IoT αλληλοεπιδρούν και επικοινωνούν μεταξύ τους, μπορούν να αυτοματοποιήσουν

τις εργασίες συμβάλλοντας στη βελτίωση της ποιότητας των υπηρεσιών μιας επιχείρησης και μειώνοντας την ανάγκη για ανθρώπινη παρέμβαση.

- **Εξοικονομεί χρόνο:** Μειώνοντας την ανθρώπινη προσπάθεια, εξοικονομείτε πολύ χρόνο. Η εξοικονόμηση χρόνου είναι ένα από τα κύρια πλεονεκτήματα της χρήσης της πλατφόρμας IoT.
- **Βελτιώνει την συλλογή δεδομένων:** Οι πληροφορίες είναι εύκολα προσβάσιμες, ακόμα κι αν βρισκόμαστε μακριά από την πραγματική μας τοποθεσία, και ενημερώνονται συχνά σε πραγματικό χρόνο. Ως εκ τούτου, αυτές οι συσκευές μπορούν να έχουν πρόσβαση σε πληροφορίες από οπουδήποτε και ανά πάσα στιγμή σε οποιαδήποτε συσκευή.
- **Βελτιώνει την ασφάλεια:** Εάν έχουμε ένα διασυνδεδεμένο σύστημα, μπορεί να βοηθήσει στον πιο έξυπνο έλεγχο των σπιτιών και των πόλεων μέσω κινητών τηλεφώνων. Ενισχύει την ασφάλεια και προσφέρει προσωπική προστασία.
- **Γίνεται αποτελεσματική χρήση πόρων:** Μπορούμε να αυξήσουμε τη χρήση πόρων και να παρακολουθούμε τους φυσικούς πόρους γνωρίζοντας τη λειτουργικότητα και τον τρόπο λειτουργίας κάθε συσκευής.
- **Μειώνει την χρήση άλλου ηλεκτρονικού εξοπλισμού:** Οι ηλεκτρικές συσκευές συνδέονται απευθείας και μπορούν να επικοινωνούν με έναν ελεγκτή υπολογιστή, όπως ένα κινητό τηλέφωνο, με αποτέλεσμα την αποδοτική χρήση ηλεκτρικής ενέργειας. Ως εκ τούτου, δεν θα υπάρχει περιττή χρήση ηλεκτρικού εξοπλισμού.
- **Είναι χρήσιμο σε συστήματα κυκλοφορίας:** Η παρακολούθηση περιουσιακών στοιχείων, η παράδοση, η επιτήρηση, η παρακολούθηση της κυκλοφορίας ή της μεταφοράς, ο έλεγχος αποθέματος, η παρακολούθηση μεμονωμένων παραγγελιών και η διαχείριση πελατών μπορεί να είναι πιο οικονομικά αποδοτικές με το σωστό σύστημα παρακολούθησης που χρησιμοποιεί τεχνολογία IoT.
- **Είναι χρήσιμο για θέματα ασφάλειας:** Είναι χρήσιμο για την ασφάλεια γιατί αντιλαμβάνεται κάθε πιθανό κίνδυνο και προειδοποιεί τους χρήστες. Για παράδειγμα, το GM OnStar είναι μια ενσωματωμένη συσκευή που προσδιορίζει ένα τροχαίο ατύχημα στο δρόμο και κάνει αμέσως μια κλήση εάν εντοπιστεί ατύχημα.

- **Είναι χρήσιμο στον κλάδο της υγειονομικής περίθαλψης:** Η φροντίδα των ασθενών μπορεί να πραγματοποιηθεί πιο αποτελεσματικά σε πραγματικό χρόνο χωρίς να χρειάζεται επίσκεψη σε γιατρό. Τους δίνει τη δυνατότητα να κάνουν επιλογές καθώς και να παρέχουν φροντίδα βάσει στοιχείων.

3.4. Μειονεκτήματα του IoT

Καθώς το Διαδίκτυο των πραγμάτων προσφέρει πληθώρα θετικών υπηρεσιών εντούτοις, δημιουργεί ένα σημαντικό σύνολο μειονεκτημάτων. Μερικά από τα μειονεκτήματα του IoT δίνονται παρακάτω:

- **Υπάρχουν θέματα ασφάλειας:** Τα συστήματα IoT είναι διασυνδεδεμένα και επικοινωνούν μέσω δικτύων. Έτσι, το σύστημα προσφέρει ελάχιστο έλεγχο παρά τα τυχόν μέτρα ασφαλείας και μπορεί να οδηγήσει σε διάφορα είδη επιθέσεων δικτύου.
- **Κινδυνεύει το απόρρητο:** Το σύστημα IoT παρέχει κρίσιμα προσωπικά δεδομένα με πλήρη λεπτομέρεια χωρίς την ενεργή συμμετοχή του χρήστη.
Αυξάνει την ανεργία: Οι ανειδίκευτοι εργαζόμενοι ή ακόμα και οι ειδικευμένοι διατρέχουν υψηλό κίνδυνο να χάσουν τη δουλειά τους, οδηγώντας σε υψηλά ποσοστά ανεργίας. Έξυπνες κάμερες παρακολούθησης, ρομπότ, έξυπνα συστήματα σιδερώματος, έξυπνα πλυντήρια ρούχων και άλλες εγκαταστάσεις αντικαθιστούν τους ανθρώπους που νωρίτερα θα έκαναν αυτές τις εργασίες.
- **Πολυπλοκότητα του συστήματος:** Ο σχεδιασμός, η ανάπτυξη, η συντήρηση και η ενεργοποίηση της εκτεταμένης τεχνολογίας στο σύστημα IoT είναι αρκετά περίπλοκη.
- **Υψηλές πιθανότητες να καταστραφεί ολόκληρο το σύστημα:** Εάν υπάρχει σφάλμα στο σύστημα, είναι πιθανό κάθε συνδεδεμένη συσκευή να καταστραφεί.
- **Έλλειψη διεθνών τυποποιήσεων:** Καθώς δεν υπάρχει διεθνές πρότυπο συμβατότητας για το IoT, είναι προβληματικό για την επικοινωνία συσκευών διαφορετικών κατασκευαστών.
- **Υψηλή εξάρτηση από το διαδίκτυο:** Βασίζονται σε μεγάλο βαθμό στο διαδίκτυο και δεν μπορούν να λειτουργήσουν αποτελεσματικά χωρίς αυτό.

- **Μειωμένη πνευματική και σωματική δραστηριότητα:** Η υπερβολική χρήση του διαδικτύου και της τεχνολογίας κάνει τους ανθρώπους να αγνοούν επειδή βασίζονται σε έξυπνες συσκευές αντί να κάνουν σωματική εργασία, με αποτέλεσμα να γίνονται ληθαργικοί και αδρανείς.

3.5. Αρχιτεκτονική IoT

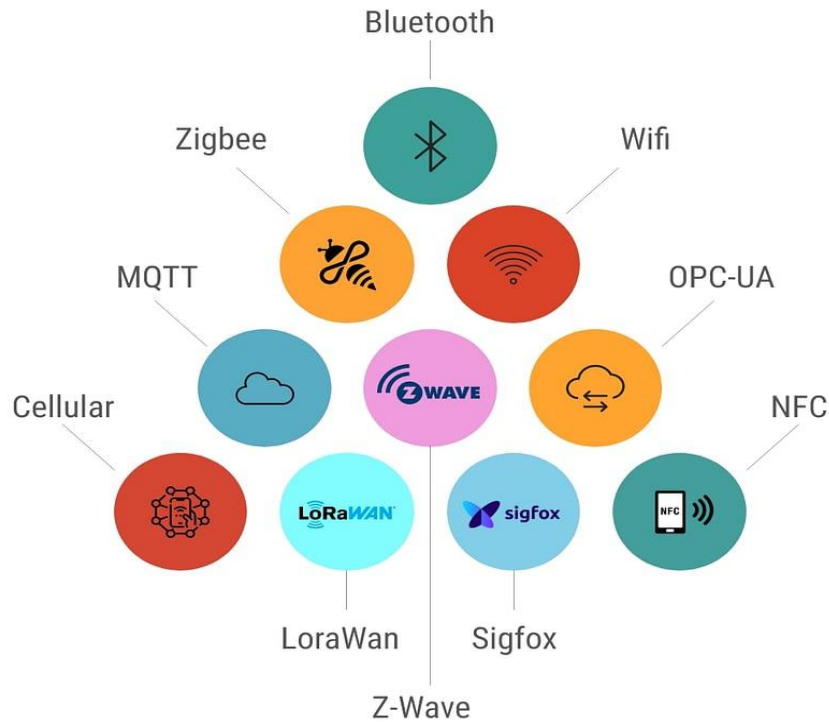
Η ιδέα και η δομή του IoT περιλαμβάνει τον συνδυασμό διαφορετικών τεχνολογιών. Για να επιτευχθεί η σωστή λειτουργία, είναι απαραίτητο να αλληλοεπιδρούν συνεχώς μεταξύ τους, δηλαδή να έχουν συνεχής αμφίδρομη επικοινωνία.

Η αρχιτεκτονική του IoT περιλαμβάνει:

- Μοναδικό τρόπο αναγνώρισης (identification-based connectivity).
- Διαλειτουργικότητα (interoperability)
- Αυτόνομη δικτύωση (autonomic networking)
- Αυτόνομη παροχή υπηρεσιών (autonomic services provisioning)
- Λειτουργικότητα βάσει τοποθεσίας (location-based capabilities)
- Ασφάλεια και Προστασία απορρήτου (security and privacy protection)
- Άμεση λειτουργία (plug and play)
- Διαχειρισιμότητα (manageability)

3.6. Επικοινωνία IoT

Σχεδόν όλοι οι αισθητήρες έχουν ως χαρακτηριστικό τη συνδεσιμότητα με την πόρτα του αισθητήρα. Μπορεί να είναι σε διάφορες μορφές όπως LAN ή Wi-Fi, Ethernet, PAN (Personal Area Network), ZigBee, UWB (Ultra-Wideband) και συνδέσεις Bluetooth.



Εικόνα 10 Πρωτόκολλα επικοινωνίας IoT

Πηγή :<https://hashstudioz.com/blog/top-iot-communication-protocols-2020/>

Φυσικά, ορισμένοι αισθητήρες δεν απαιτούν καμία σύνδεση με τον πελάτη και χρησιμοποιούν δίκτυο ευρείας περιοχής (WAN) όπως GSM, GPRS και LTE. Αυτοί οι αισθητήρες χρησιμοποιούν χαμηλή κατανάλωση ενέργειας, αλλά η μεταφορά δεδομένων και η ταχύτητα ανταλλαγής είναι χαμηλή και το ποσοστό μιας ανταλλαγής είναι μικρό. Κυρίως, αυτοί οι αισθητήρες σχηματίζουν ένα δίκτυο που ονομάζεται WSN (Wireless Sensor Network).

3.6.1. Device to Device

Η επικοινωνία από συσκευή σε συσκευή σημαίνει ότι δύο ή περισσότερες συσκευές συνδέονται απευθείας και επικοινωνούν μεταξύ τους χωρίς να απαιτείται διακομιστής. Μπορούν να επικοινωνούν μέσω πολλών τύπων δικτύων, συμπεριλαμβανομένων των δικτύων IP. Τα πιο κοινά πρωτόκολλα είναι το Bluetooth, το Z-Wave και το ZigBee. Αυτό το μοντέλο χρησιμοποιείται συχνά σε συστήματα οικιακού αυτοματισμού για τη μετάδοση μικρών πακέτων δεδομένων μεταξύ συσκευών με σχετικά χαμηλούς ρυθμούς δεδομένων, π.χ. σε οικιακές συσκευές όπως λαμπτήρες, διακόπτες φώτων, θερμοστάτες και κλειδαριές πόρτας, Αυτό το μοντέλο είναι δημοφιλές σε

φορητές συσκευές IoT που ανταλλάσσουν μικρές ποσότητες πληροφοριών, όπως συσκευές παρακολούθησης της ανθρώπινης καρδιάς μέσω ενός έξυπνου ρολογιού, μεταξύ των οποίων τα δεδομένα δεν μοιράζονται σε πολλά άτομα. Στο μοντέλο συνδεσιμότητας Device to Device επιτυγχάνεται εύκολα η ασφάλεια, επειδή η εμβέλεια των δυο συσκευών λόγω τεχνολογίας είναι μικρή.

3.6.2. Device to Cloud

Η επικοινωνία συσκευής με το Cloud (σύννεφο) περιλαμβάνει συσκευές IoT απευθείας συνδεδεμένες με υπηρεσίες Cloud Internet, όπως παρόχους υπηρεσιών εφαρμογών που είναι υπεύθυνοι για την ανταλλαγή δεδομένων και τον έλεγχο της ροής μηνυμάτων. Συνήθως χρησιμοποιεί μια παραδοσιακή ενσύρματη σύνδεση Ethernet ή Wi-Fi, αλλά μπορεί επίσης να χρησιμοποιήσει και τεχνολογία κινητής τηλεφωνίας. Η συνδεσιμότητα Cloud επιτρέπει στους χρήστες και τις εφαρμογές να έχουν απομακρυσμένη πρόσβαση σε συσκευές. Υποστηρίζει επίσης τη διάθεση ενημερώσεων λογισμικού σε συσκευές.

Για παράδειγμα η απομακρυσμένη παρακολούθηση του σπιτιού ενός χρήστη μέσω κάμερας, η οποία απαιτεί εύρος ζώνης, που παρέχεται από το Wi-Fi ή το Ethernet. Αυτό μπορεί να επιτευχθεί εάν τα δεδομένα της κάμερας μεταφερθούν στο Cloud έτσι ώστε οι απομακρυσμένοι χρήστες να έχουν πρόσβαση σε αυτά. Συγκεκριμένα, ο απομακρυσμένος χρήστης επικοινωνεί με την υποδομή Cloud και η αυτή αναμεταδίδει το περιεχόμενο της κάμερας. Η ασφάλεια αυτού του μοντέλου είναι πιο περίπλοκη από το μοντέλο από συσκευή σε συσκευή, επειδή περιλαμβάνει δύο διαφορετικούς τύπους διαπιστευτηρίων, διαπιστευτήρια πρόσβασης συσκευής (όπως κάρτα SIM κινητού τηλεφώνου) και διαπιστευτήρια πρόσβασης συσκευής.

3.6.3. Device to Gateway

Το μοντέλο συσκευής προς πύλη ουσιαστικά συνδέει συσκευές IoT μέσω μια πύλης με τις υπηρεσίες Cloud. Το λογισμικό που τρέχει σε μια τοπική συσκευή πύλης που λειτουργεί ως ενδιάμεσος μεταξύ των δύο συσκευών IoT και υπηρεσιών Cloud. Οι συσκευές πύλης παρέχουν ασφάλεια και πρόσθετες λειτουργίες, όπως η μετάφραση δεδομένων ή πρωτοκόλλων, έτσι είναι δυνατή η επικοινωνία μεταξύ συσκευών και υπηρεσιών IoT. Η συσκευή πύλης είναι συνήθως Smartphone και συσκευές IoT που

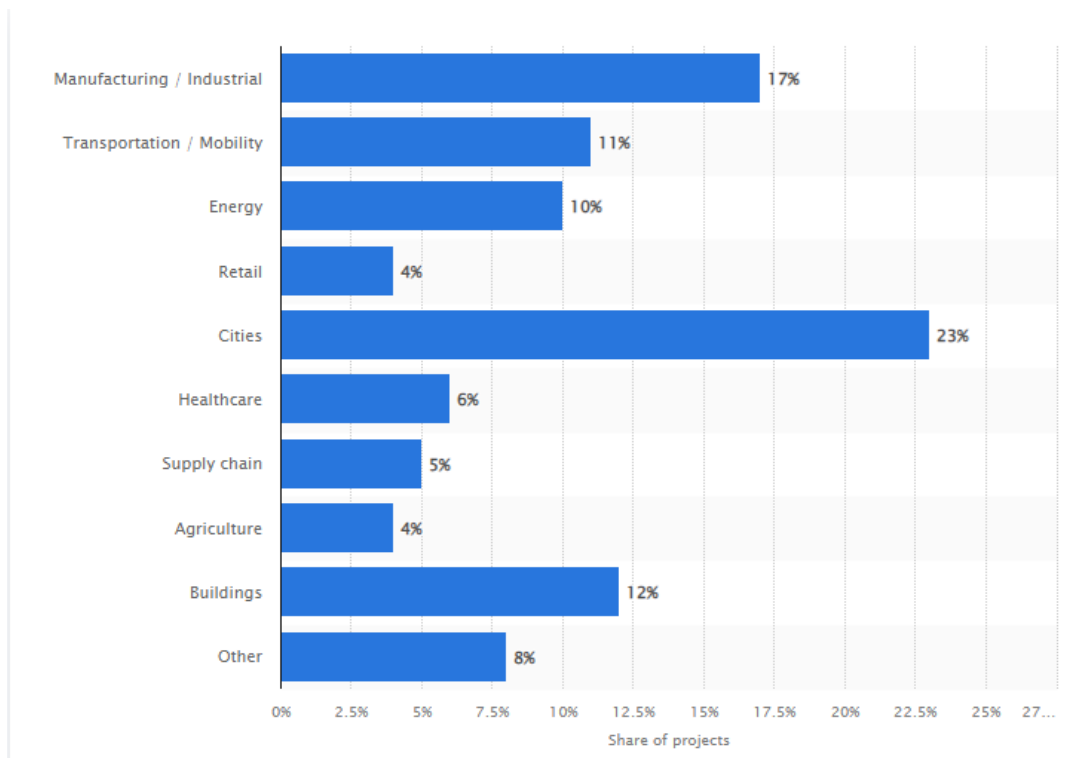
συνδέονται μέσω εφαρμογής (εφαρμογής) μεταφέρουν τα δεδομένα σε μια υπηρεσία Cloud.

3.6.4. Back-End Data-Sharing

Το μοντέλο κοινής χρήσης δεδομένων είναι μια αρχιτεκτονική επικοινωνίας που επιτρέπει οι χρήστες να εξάγουν και να αναλύουν δεδομένα συσκευών IoT από το Cloud. Αυτό το μοντέλο επικοινωνίας από συσκευή σε σύννεφο, που επιτρέπει την μεταφόρτωση δεδομένων μόνο για έναν πάροχο υπηρεσιών εφαρμογής. Η κοινή χρήση δεδομένων επιτρέπει τη συλλογή και ανάλυση των συλλεγόμενων δεδομένων από συσκευές IoT. Επιτρέπει επίσης σε τρίτους πρόσβαση σε δεδομένα αισθητήρων που έχουν ανέβει στο Cloud.

3.7. Αξιοποίηση του IoT στις «Έξυπνες Πόλεις»

Το IoT χρησιμοποιείται πλέον όλο και περισσότερο για διάφορους τομείς της καθημερινής ζωής. Σύμφωνα με στατιστικά στοιχεία, το έτος 2018 η χρήση του Διαδικτύου των Πραγμάτων στις έξυπνες πόλεις, βρίσκεται στην πρώτη παρέχοντας λύσεις που έχουν εφαρμοστεί σε πολλές πόλεις σε όλο τον κόσμο επιλύοντας μεγάλα προβλημάτων στην καθημερινότητα των ανθρώπων



Εικόνα 11 Κατανομή των (IoT) σε όλο τον κόσμο

(Πηγή: <https://www.statista.com/statistics/869335/world-internet-of-things-projects-by-segment-enterprise/>)

Μερικά παραδείγματα εφαρμογών έξυπνων πόλεων περιλαμβάνουν την έξυπνη στάθμευση, τον αστικό σχεδιασμό για την υγεία, χάρτες αστικού θορύβου, ανίχνευση με smartphone, ανακούφιση της κυκλοφοριακής συμφόρησης, έξυπνος φωτισμός, έξυπνοι δρόμοι και συστήματα διαχείρισης απορριμμάτων. Η συνδυασμένη χρήση των ΤΠΕ, με επικεφαλής τις προαναφερθείσες τεχνολογίες, μπορεί πλέον να καλύψει καινοτόμες δράσεις σε όλους τους τομείς των «έξυπνων πόλεων». Συνήθως αναφέρουμε τα εξής:

- **Μέτρηση ατμοσφαιρικής ρύπανσης.** Αισθητήρες μετρούν τα επίπεδα ατμοσφαιρικής ρύπανσης για να λάβουν μέτρα για την προστασία της υγείας των πολιτών.
- **Ευφυές Κτίριο.** Οι αισθητήρες μετρούν τους κραδασμούς και τις υλικές συνθήκες σε κτίρια, γέφυρες και ιστορικά μνημεία και παρέχουν πληροφορίες και ειδοποιήσεις σε περίπτωση ζημιάς.
- **Διαχείριση στόλου απορριματοφόρων.** Οι αισθητήρες ειδοποιούν την πληρότητα του κάδου και παρέχουν τη γεωγραφική θέση του στόλου για να βελτιστοποιήσουν τον υπολογισμό των διαδρομών συλλογής.

- **Έξυπνοι χώροι στάθμευσης.** Αισθητήρες που ανιχνεύουν αλλαγές στα μαγνητικά πεδία, ανιχνεύουν τη διαθεσιμότητα δωρεάν θέσεων στάθμευσης και παρέχουν πληροφορίες στους ενδιαφερόμενους οδηγούς μέσω μιας εφαρμογής.
- **Έξυπνα κτίρια.** Αξιοποιήστε την ενέργεια που διαχειρίζονται οι ηλιακοί συλλέκτες, ο φωτισμός και ο κλιματισμός μέσω αισθητήρων που αντιλαμβάνονται την ανθρώπινη παρουσία. Οι ενημερώσεις στις μετρήσεις κατανάλωσης ενέργειας είναι άμεσες, επιτρέποντας καλύτερη διαχείριση.
- **Μέτρηση στάθμης θορύβου.** Καταγραφή επιπέδου θορύβου σε πραγματικό χρόνο. Η παραβίαση ορίου εντοπίζεται αμέσως. Αισθητήρες παρόμοιας τεχνολογίας μπορούν να ανιχνεύσουν φωτιά με ακρίβεια 10 μέτρων, βοηθώντας έτσι το έργο της αστυνομίας.
- **Έξυπνη υγεία.** Οι αισθητήρες παρακολουθούν ζωτικές λειτουργίες σε άτομα, όπως η αρτηριακή πίεση και η καρδιακή λειτουργία, ή ανιχνεύουν πιθανές πτώσεις στο σπίτι για ηλικιωμένους και άτομα με ειδικές ανάγκες.
- **Διαχείριση κυκλοφορίας.** Οι αισθητήρες μπορούν να ανιχνεύσουν την ταχύτητα του οχήματος και τη συγκέντρωσή του. Αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν για να αποφευχθεί η συμφόρηση και να δείξουν στους οδηγούς την καλύτερη διαδρομή.
- **Μόλυνση του περιβάλλοντος.** Οι αισθητήρες ανιχνεύουν πιθανές διαρροές λυμάτων εργοστασίων, αέρια καύσης και το επίπεδο ρύπανσης σε ποτάμια ή ωκεανούς από ξαφνικές αλλαγές θερμοκρασίας για την προστασία των δασών.
- **Έξυπνος φωτισμός.** Οι αισθητήρες ανιχνεύουν την ένταση του φυσικού φωτός και την παρουσία ανθρώπων ή αυτοκινήτων για να προσαρμόσουν το φωτισμό στους δημόσιους χώρους στο επιθυμητό επίπεδο.
- **Έξυπνη ύδρευση.** Αισθητήρες παρακολουθούν την ποιότητα του νερού στο δίκτυο ύδρευσης και επίσης εντοπίζουν ενδεχόμενες διαρροές.
- **Έξυπνη ασφάλεια.** Αισθητήρες εντοπίζουν τις δονήσεις του εδάφους ή την ένταση των ηλεκτρομαγνητικών πεδίων για προειδοποίηση σεισμού, ελέγχουν τη στάθμη των ποταμών και της θάλασσας για το ενδεχόμενο πλημμύρας ή τσουνάμι.
- **Κλειστά κυκλώματα παρακολούθησης.** Χρησιμοποιούνται για τον έλεγχο πληρωμών (π.χ. στα μέσα μαζικής μεταφοράς) και τη δημόσια ασφάλεια. (Βογιατζής-Κολοκοτρώνης, 2017).

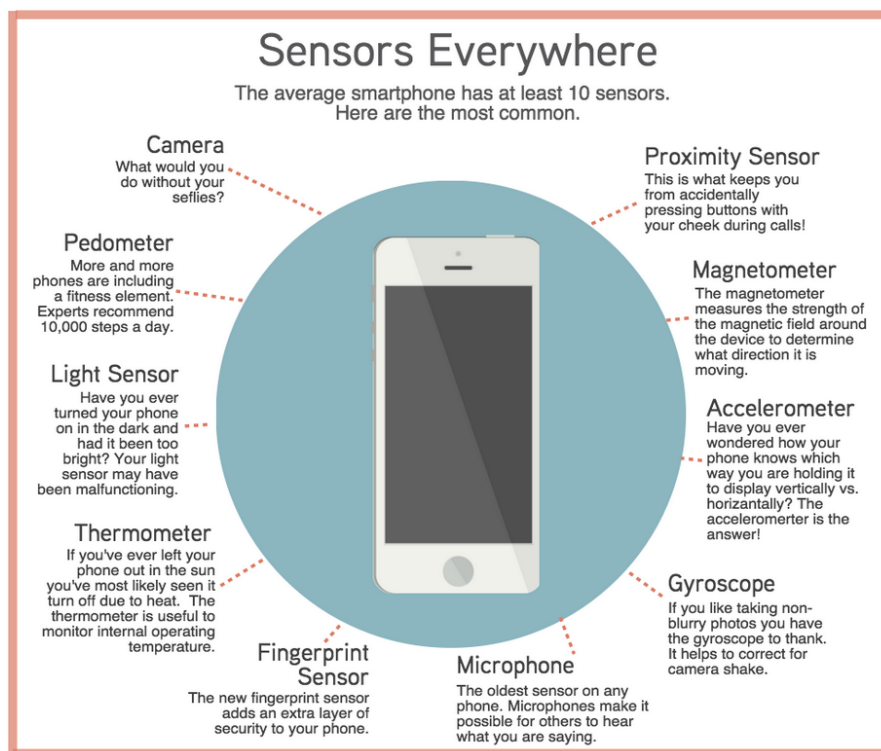
4. Εφαρμογές για Έξυπνες Πόλεις

4.1. Smartphones και εφαρμογές

Η χρήση των smartphones εξαπλώνεται με ταχύτατο ρυθμό. Σύμφωνα με εκτιμήσεις υπολογίζεται ότι μέσα σε 4 χρόνια, το 70% του παγκόσμιου πληθυσμού θα έχει ένα smartphone και το 80% των δεδομένων θα δημιουργηθεί από φορητές συσκευές. Τα smartphones, είναι συσκευές που συνδυάζουν την ευκολία χρήσης των κινητών τηλεφώνων, με τις δυνατότητες επεξεργαστικής ισχύος των σύγχρονων υπολογιστών και τις αυξημένες δυνατότητες αποθήκευσης δεδομένων και συνδεσιμότητάς τους. Τα smartphones τελευταίας τεχνολογίας είναι εφοδιασμένα με πολλαπλούς ενσωματωμένους αισθητήρες και ως εκ τούτου μπορούν να συλλέξουν σε πραγματικό χρόνο δεδομένα και πληροφορίες, που μπορούν να χρησιμοποιηθούν για την ανάπτυξη καινοτόμων εφαρμογών για τις «έξυπνες πόλεις».

Τα σύγχρονα smartphones, εκτός από τη δυνατότητα καταγραφής εικόνας και ήχου, είναι εξοπλισμένα με ενσωματωμένους αισθητήρες όπως (van Dijk, 2015, p.42):

- Μετρητή επιτάχυνσης φωτός
- Βαρόμετρο (αισθητήρας ανίχνευσης ατμοσφαιρικής πίεσης)
- Κάμερα
- GPS (αισθητήρας γεωγραφικού εντοπισμού)
- Γυροσκόπιο (αισθητήρας ανίχνευσης προσανατολισμού)
- Μετρητή μαγνητικού πεδίου (αισθητήρας για τη λειτουργία πυξίδας)
- Μικρόφωνο Εγγύτητας
- Θερμόμετρο



Εικόνα 12 Αισθητήρες στα Smartphone

Πηγή: <https://github.com/Apt3kStudio/Phone/wiki/Sensors-on-Smartphones>

Έτσι, οι πολίτες με την αξιοποίηση των εφαρμογών για smartphones με την χρήση των αισθητήρων τους έχουν τη δυνατότητα να επωφεληθούν λαμβάνοντας δεδομένα από τις τοπικές αρχές (για παράδειγμα από μια εφαρμογή που τους ειδοποιεί για τα ακριβή δρομολόγια των μέσων μαζικής μεταφοράς), αλλά και να ωφελήσουν την κοινότητα στέλνοντας δεδομένα που βελτιώνουν την ποιότητα ζωής στη πόλη αλλά και την ικανότητα λήψης αποφάσεων από τις τοπικές αρχές.

Η λειτουργικότητα μιας εφαρμογής και ο αποτελεσματικός σχεδιασμός της για την υποστήριξη της εκτέλεσης των λειτουργικών της προδιαγραφών είναι δύο χαρακτηριστικά που υπερβαίνουν τη συνεχή χρήση της εφαρμογής και την ενασχόληση του χρήστη με την εφαρμογή. Ειδικά στον σχεδιασμό εφαρμογών, η ελκυστική σχεδίαση του UI, και οι κατάλληλοι ενσωματωμένοι μηχανισμοί απορρήτου, ενθαρρύνουν την ψυχολογική δέσμευση των χρηστών καθώς προσελκύουν την προσοχή, προκαλούν αλληλεπίδραση και είναι μοναδική και θετική εμπειρία χρήστη.

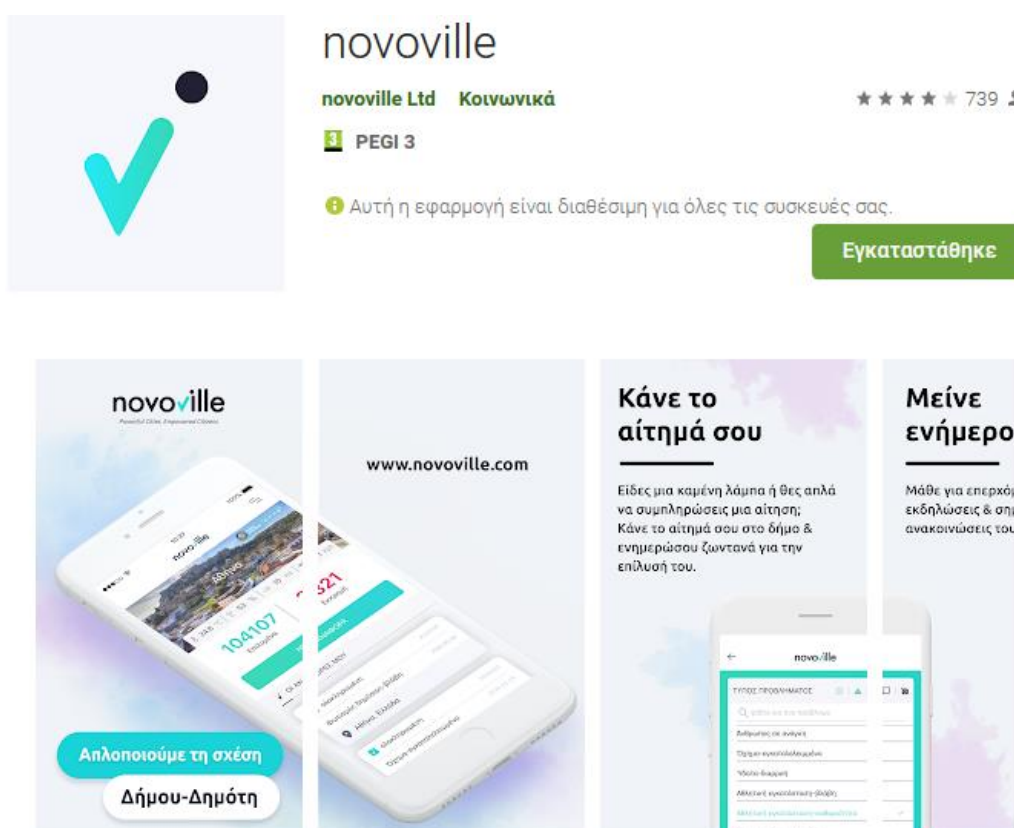
Στα στοιχεία μιας εφαρμογής θα πρέπει να δίνονται ιδιαίτερη προσοχή κατά τη διαδικασία σχεδιασμού, διότι έχουν επίσης αντίκτυπο στη λειτουργικότητά της, όπως το περιεχόμενο, τα γραφικά, οι δυνατότητες, η πλοήγηση και οι αλληλεπιδράσεις.

Στο back-end περιβάλλον, πρέπει επίσης να ληφθούν υπόψη η βάση δεδομένων καθώς και πρόσθετοι παράγοντες που επηρεάζουν τη λειτουργικότητα και αφορούν τόσο τη συσκευή όσο και το δίκτυο, όπως το λειτουργικό σύστημα, η κάλυψη δικτύου και τα χαρακτηριστικά υλικού της συσκευής. Όσον αφορά τα χαρακτηριστικά απόδοσης της εφαρμογής, στόχος είναι να βελτιωθεί η ικανότητα της εφαρμογής να προσαρμόζεται στις ανάγκες και προτιμήσεις των χρηστών.

4.2. Εφαρμογές «Έξυπνων Πόλεων»

Παράλληλα με την εξάπλωση των smartphones οι εφαρμογές smartphone αναπτύσσονται πλέον ραγδαία σε ένα άκρως ανταγωνιστικό περιβάλλον και η λειτουργικότητά τους είναι πολύ σημαντική για τις παρεχόμενες υπηρεσίες. Υπάρχει πλέον πληθώρα εφαρμογών για «Έξυπνες Πόλεις» στο Google Play και στο App Store. Ενδεικτικά θα δούμε 2 εφαρμογές που χρησιμοποιούνται σε ελληνικές πόλεις.

4.2.1. Novoville



Εικόνα 13 Η εφαρμογή Novoville Πηγή:

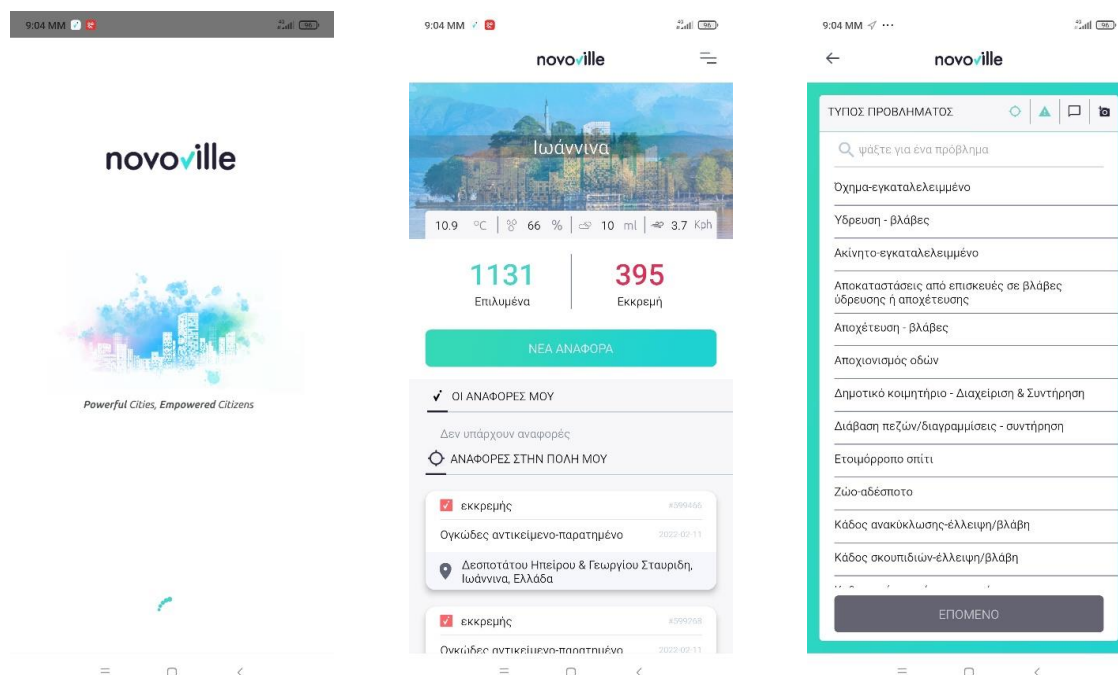
<https://play.google.com/store/apps/details?id=com.novoville>

Πρόκειται για μια από τις πιο δημοφιλείς εφαρμογή με πάνω από 50.000 εγκαταστάσεις καταγραφής και επίλυσης προβλημάτων για πόλεις με στόχο τη βελτίωση της ποιότητας ζωής στις πόλεις αυτές. Η Nononville χρησιμοποιείται από τον Δήμο Αθηναίων, τον Δήμο Πειραιώς, τον Δήμο Λάρισας, τον Δήμο Ιωαννιτών, τον Δήμο Κέρκυρας κ.α. Η Nononville αποτελεί μια πλατφόρμα ενεργοποίησης πολιτών που γεφυρώνει το κενό ανάμεσα στους πολίτες και τους Δήμους και φιλοδοξεί να ενεργοποιήσει όλους τους πολίτες να αλλάξουν μαζί τις πόλεις τους. Οι πολίτες μπορούν να δηλώνουν με εύκολο και γρήγορο τρόπο προβλήματα δημοσίου ενδιαφέροντος, να εκφράζουν την άποψη σας & να συν-δημιουργούν το μέλλον της γειτονιάς τους απαντώντας σε σύντομες δημοσκοπήσεις για τα πράγματα που συμβαίνουν γύρω σας. Η εφαρμογή επιτρέπει στους πολίτες :

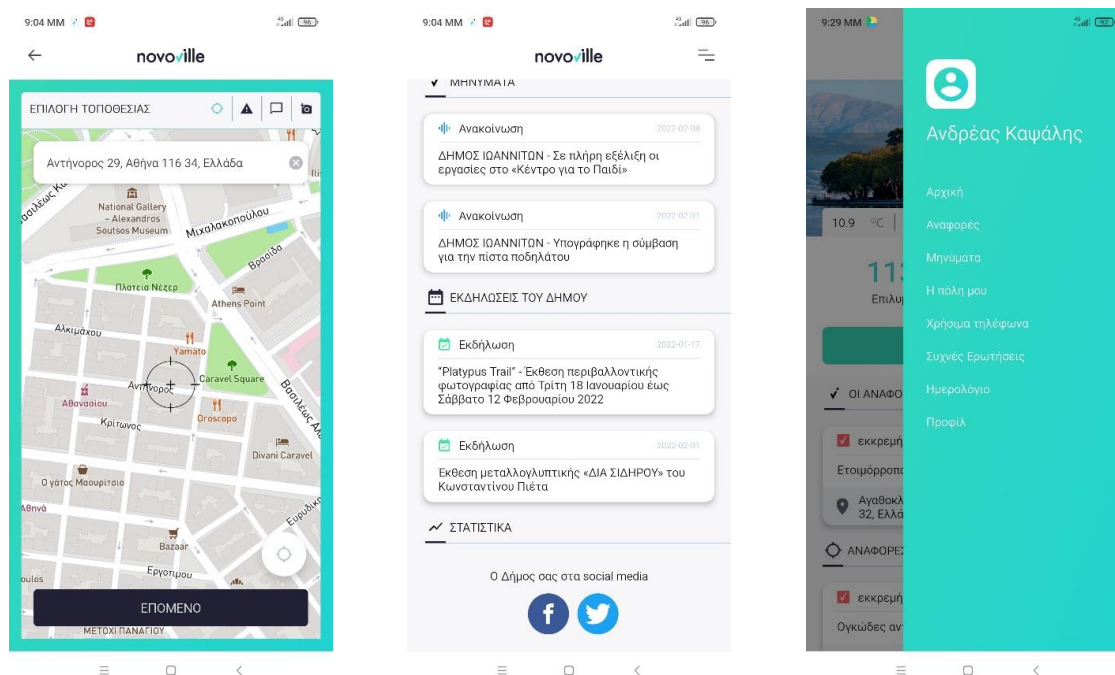
- Να αναφέρουν με απλό και γρήγορο τρόπο προβλήματα που συναντούν στην καθημερινότητα τους της πόλη όπως. λακκούβες, προβληματική οδική σήμανση, δημόσιο φωτισμό που δεν λειτουργεί καθώς και δεκάδες άλλα προβλήματα που συναντάτε καθημερινά και να ενημερώνεστε για την πρόοδο επίλυσης τους.
- Να συμμετέχουν στην διαμόρφωση των αποφάσεων που επηρεάζουν την γειτονιά τους μέσω σύντομων δημοσκοπήσεων.
- Να βρουν γρήγορα σημεία ενδιαφέροντος όπως φαρμακεία, βενζινάδικα, ΚΕΠ κλπ. και προηγηθούν σε αυτά με ακρίβεια.
- Να επικοινωνήσουν απευθείας με τις δημοτικές υπηρεσίες.
- Να ενημερωθούν εύκολα για εκδηλώσεις, ανακοινώσεις και σημαντικές ενημερώσεις μόνο για τις περιοχές που σας ενδιαφέρουν.

Όλα αυτά με διαδραστικούς χάρτες που επιτρέπουν γρήγορη πλοήγηση σε όλα τα σημεία. Στις πόλεις που έχουν εφαρμόσει η Nononville οι αναφορές και αιτήσεις των πολιτών φτάνουν ζωντανά στο διαχειριστικό τους σύστημα. Αυτό τους βοηθά να γίνουν πιο αποδοτικοί και άμεσα ανταποκρίσου στην διαχείριση των θεμάτων που σας αφορούν. Οποιαδήποτε αλλαγή σε κάποιο από τα αιτήματά σας φθάνει απευθείας στο κινητό σας με ηλεκτρονικές ειδοποιήσεις.

4.2.1.1. Ενδεικτικές οθόνες εφαρμογής

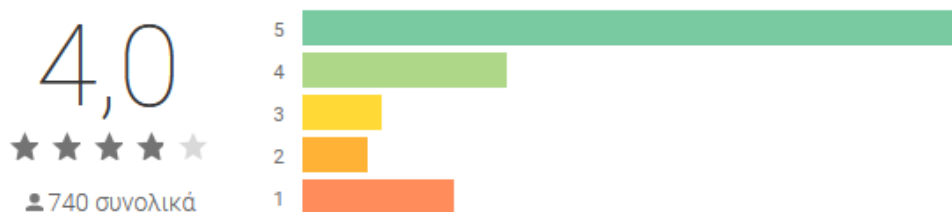


Εικόνα 14 Ενδεικτικές Οθόνες Novonville (a)



Εικόνα 15 Ενδεικτικές Οθόνες Novonville (b)

4.2.1.2. Αξιολόγηση εφαρμογής

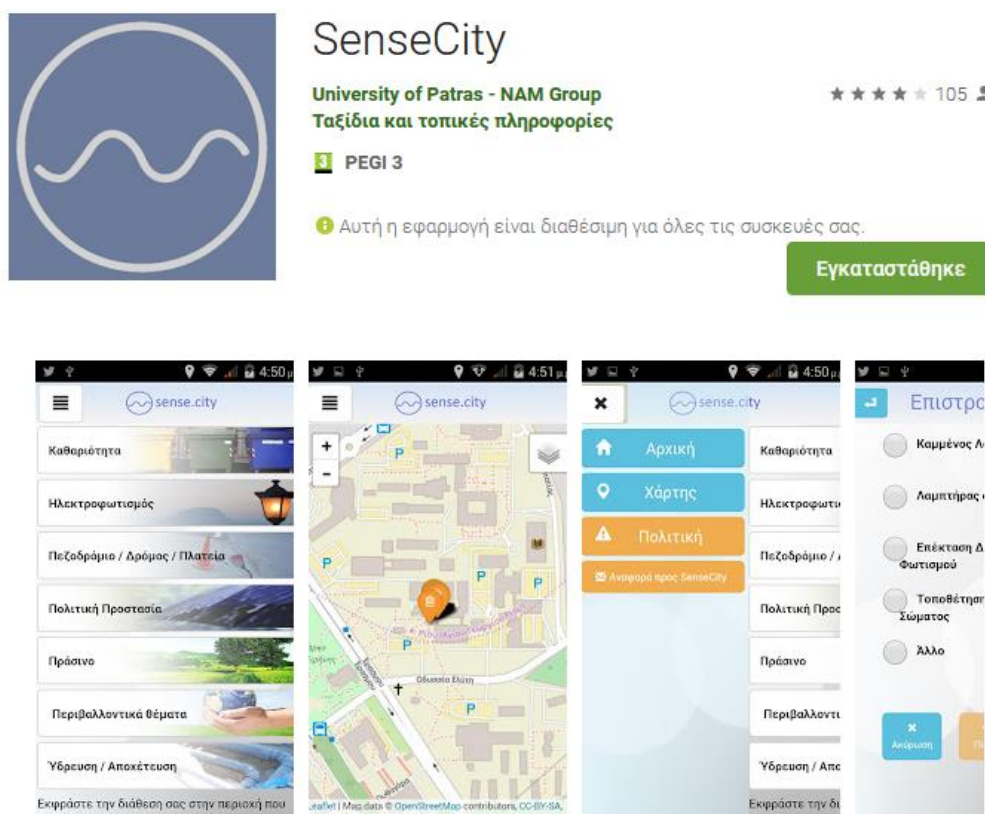


Εικόνα 16 Αξιολόγηση εφαρμογής Novonville Πηγή:

<https://play.google.com/store/apps/details?id=com.novonville>

Σύμφωνα με τις αξιολογήσεις που υποβλήθηκαν στο Google Play, η εφαρμογή αξιολογήθηκε από 740 άτομα και όπως φαίνεται και στο διάγραμμα συγκεντρώνει συντριπτικά τις περισσότερες θετικές κριτικές.

4.2.2. SenceCity



Εικόνα 17 Η εφαρμογή SenceCity

Πηγή: <https://play.google.com/store/apps/details?id=gr.upatras.ece.nam.sensecity>

Πρόκειται για μια εφαρμογή αναφοράς προβλημάτων που αναπτύχθηκε από το Πανεπιστήμιο Πατρών και χρησιμοποιείται από τον Δήμο Πάτρας, Δήμο Δυτικής Αχαΐας, Δήμο Πύργου, Δήμο Διδυμοτείχου κ.α.

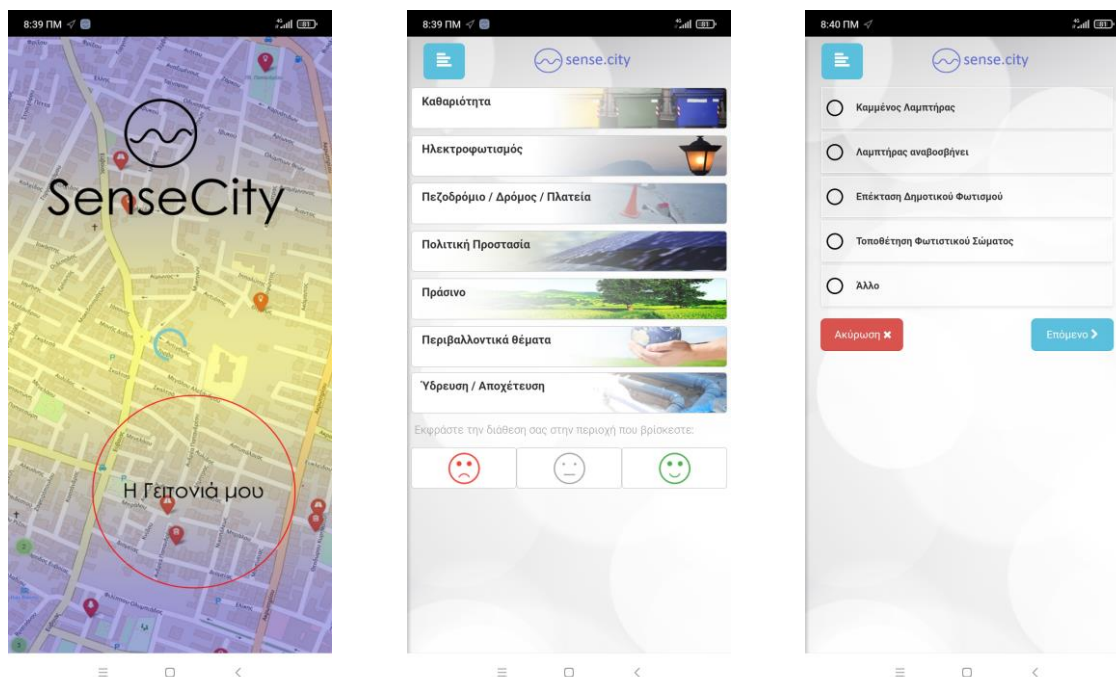
Η πλατφόρμα sense.city (<http://sense.city>) παρέχει εργαλεία στους πολίτες για να ενεργοποιήσουν τη δημιουργικότητα, το σχεδιασμό και την επικοινωνία τους και εμπλέκει περισσότερο τον προβληματισμό και τη δράση τόσο των πολιτών όσο και των δημοτικών και δημόσιων υπηρεσιών. Χρησιμοποιώντας την πλατφόρμα sense.city, οι πολίτες με τις δικές τους συσκευές επικοινωνίας (κινητά τηλέφωνα) ή μέσω εφαρμογής sense.city, οι πολίτες μπορούν να δημοσιεύσουν σε πραγματικό χρόνο θέματα και προβλήματα για κάτι που συμβαίνει στην πόλη τους, να ενημερώσουν τους συμπολίτες τους και το δήμο για προβλήματα και περιστατικά που συμβαίνουν κάθε στιγμή.

Οι πολίτες συμμετέχουν ενεργά στις διαδικασίες και επιλύουν προβλήματα που αφορούν τη ζωή τους στην πόλη. Βοηθούν στην αστική ανάπτυξη και βελτιώνουν τη σχέση μεταξύ των πολιτών και των δημόσιων υπηρεσιών της πόλης.

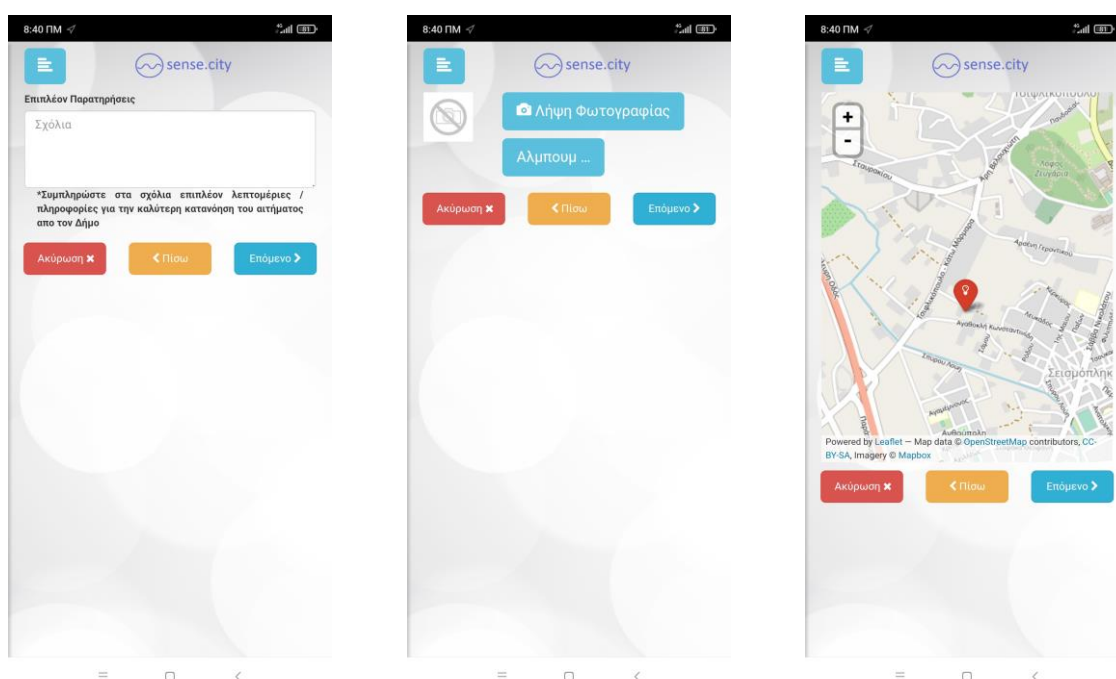
Με την εφαρμογή οι πολίτες μπορούν να στείλουν ένα θέμα ή πρόβλημα στις δημόσιες υπηρεσίες της πόλης χρησιμοποιώντας τις κινητές συσκευές τους για να την αναφορά προβλημάτων στην πόλη σε πραγματικό χρόνο.

Μέσα από την πλατφόρμα [Sense.City](http://sense.city) ενισχύεται η συν δημιουργία καθώς προσπαθεί να βάλει τα εργαλεία για την ενεργοποίηση των πολιτών, τη δημιουργικότητα, το σχεδιασμό και την επικοινωνία στα χέρια του καθενός και καλεί για συλλογική σκέψη και δράσεις τόσο τους πολίτες όσο και τους φορείς της πόλης.

4.2.2.1. Ενδεικτικές οθόνες εφαρμογής

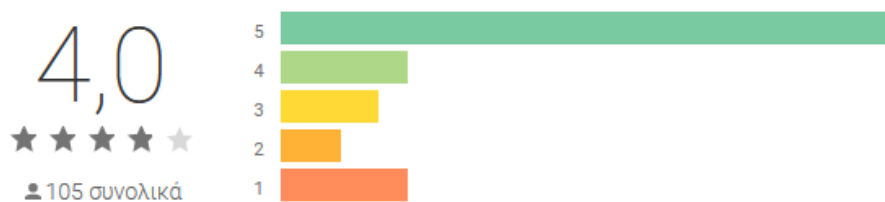


Εικόνα 18 Ενδεικτικές Οθόνες SenceCity (a)



Εικόνα 19 Ενδεικτικές Οθόνες SenceCity (b)

4.2.2.2. Αξιολόγηση εφαρμογής



Εικόνα 20 Αξιολόγηση εφαρμογής SenceCity

Πηγή: <https://play.google.com/store/apps/details?id=gr.upatras.ece.nam.sensecity>

Σύμφωνα με τις αξιολογήσεις που υποβλήθηκαν στο Google Play , η εφαρμογή αξιολογήθηκε από 105 άτομα και όπως φαίνεται και στο διάγραμμα συγκεντρώνει και αυτή πάρα πολλές θετικές κριτικές.

5. Ανάλυση και σχεδίαση Mobile εφαρμογής «MySmartCity».

Η mobile εφαρμογή **MySmartCity** είναι μια ολοκληρωμένη εφαρμογή που παρέχει όλα τα απαραίτητα εργαλεία και ψηφιακές υπηρεσίες για την υποστήριξη των πολιτών και των τοπικών κοινοτήτων αξιοποιώντας τεχνολογίες IoT.

Μέσω της εφαρμογής οι χρήστες θα μπορούν να καταγράψουν σε χάρτη διάφορα καθημερινά πρόβλημα στην πόλη ώστε να ενημερωθεί η αρμόδια υπηρεσία του Δήμου και να επιληφθεί για την επίλυσή τους. Ακόμη μπορούν να ενημερώνονται για προβλήματα που έχουν αναφερθεί από άλλους χρήστες.

Αξιοποιώντας τις τεχνολογίες IoT οι χρήστες μπορούν να βλέπουν σε πραγματικό χρόνο τις περιβαλλοντικές συνθήκες (Θερμοκρασία -Υγρασία) στην πόλη ή να βλέπουν σε χάρτη τους διαθέσιμους σταθμούς στάθμευσης καθώς και τις διαθέσιμες σε αυτούς θέσεις. Μπορούν ακόμη να ενημερωθούν για Νέα, Ανακοινώσεις, Εκδηλώσεις που αφορούν την πόλη.

Ο τρόπος σχεδίασης αυτής της εφαρμογής έγινε λαμβάνοντας υπόψιν ως κεντρικό άξονα τον χρήστη της εφαρμογής. Στόχος λοιπόν είναι η δημιουργία μιας εφαρμογής που θα τον εξυπηρετήσει, και θα τον διευκολύνει στην καθημερινότητά του. Τελευταίο αλλά όχι λιγότερο σημαντικό είναι η ευκολία χρήσης, δηλαδή, πόσο θα φιλικό στο περιβάλλον του χρήστη, το οποίο δεν θα τον δυσκολεύει στη χρήση και θα είναι εύκολο να γίνει κατανοητό.

Ακόμη λήφθηκε υπόψιν ότι την εφαρμογή θα μπορεί να τη χρησιμοποιήσουν χρήστες οποιουδήποτε επιπέδου χωρίς εξειδίκευση και εξοικείωση με την τεχνολογία.

Γενικές απαιτήσεις για το σύστημα:

- Φιλικό και εύχρηστο γραφικό περιβάλλον χρήστη.
- Υποστήριξη διαφορετικών συσκευών σύμφωνα με τα τεχνικά χαρακτηριστικά, την έκδοση λογισμικού και μέγεθος οθόνης.
- Χρήση Cloud βάσης δεδομένων με όλες τις πληροφορίες και φωτογραφίες ζητημάτων που αναφέρθηκαν από τους χρήστες.
- Δημιουργία λογαριασμού από τους χρήστες ή είσοδος με έλεγχο ταυτότητας λογαριασμού Google
- Εμφάνιση αναφοράς χάρτη με Marker χρησιμοποιώντας τους χάρτες της Google.
- Χρήση χρωμάτων στους Marker για πιο εύκολη ενημέρωση για την κατάσταση της αναφοράς

- Λήψη ειδοποιήσεων push για Ανακοινώσεις - Νέα.
- Έλεγχος συνδεσιμότητας Internet
- Ενημέρωση ενεργοποίησης GPS

Η Mobile εφαρμογή **MySmartCity** στην παρούσα φάση είναι συμβατή μόνο έξυπνες συσκευές με λειτουργικό σύστημα Android 4.4 και νεότερο.

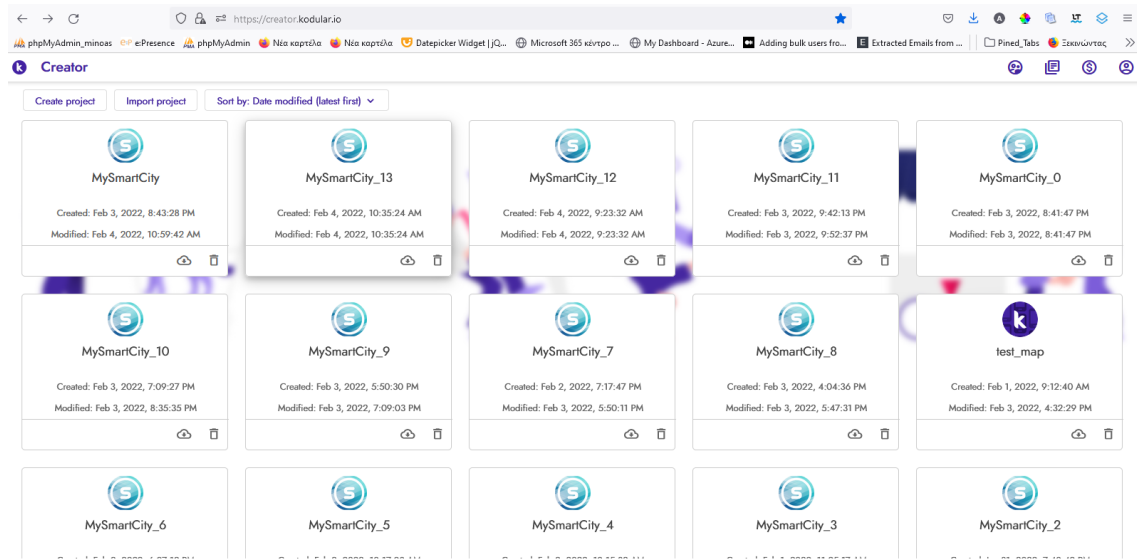
Η εφαρμογή σχεδιάστηκε έτσι ώστε ο χρήστης να έχει τις παρακάτω δυνατότητες:

- να δημιουργεί την δική του αναφορά προβλήματος προς το δήμο. Η εφαρμογή αυτόματα καταγράφει την διεύθυνση καθώς και τις γεωγραφικές συντεταγμένες του χρήστη. Στην αναφορά ο χρήστης μπορεί να κάνει μια σύντομη περιγραφή και να στείλει με φωνογραφία από το πρόβλημα δημιουργώντας και τον δικό του marker πάνω στον χάρτη. Για κάθε αναφορά που ο χρήστης υποβάλει, λαμβάνει σχετικό email επιβεβαίωσης.
- να μπορεί να βλέπει την λίστα με τα προβλήματα που έχουν αναφερθεί από άλλους συμπολίτες και επιλέγοντας κάποιο από αυτά να δει αναλυτικές λεπτομέρειες.
- να μπορεί να βλέπει σε πραγματικό χρόνο τις περιβαλλοντικές συνθήκες στην πόλη (Θερμοκρασία, Υγρασία) τις οποίες λαμβάνει μέσω Internet από αισθητήρα (IoT) που είναι τοποθετημένος στην πόλη.
- να μπορεί να βλέπει στον χάρτη σε πραγματικό χρόνο τους διαθέσιμους σταθμούς στάθμευσης αυτοκινήτων καθώς και τις διαθέσιμες θέσεις αυτών τις οποίες λαμβάνει μέσω Internet από αισθητήρες (IoT) που τοποθετημένοι στους αντίστοιχους σταθμούς στάθμευσης.
- να μπορεί να βλέπει σε χάρτη υπό την μορφή marker τα προβλήματα που έχουν αναφερθεί από άλλους συμπολίτες και επιλέγοντας κάποιο από αυτά να δει συνοπτικά στοιχεία.
- να μπορεί να βλέπει πληροφορίες σχετικά με τον Δήμο, τον Δήμαρχο, το Δημοτικό Συμβούλιο και πληροφορίες για υπηρεσίες.
- να μπορεί να βλέπει πληροφορίες για Εκδηλώσεις που πρόκειται να πραγματοποιηθούν στον Δήμο.
- να μπορεί να βλέπει σημαντικά Νέα και Ανακοινώσεις του Δήμου που αφορούν τους πολίτες.
- Να λαμβάνει μηνύματα που θα προωθεί ο δήμος προς του πολίτες

5.1. Εργαλεία ανάπτυξης της mobile εφαρμογής

5.1.1. Kodular

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε το **KODULAR** , μια δωρεάν διαδικτυακή σουίτα για ανάπτυξη εφαρμογών για κινητά (<https://creator.kodular.io/>.)



Εικόνα 21 Η πλατφόρμα Kodular

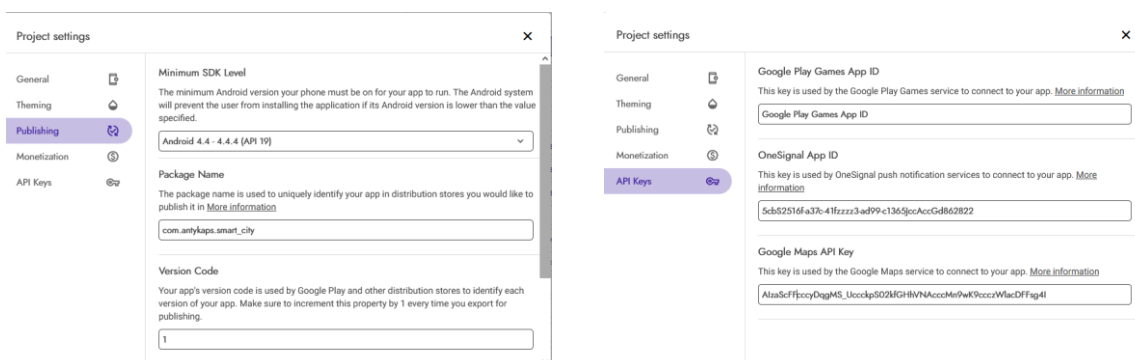
Το Kodular (πρώην Makeroid) παρέχει κυρίως έναν διαδικτυακό δημιουργό Android εφαρμογών τύπου drag-and-drop, στον οποίο ο καθένας μπορεί να δημιουργήσει οποιοδήποτε είδος εφαρμογής χωρίς να προγραμματίσει ούτε μια γραμμή. Το Kodular είναι βασισμένο πάνω στο διάσημο MIT AppInventor.

Τα πλεονεκτήματα του Kodular έναντι του App Inventor είναι ότι το Kodular διαθέτει πολλά περισσότερα πράγματα (εξαρτήματα) που λείπουν στο AI2 ενώ υπάρχουν διαθέσιμες πάρα πολλές επεκτάσεις.

Το Kodular βασίζεται στη λογική όταν συμβεί κάτι, τότε να εκτελεστεί μια εντολή ή μια σειρά από εντολές. Το περιβάλλον του Kodular είναι ιδιαίτερα ελκυστικό και εύχρηστο και υπάρχουν πληθώρα οδηγιών χρήσης δωρεάν διαθέσιμες στο διαδίκτυο, μαθήματα εκμάθησης, εκπαιδευτικά εγχειρίδια, βιβλιοθήκες και Community's.

Για την δημιουργία εφαρμογών με το Kodular είναι απαραίτητα τα εξής: η χρήση ενός φυλλομετρητή ,σύνδεση στο Internet , ενός τηλεφώνου με λογισμικό Android και ενός λογαριασμού στην Google.

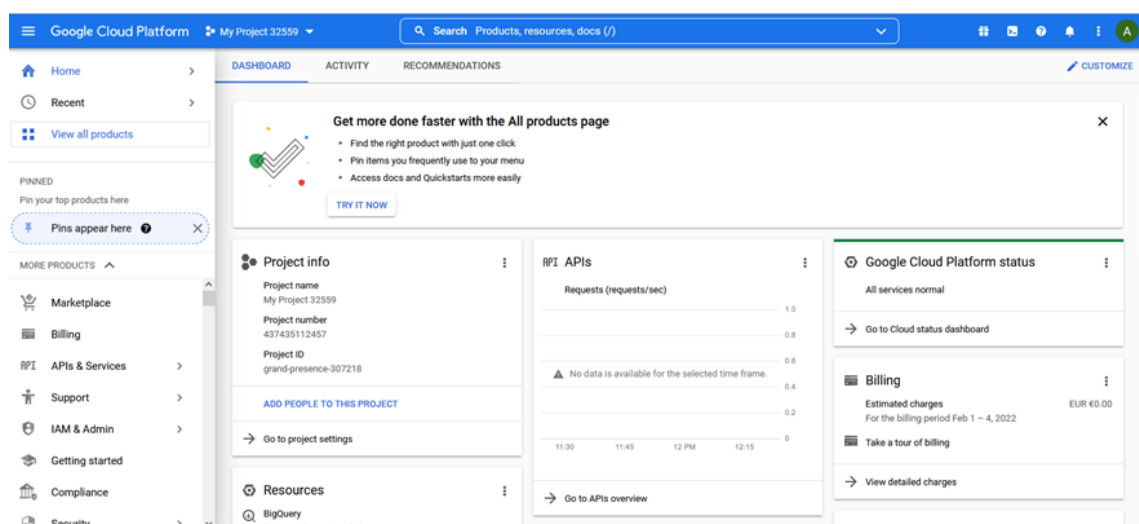
Για την υλοποίηση της εφαρμογής είναι απαραίτητη η παραμετροποίηση του Kodular ώστε να επικοινωνεί επιτυχώς με το Google Map Api και με το OneSignal



Εικόνα 22 Παραμετροποίηση Kodular

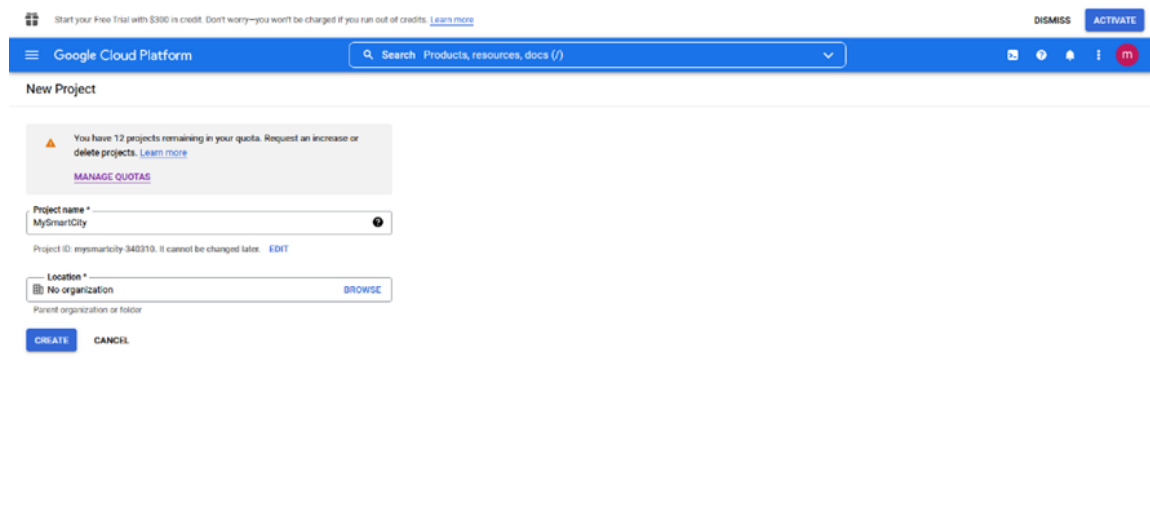
5.1.2. Google Map Api

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το Google Maps Android API. Το API αυτό είναι παρέχει το απαραίτητο κλειδί, ώστε να εμφανίσουμε χάρτες μαζί με τα στοιχεία τους στην εφαρμογή. Η δημιουργία του API key γίνεται μέσα από το περιβάλλον διαχείρισης της Google <https://console.cloud.google.com/>



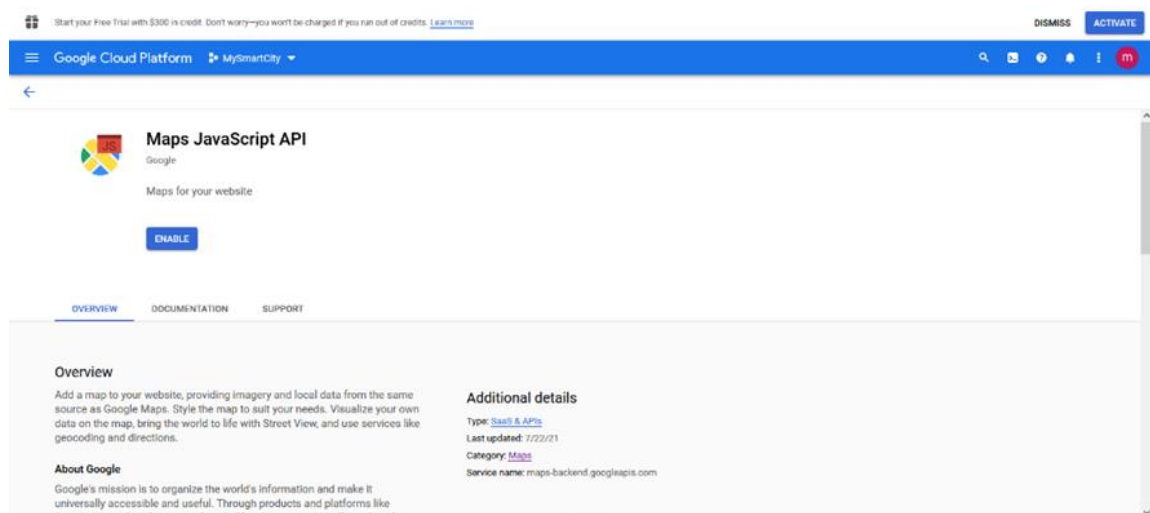
Εικόνα 23 Google Cloud Console

Στο οποίο θα πρέπει να δημιουργήσουμε το σχετικό Project, για την εφαρμογή μας θα δημιουργήσουμε το Project “MySmartCity”



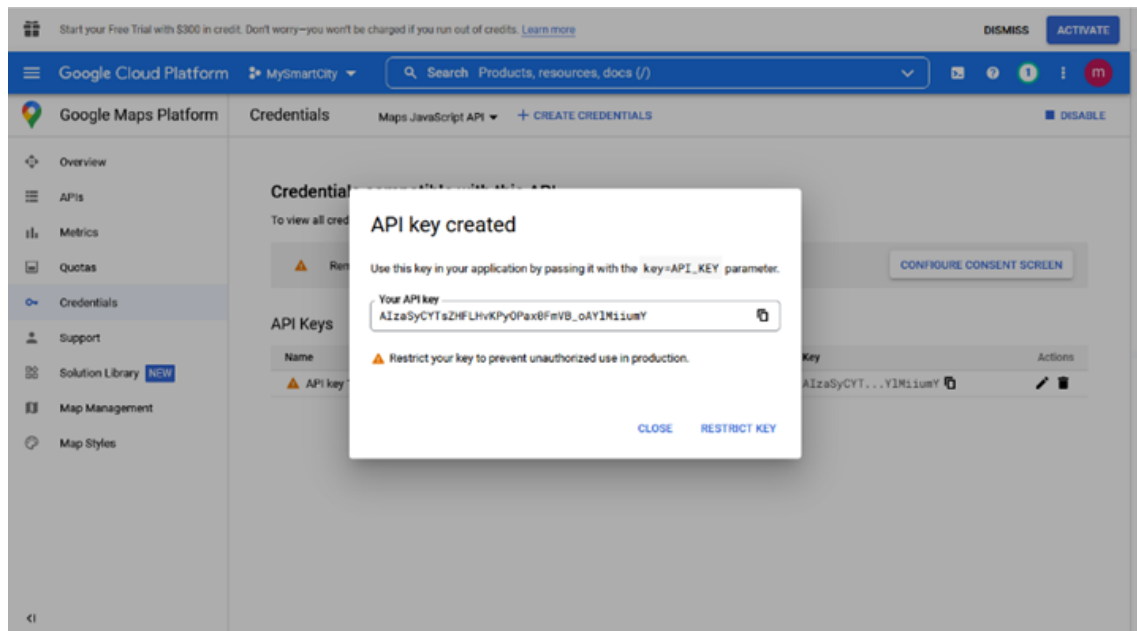
Εικόνα 24 Δημιουργία Project

Ενεργοποιούμε το Maps JavaScript Api που θέλουμε



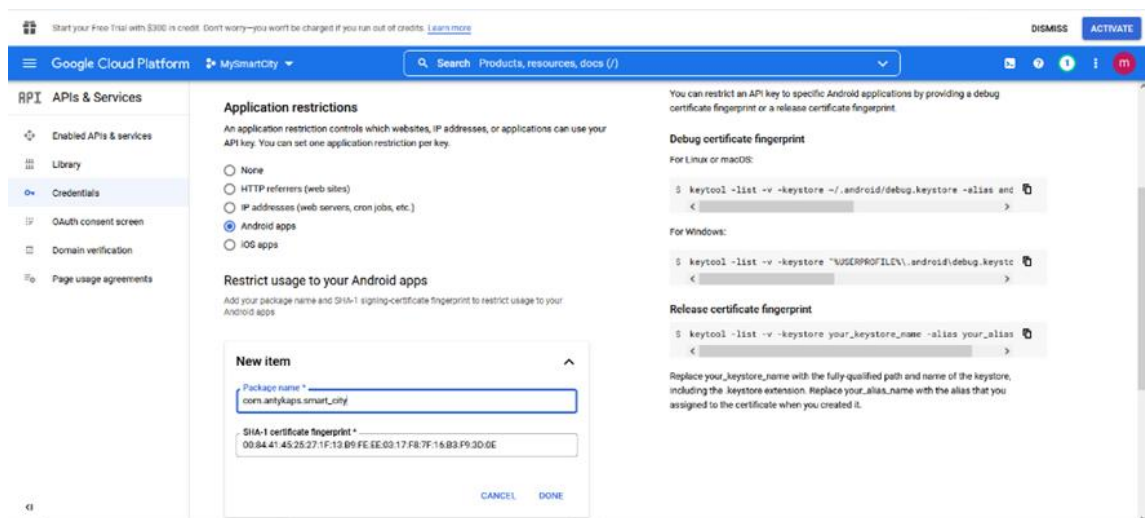
Εικόνα 25 Ενεργοποίηση Maps JavaScript Api

Και στην συνέχεια δημιουργούμε τα κατάλληλα Credentials.



Εικόνα 26 Δημιουργία API Key

Για λόγους ασφαλείας ορίζουμε την εφαρμογή που θα επιτρέπει κλήσεις σε αυτό το API.



Εικόνα 27 Δικαιώματα εφαρμογών στο API

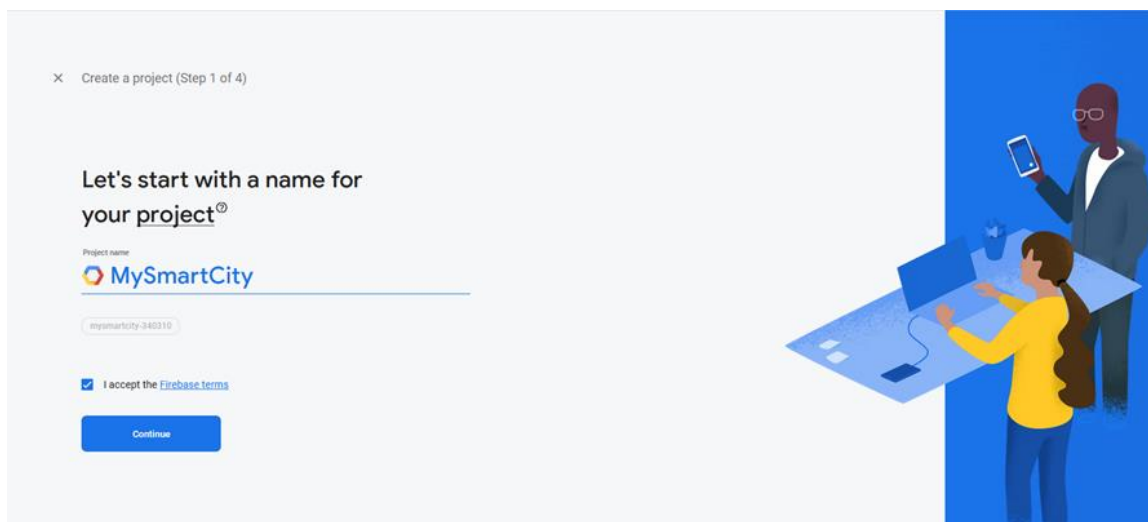
Τα στοιχεία Package Name και SHA-1 certificate fingerprint τα είναι στοιχεία της εφαρμογής και τα λαμβάνουμε από το Kodular.

5.1.3. Firebase

Για την αποθήκευση των αναφορών και των άλλων δεδομένων της εφαρμογής χρησιμοποιήθηκε η Google Firebase , μια από τις γνωστές πλατφόρμες ανάπτυξης εφαρμογών για κινητά και διαδικτυακές εφαρμογές, η οποία παρέχει στους προγραμματιστές διάφορα εργαλεία και υπηρεσίες τα οποία συμβάλουν την ανάπτυξη εφαρμογών υψηλής ποιότητας .

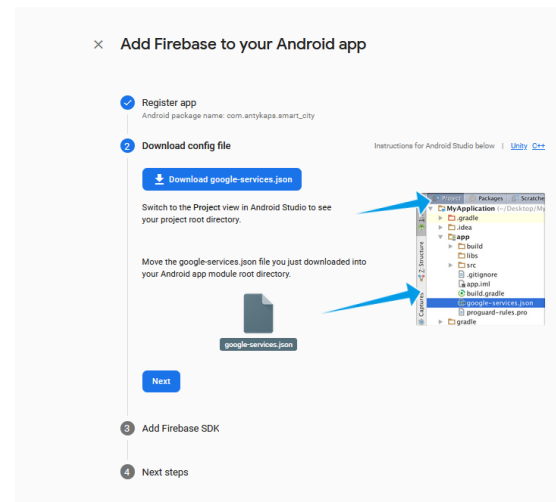
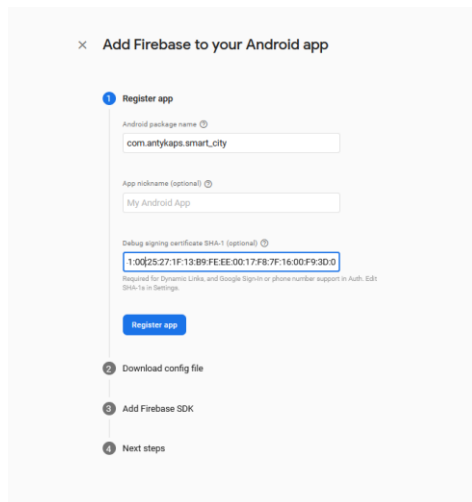
5.1.3.1. Δημιουργία Firebase Project

Αρχικά θα πρέπει να δημιουργήσουμε το σχετικό Project



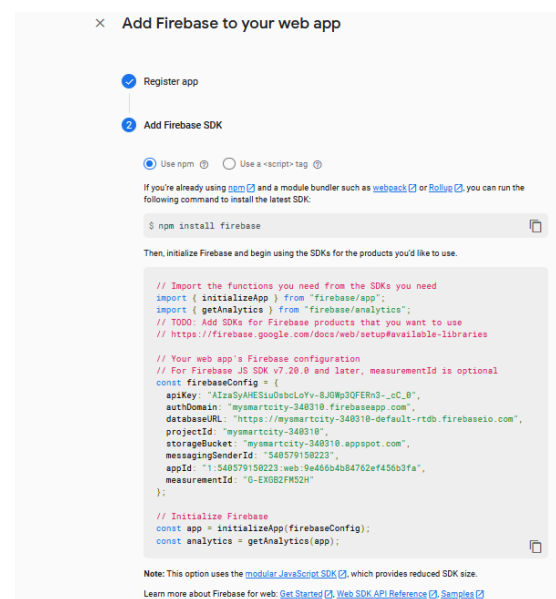
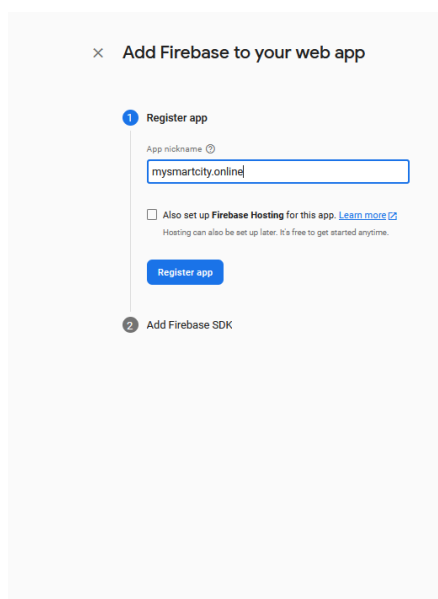
Εικόνα 28 Δημιουργία Firebase Project

Στα project settings ορίζουμε τις εφαρμογές καθώς και το είδος τους (android , ios , web) που θα έχουν πρόσβαση στο Project. Έτσι για την χρήση της Firebase από την android εφαρμογή μας ,επιλέγουμε Android και καταχωρούμε το Package Name και SHA-1 certificate και αφού κάνουμε register τη εφαρμογή λαμβάνουμε ένα αρχείο σύνδεσης της Firebase με την mobile εφαρμογή. Το αρχείο αυτό το κάνουμε upload στα assets της εφαρμογής στο Kodular



Εικόνα 29 Δημιουργία Credentials

Για την χρήση της firebase από την web εφαρμογή κάνουμε προσθήκη νέας Web εφαρμογής και αφού κάνουμε register την εφαρμογή παράγεται το SDK που είναι απαραίτητο για την σύνδεση της Firebase με την web εφαρμογή.



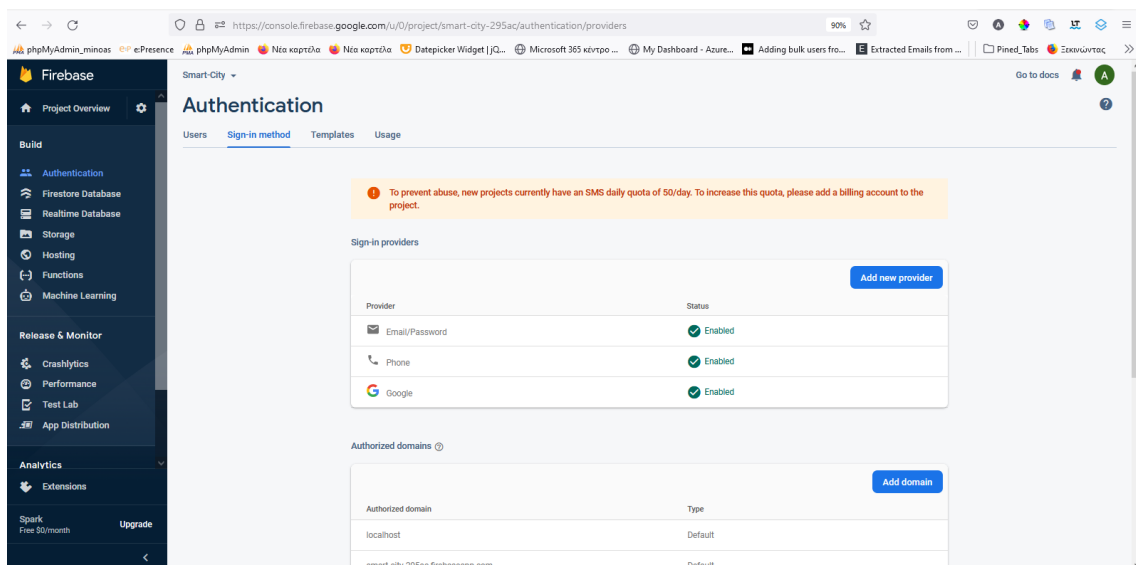
Εικόνα 30 Web app registration

5.1.3.2. Firebase Authentication

Ο σχεδιασμός της εφαρμογής προβλέπει την χρήση της μόνο σε εγγεγραμμένους - πιστοποιημένους χρήστες . Για την λειτουργία αυτή θα αξιοποιήσουμε την αυθεντικοποίηση χρηστών (Authentication) της Firebase.

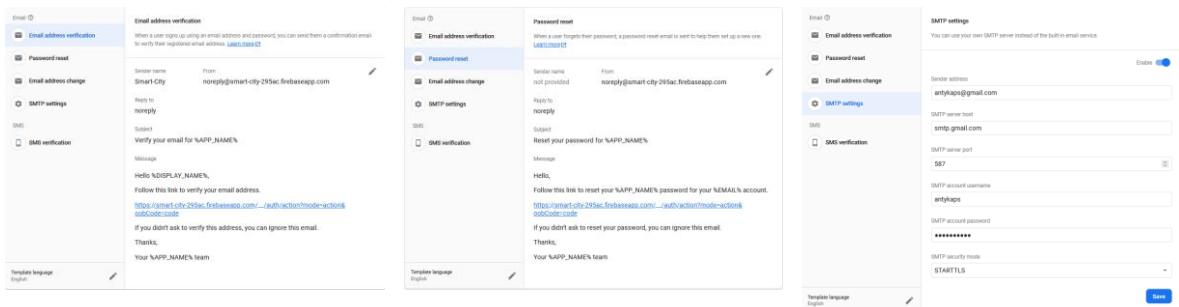
Κάνοντας χρήση της αυθεντικοποίησης χρηστών που παρέχει η Firebase οι εφαρμογές επαληθεύουν την ταυτότητά τους χρήστη . Η γνώση αυτής της ταυτότητας κρατά ασφαλές τα δεδομένα του χρήστη στο Cloud.

Οι Provider αυθεντικοποίησης που ενεργοποιήσαμε είναι μέσω Google Account , με την χρήση email , και με αριθμό τηλεφώνου (δεν αξιοποιείται ακόμη στην εφαρμογή).



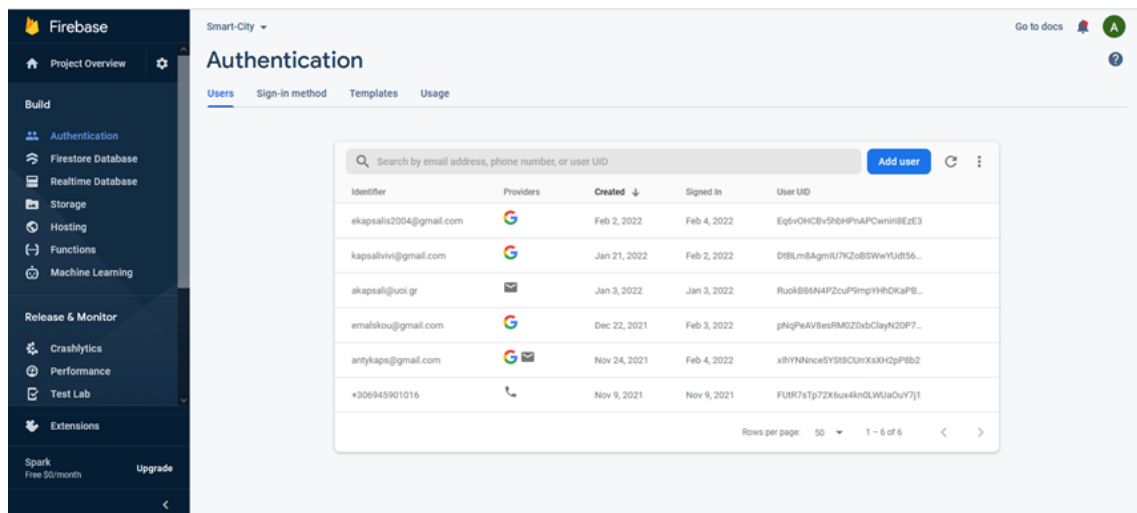
Εικόνα 31 Firebase Authentication

Από την καρτέλα Templates ρυθμίζουμε τα πρότυπα απαντήσεων για τις διαδικασίες Επαλήθευσης, Αλλαγής Password , Αλλαγής Email και κάνουμε τις σχετικές ρυθμίσεις για την αποστολή email μέσω SMTP Server.



Εικόνα 32 Παραμετροποίηση Authenticator

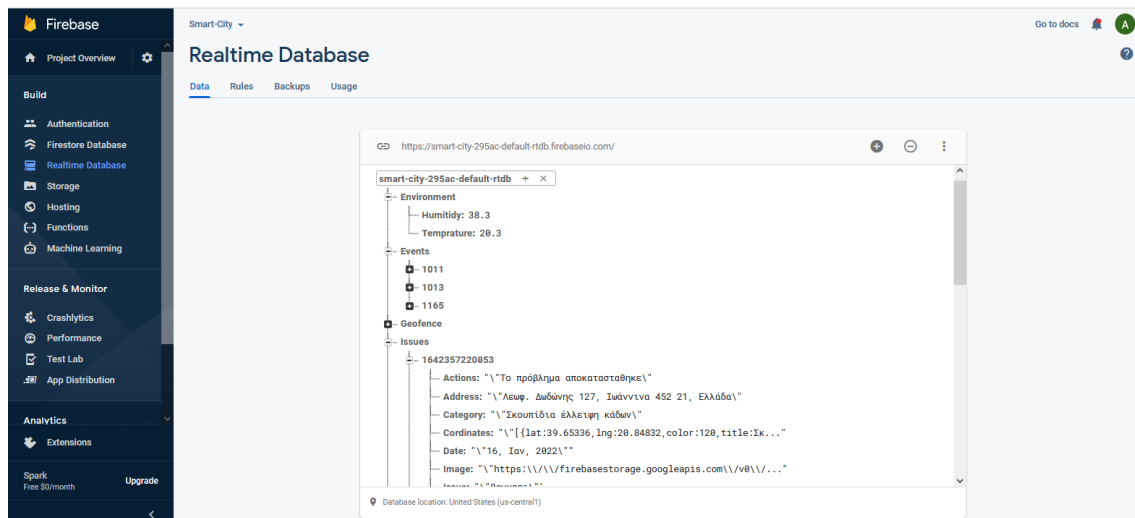
Οι χρήστες που έχουν εγγραφεί- αυθεντικοποιηθεί εμφανίζονται στην καρτέλα Users



Εικόνα 33 Authenticated Users

5.1.3.3. Realtime Database

Για την αποθήκευση των δεδομένων της εφαρμογής δημιουργήθηκε η βάση δεδομένων σε πραγματικό χρόνο της Firebase (Realtime Database).

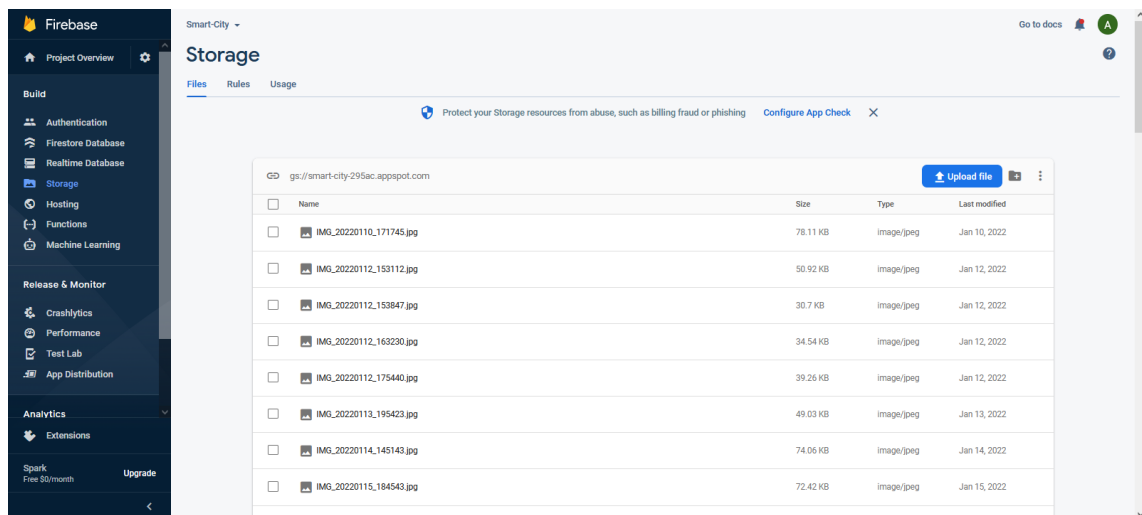


Εικόνα 34 Realtime Database

Η Realtime Database της Firebase είναι μια NoSQL βάση δεδομένων η οποία φιλοξενείται στο Cloud της Google και μας δίνει την δυνατότητα να αποθηκεύουμε και να συγχρονίζουμε δεδομένα χρηστών σε πραγματικό χρόνο. Σε αντίθεση με τις σχεσιακές βάσεις δεδομένων, η Firebase αποθηκεύει δεδομένα σε μορφή JSON. Σκεφτείτε κάθε σειρά σε μια σχεσιακή βάση δεδομένων ως αντικείμενο JSON (το οποίο είναι βασικά μη ταξινομημένο ζεύγος κλειδιού-τιμής). Έτσι, το όνομα της στήλης γίνεται κλειδί και η τιμή που είναι αποθηκευμένη σε αυτήν τη στήλη για μια συγκεκριμένη γραμμή είναι η τιμή.

5.1.3.4. Storage

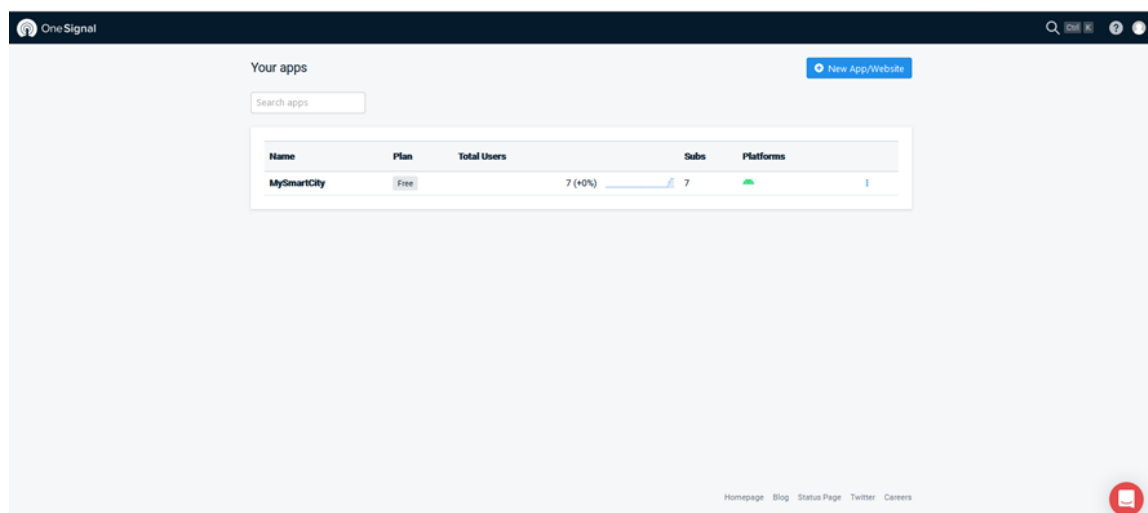
Για την αποθήκευση των αρχείων της εφαρμογής χρησιμοποιήσαμε τον χώρο αποθήκευσης (Storage) της Firebase. Το Storage υποστηρίζει αποθήκευση αρχείων, όπως εικόνες, ήχος και βίντεο που δημιουργούνται από χρήστες εφαρμογών. Η «μεταφόρτωση» και η «λήψη» των αρχείων είναι απολύτως ασφαλείς και δεν εξαρτάται από την ποιότητα της σύνδεσης στο διαδίκτυο.



Εικόνα 35 Storage

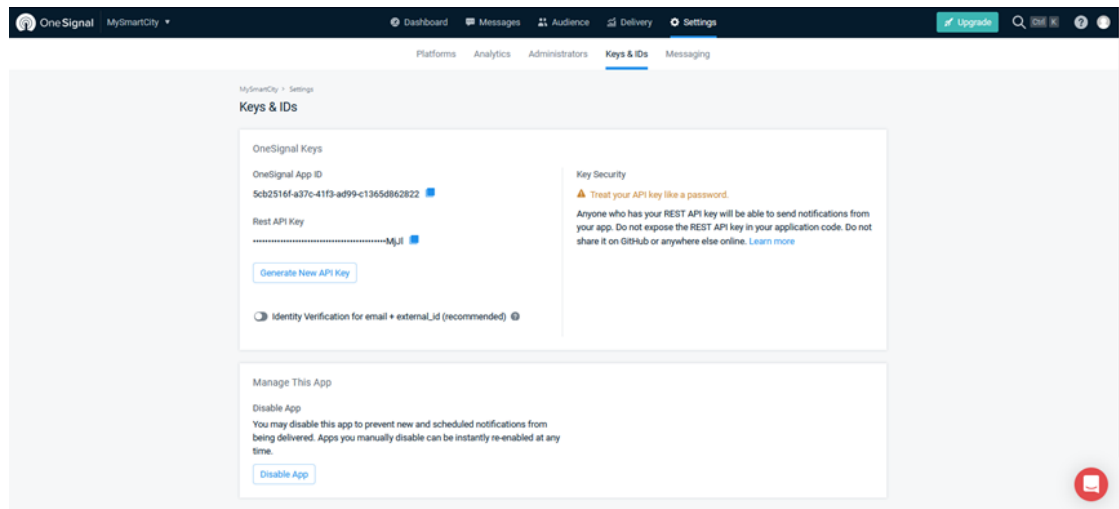
5.1.4. OneSignal

Η OneSignal (<https://onesignal.com/>) είναι μια υπηρεσία Push Notification που επιτρέπει ειδοποιήσεις σε Web Push, iOS, Android, Huawei, Chrome, Unity 3D, Amazon, Solar2D και άλλα. Στην εφαρμογή μας με την χρήση της OneSignal, σε συνεργασία με την Firebase δίνεται η δυνατότητα στην εφαρμογή να στέλνει ειδοποιήσεις push στους εγγεγραμμένους χρήστες της εφαρμογής που έχουν λογαριασμό στην Firebase. Για την χρήση της OneSignal είναι απαραίτητη η δημιουργία λογαριασμού στην (δυνατότητα αυθεντικοποίησης με Google Account) , στην συνέχεια δημιουργούμε την εφαρμογή. Η δωρεά άδεια χρήσης του OneSignal μας επιτρέπει την απεριόριστη προώθηση ειδοποιήσεων για μία όμως εφαρμογή .



Εικόνα 36 OneSignal

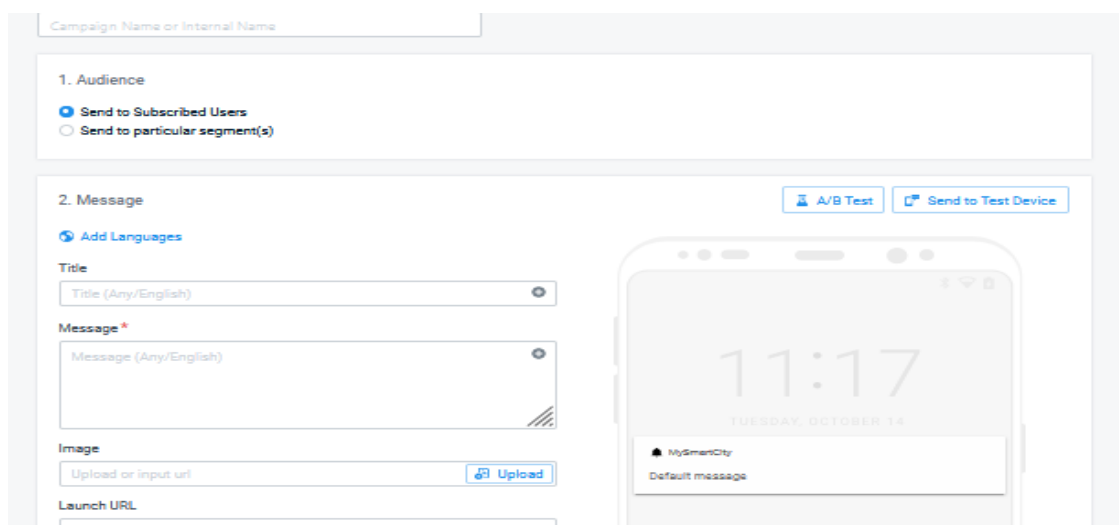
Και από την επιλογή settings → Keys&IDs λαμβάνουμε το OneSignal App ID που θα το καταχωρήσουμε στο Kodular ώστε η εφαρμογή μας να δέχεται τα μηνύματα που κάνει push ο διαχειριστής



Εικόνα 37 Παραμετροποίηση OneSignal

Η αποστολή push ειδοποιήσεων γίνεται με από το Web περιβάλλον της εφαρμογής OneSignal.

Μπορούμε να προωθήσουμε μηνύματα κείμενου ή και εικόνων σε όλους τους εγγεγραμμένους χρήστες ή σε group χρηστών.



Εικόνα 38 Αποστολή (Push) μηνυμάτων

5.2. Υλοποίηση της mobile εφαρμογής

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν τρεις οθόνες.

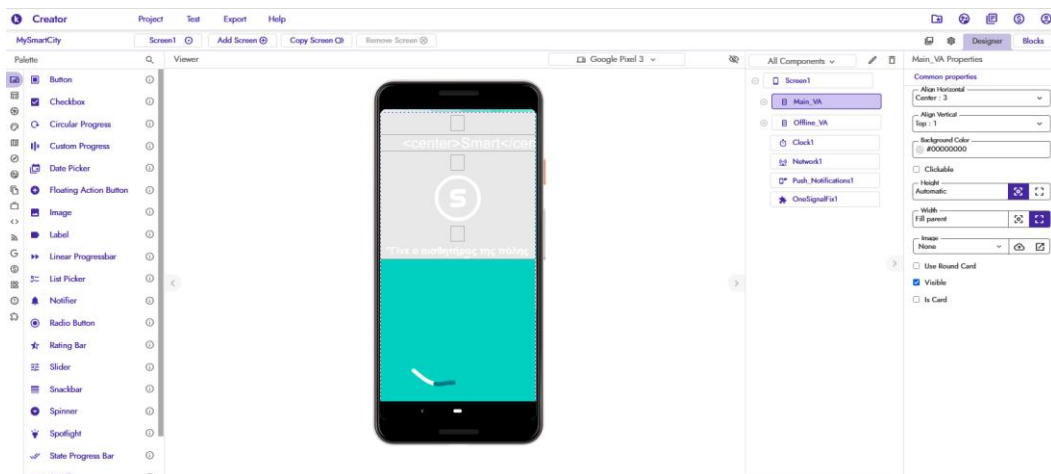
Η κάθε μία οθόνη εξυπηρετεί και ένα συγκεκριμένο σκοπό στην εφαρμογή.

- **Screen 1** (Οθόνη έναρξης με το λογότυπο)
- **Login** (Οθόνη για την είσοδο στην εφαρμογή , για την δημιουργία λογαριασμού και το μενού λειτουργιών της εφαρμογής)
- **Report** (Οθόνη που περιέχει όλες τις λειτουργίες της εφαρμογής)

Ο σχεδιασμός της εφαρμογής έγινε μέσα από τον Designer του Kodular .

Παρακάτω αναλύονται όλες οι οθόνες που αναφέρθηκαν προηγουμένως και ο τρόπος υλοποίησής τους κάθε μιας ξεχωριστά

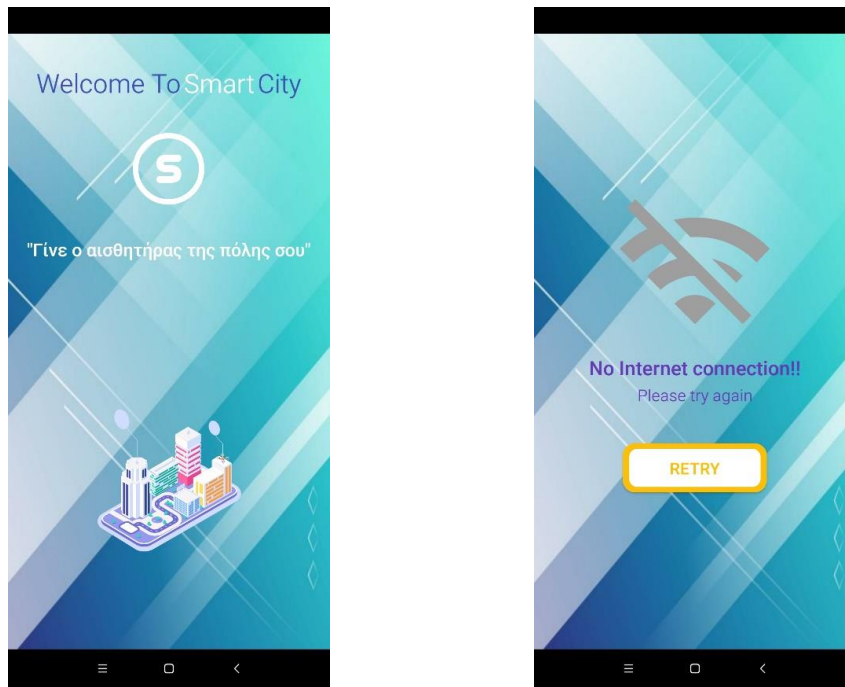
5.2.1. Screen 1



Εικόνα 39 Η οθόνη Screen 1

Η οθόνη “**Screen1**” εμφανίζεται αυτόματα με την εκτέλεση της εφαρμογής και έχει καθαρά εισαγωγικό χαρακτήρα και οργανώνεται σε 2 (δύο) Vertical Arrangements.

- **Main_VA** (Με την εκκίνηση της εφαρμογής εκτελείται ένα αρχείο κινούμενης εικόνας Lottie File τύπου json.)
- **Offline_VA** (Εμφάνιση σχετικού μηνύματος αν διαπιστωθεί αδυναμία σύνδεσης στο Internet).

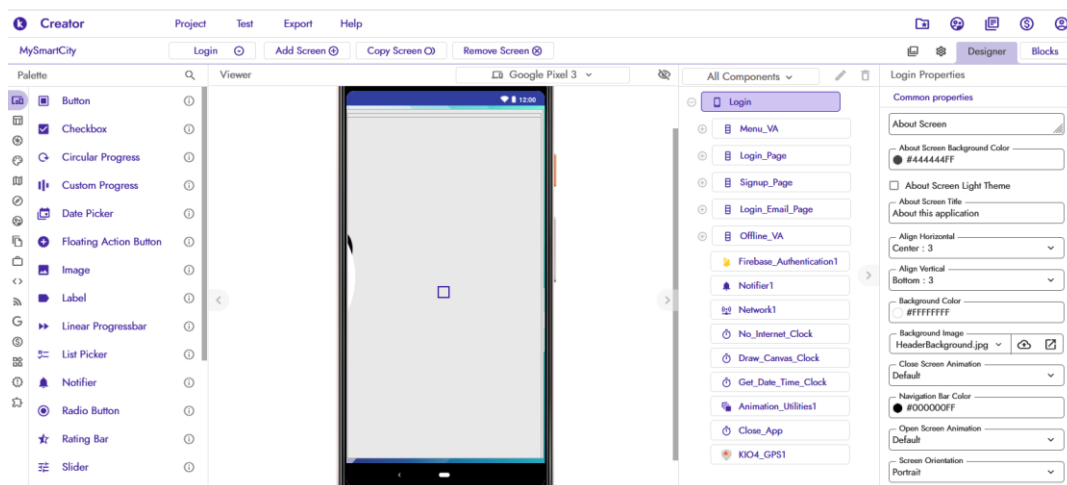


Εικόνα 40 Ενδεικτικές οθόνες της Screen 1

Βασικό χαρακτηριστικό της οθόνης είναι το στοιχείο “**Network**” διότι εφαρμογή απαιτεί σύνδεση στο Internet , έτσι με το στοιχείο αυτό αν ανιχνευτεί αδυναμία σύνδεσης εμφανίζεται η σχετική οθόνη που πληροφορεί τον χρήστη και τον εμποδίζει να συνεχίσει την εκτέλεση της εφαρμογής.

Επίσης , αξιοποιώντας το στοιχείο “**Push Notifications**” επιτυγχάνεται η λήψη μηνυμάτων μέσω της πλατφόρμας OneSignal που προωθεί ο διαχειριστής της εφαρμογής προς τους χρήστες της εφαρμογής.

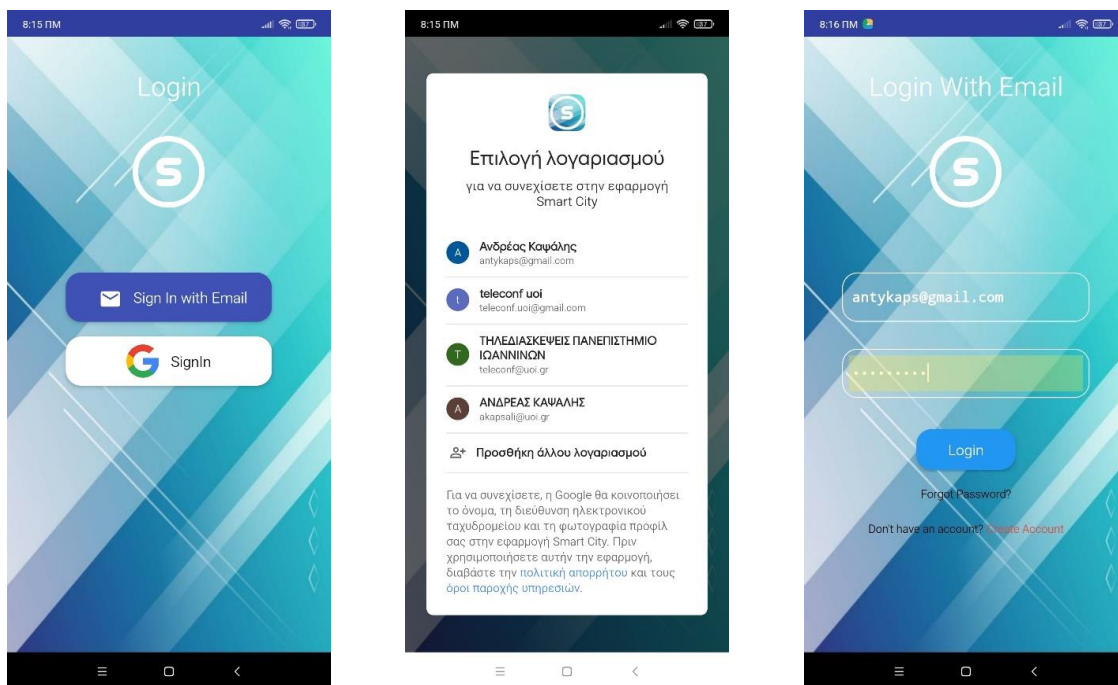
5.2.2. Login Screen



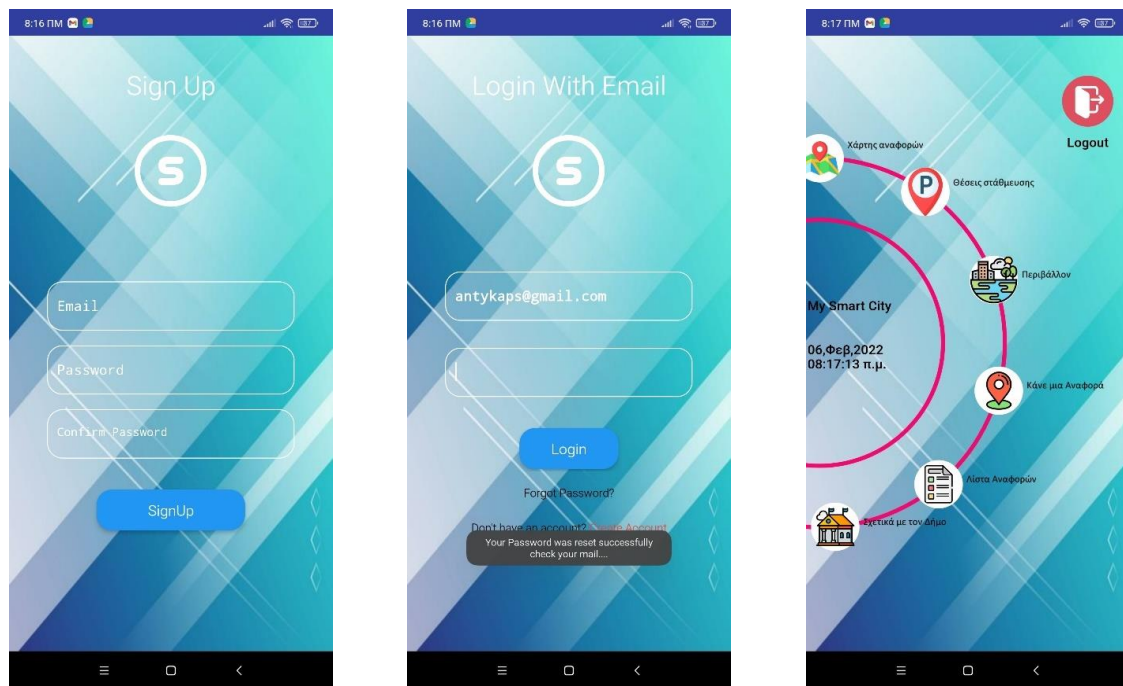
Εικόνα 41 Login Screen

Στην οθόνη “**Login**” γίνεται ο έλεγχος αυθεντικοποίησης , η εγγραφή του χρήστη και η απόκτηση λογαριασμού στην εφαρμογή καθώς και η δημιουργία- εμφάνιση του Menu επιλογών. Η οθόνη οργανώνεται σε 5 (πέντε) Vertical Arrangements τα οποία εμφανίζονται ή αποκρύπτονται μέσω του κώδικα της εφαρμογής και πραγματοποιούνται οι παρακάτω λειτουργίες της εφαρμογής:

- **Login_Page** (Είσοδος στην εφαρμογή με αυθεντικοποίηση μέσω Google Account)
- **Login_Email_Page** (Είσοδος στην εφαρμογή με την χρήση Email – Password)
- **Signup_Page** (Δημιουργία λογαριασμού στην εφαρμογή με την χρήση Email – Password - Αλλαγή Password)
- **Menu_VA** (Δημιουργία Μενού και «πέραςμα» της αντίστοιχης επιλογής από το μενού με την χρήση μεταβλητής στην οθόνη Report)
- **Offline_VA** (Εμφάνιση σχετικού μηνύματος αν διαπιστωθεί αδυναμία σύνδεσης στο Internet)

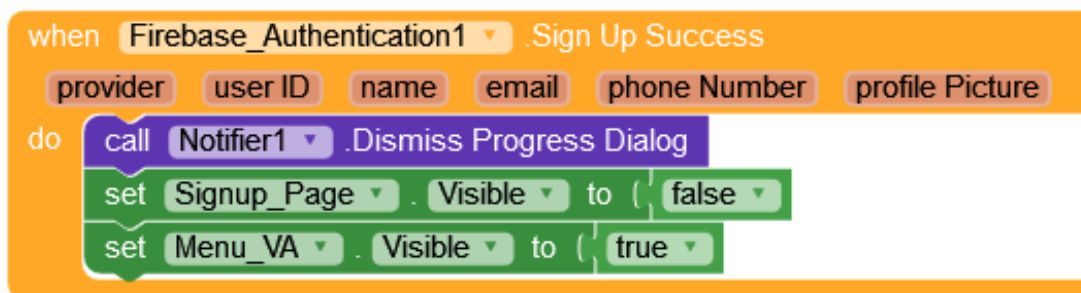


Εικόνα 42 Ενδεικτικές οθόνες Login Screen (a)



Εικόνα 43 Ενδεικτικές οθόνες Login Screen (b)

Βασικό στοιχείο στην οθόνη αυτή είναι το στοιχείο **“Firebase_Authentication”** με το οποίο γίνεται ο έλεγχος αν η χρήστης έχει αυθεντικοποιηθεί (Google Account ή Email -Password)

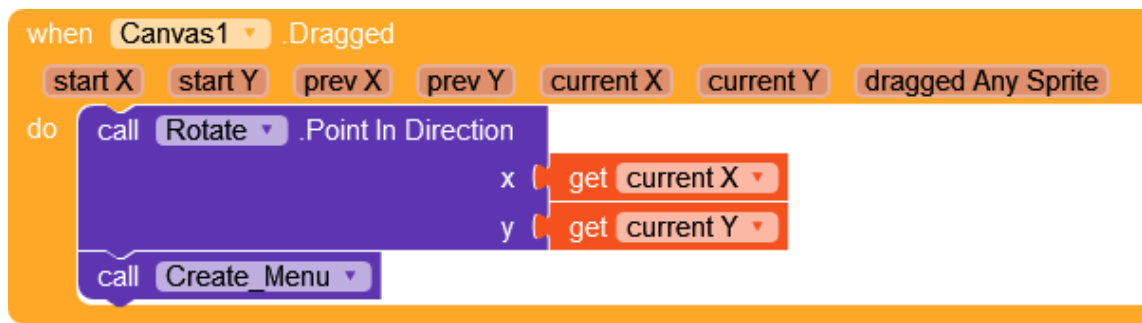


Εικόνα 44 Έλεγχος αυθεντικοποίησης

Επίσης για την δημιουργία αναφοράς είναι απαραίτητη η καταχώρηση θέσης με την χρήση της επέκτασης **“KIO4_GPS”** ανιχνεύεται αυτόματα αν δεν είναι ενεργοποιημένο το GPS και ενημερώνεται ο χρήστης να το ενεργοποιήσει .

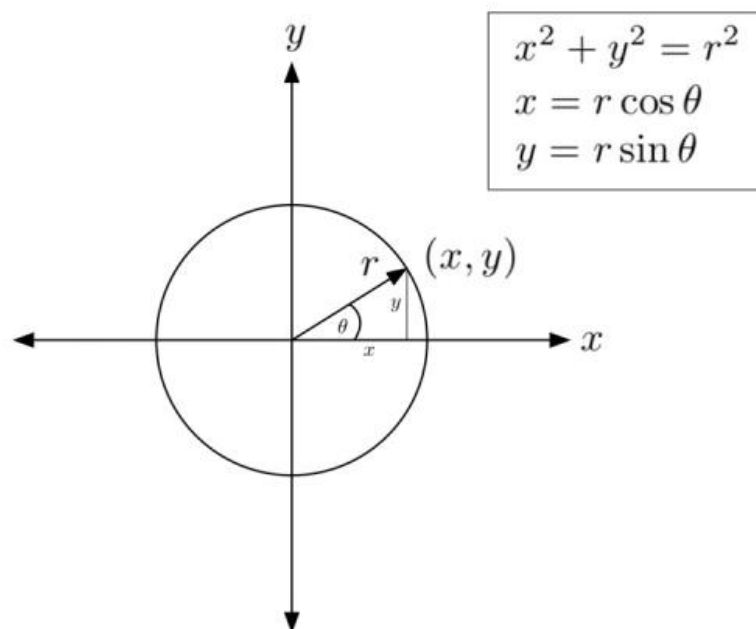
Η δημιουργία του μενού γίνεται με τα στοιχεία **“Canvas”** και **“Image Sprites”**

Η γωνία περιστροφής λαμβάνεται όταν κάνουμε **“drag”** πάνω στην οθόνη διαβάζοντας την γωνία περιστροφής (**Header**) του Image Sprite **“Rotate”**



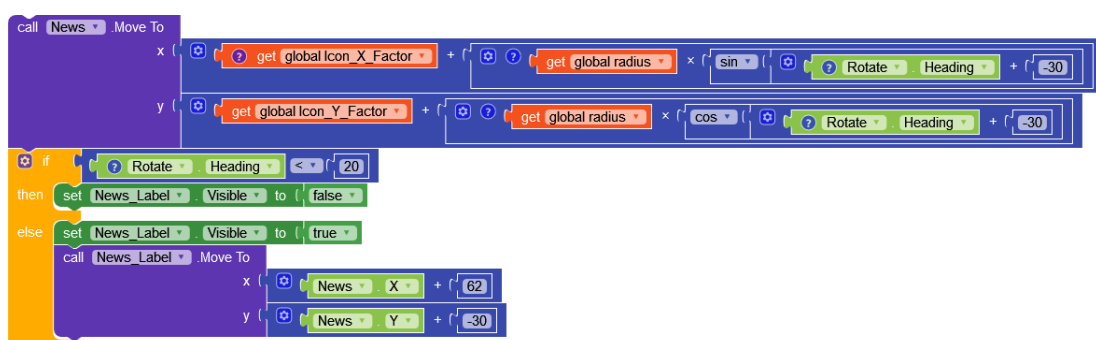
Εικόνα 45 Λήψη συντεταγμένων

Ο Υπολογισμός των σημείων X και Y πάνω στο Canvas βρίσκεται από την συνάρτηση $X = \text{radius} \cos \theta$ και $Y = \text{radius} \sin \theta$



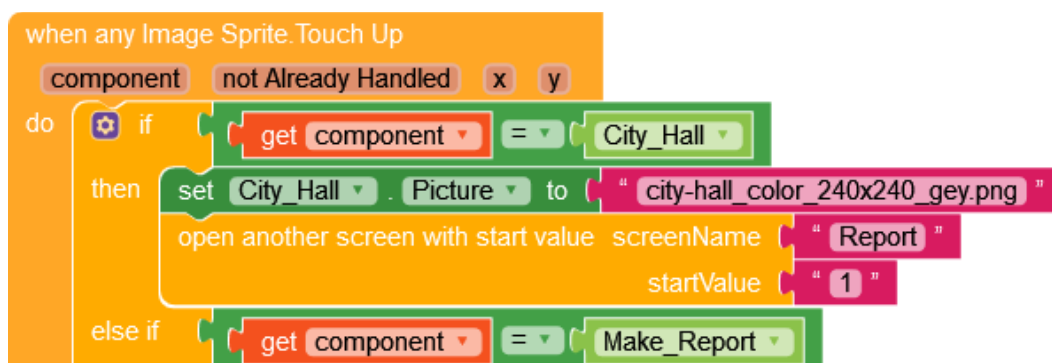
Εικόνα 46 Υπολογισμός της θέσης των Image Sprite

Πηγή : <https://thecuriousastronomer.wordpress.com/2014/09/15/derivation-of-centripetal-acceleration-using-polar-coordinates/>



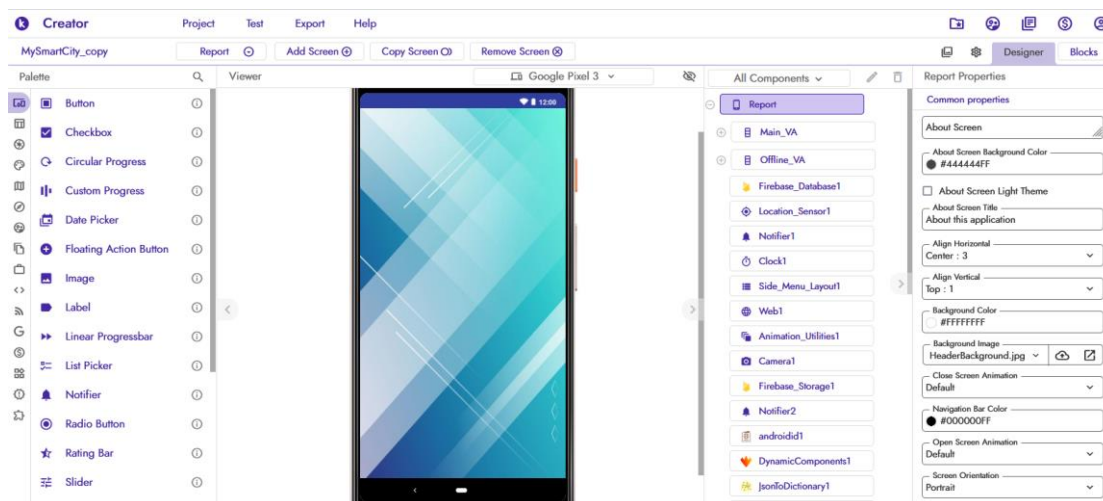
Εικόνα 47 Περιστροφή των στοιχείων του Μενού

Με την μετάβαση στην επόμενη οθόνη στέλνουμε και την σχετική μεταβλητή ανάλογα με την επιλογή που έχουμε κάνει στο μενού.



Εικόνα 48 Μετάβαση στη οθόνη Report

5.2.3. Report Screen

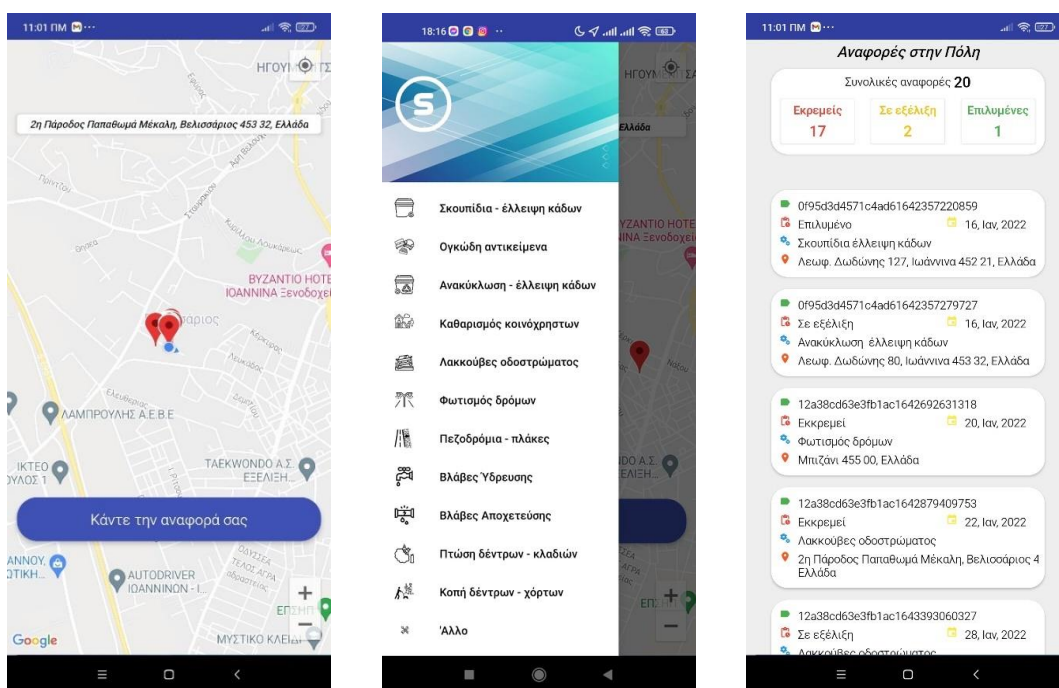


Εικόνα 49 Report Screen

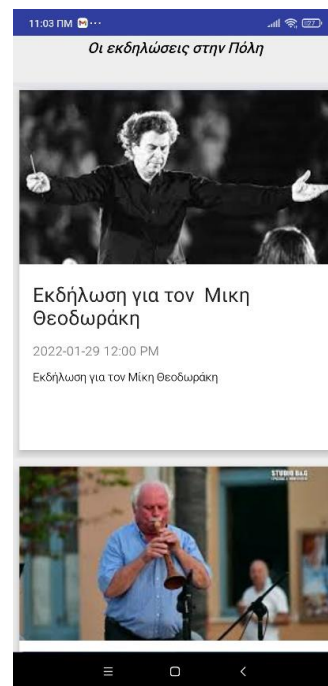
Στην οθόνη “**Report**” πραγματοποιούνται όλες οι λειτουργίες της εφαρμογής όπως η δημιουργία αναφορών με εμφάνιση των σημείων πάνω σε χάρτη, η εμφάνιση των περιβαλλοντικών συνθηκών, η εμφάνιση των διαθέσιμων θέσεων στάθμευσης, πληροφορίες για τον Δήμο, Νέα και Ανακοινώσεις καθώς και πληροφορίες για εκδηλώσεις.

Η οθόνη οργανώνεται σε 8 (οκτώ) προβολές Vertical Arrangements τα οποία εμφανίζονται ή κρύβονται μέσω του κώδικα της εφαρμογής και πραγματοποιούνται οι παρακάτω λειτουργίες της εφαρμογής:

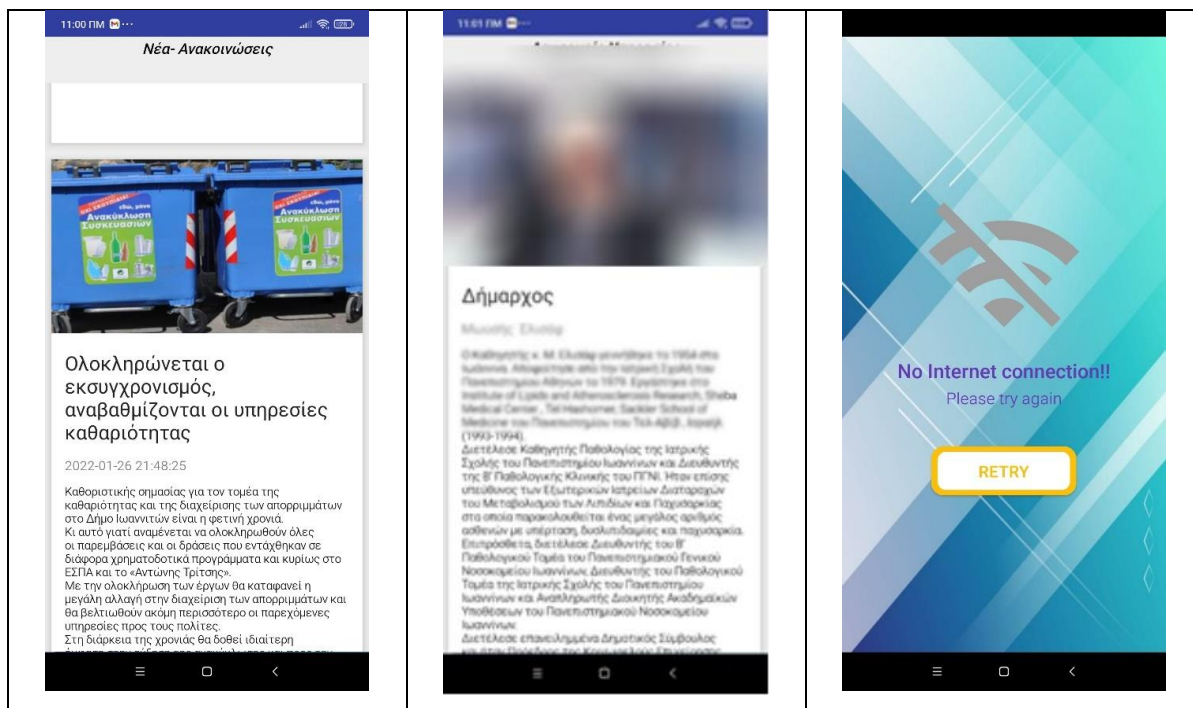
- **“Κάνε μια αναφορά»** Με αυτή την λειτουργία γίνεται η δημιουργία αναφορών και η εμφάνιση των σημείων πάνω σε χάρτη.
- **“Λίστα Αναφορών”** Με αυτή την λειτουργία εμφανίζεται η λίστα με τις αναφορές με φωτογραφία λεπτομέρειες για το πρόβλημα
- **“Περιβάλλον»** Με αυτή την λειτουργία εμφανίζονται οι περιβαλλοντικές συνθήκες όπως θερμοκρασία και υγρασία στον Δήμο σε πραγματικό χρόνο.
- **“Parking»** Με αυτή την λειτουργία εμφανίζονται οι διαθέσιμες θέσεις στάθμευσης στον Δήμο.
- **“Εκδηλώσεις”** Με αυτή την λειτουργία εμφανίζονται οι προγραμματισμένες εκδηλώσεις στον Δήμο.
- **“Νέα-Ανακοινώσεις”** Με αυτή την λειτουργία εμφανίζονται οι Νέα και Ανακοινώσεις του Δήμου που ενδιαφέρουν τους πολίτες.
- **“Σχετικά με τον Δήμο”** Με αυτή την λειτουργία εμφανίζονται πληροφορίες για τον Δήμο και το Δημοτικό συμβούλιο
- **“Offline_VA”** Με αυτή την λειτουργία έχουμε εμφάνιση σχετικού μηνύματος αν διαπιστωθεί αδυναμία σύνδεσης στο Internet.



Εικόνα 50 Ενδεικτικές οθόνες (α)



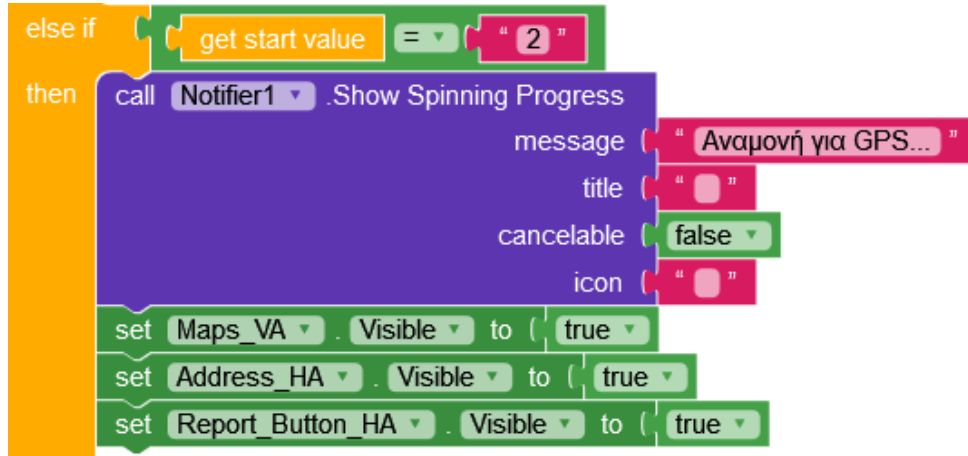
Εικόνα 51 Ενδεικτικές οθόνες (b)



Εικόνα 52 Ενδεικτικές οθόνες (c)

5.2.3.1. Κάνε μια αναφορά

Για την δημιουργία αναφοράς του προβλήματος (επιλογή 2 στο Menu,)



Εικόνα 53 Επιλογή «Κάνε μια Αναφορά»

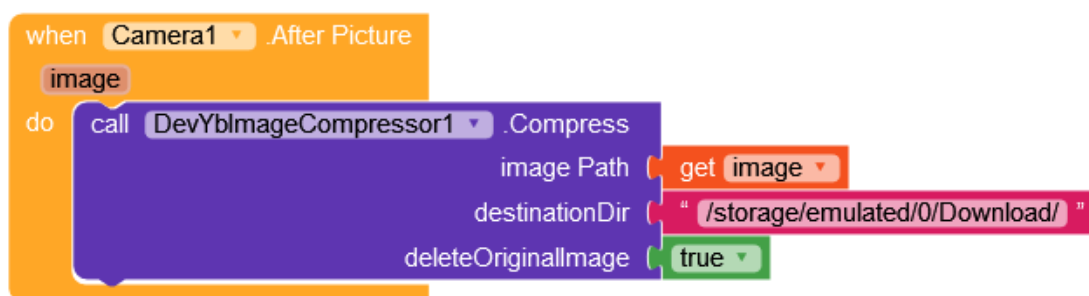
πραγματοποιείται η εμφάνιση του χάρτη (Maps_VA) , η εμφάνιση του κουμπιού καταχώρησης (Report_Button_HA) καθώς και η εμφάνιση της διεύθυνσης (Address_HA). Πατώντας το κουμπί «**Κάντε την αναφορά σας**» (**Report_Button_HA**) ο χρήστης μπορεί να επιλέξει από το πλάγιο μενού που εμφανίζεται μια από τους 12 συχνοί τύπους προβλημάτων

- Σκουπίδια έλλειψη κάδων
- Ογκώδη αντικείμενα
- Ανακύκλωση έλλειψη κάδων
- Καθαρισμός κοινόχρηστων χώρων
- Λακκούβες οδοστρώματος
- Φωτισμός δρόμων
- Πεζοδρόμια πλάκες
- Βλάβες Ύδρευσης
- Πτώση δέντρων κλαδιών
- Κοπή δέντρων – χόρτων
- Άλλο

Σε περίπτωση που ο τύπος του προβλήματος δεν αναφέρεται ο χρήστης μπορεί να επιλέξει «Άλλο» και στην συνέχεια να περιγράψει το πρόβλημα σε πλαίσιο που του εμφανίζεται.

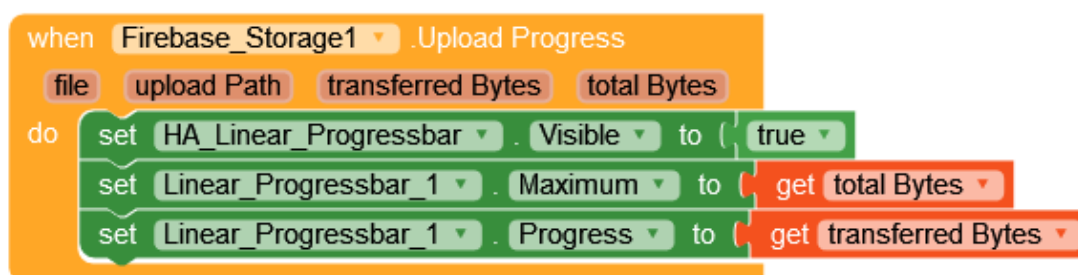
Πατώντας **Αποθήκευση** δίνεται η δυνατότητα στο χρήστη να αποστείλει μια φωνογραφία του προβλήματος είτε εκείνη την στιγμή από την **camera** είτε από το **αρχείο φωτογραφιών**.

Δεδομένου ότι πλέον οι κάμερες των κινητών τηλεφώνων είναι υψηλής ευκρινείας και παράγουν εικόνες μεγάλου σχετικά μεγέθους(5-10MB) και δεδομένου ότι η σύνδεση στον Internet συνήθως γίνεται μέσω του παρόχου της κινητής τηλεφωνίας για λόγους οικονομίας αλλά και για λόγους ταχύτητας χρησιμοποιήθηκε η συνάρτηση “DevYbImageCompressor.Compress” από το πρόσθετο “DevYbImageCompressor” που συμπιέζει την φωτογραφία περίπου στο 1/10 χωρίς η φωτογραφία να χάνει σε ποιότητα που να επηρεάζει την εφαρμογή.



Εικόνα 54 Συμπίεση φωτογραφίας

Με την επιβεβαίωση της ορθότητας των στοιχείων η φωτογραφία γίνεται upload στο Storage της Firebase.

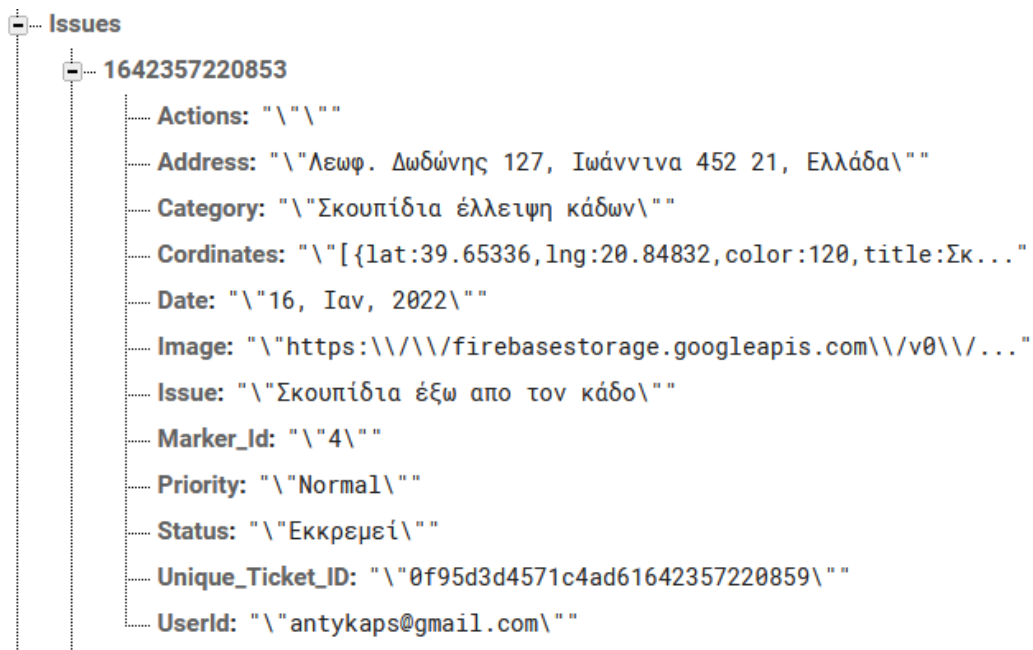


Εικόνα 55 Μεταφόρτωση στο Storage

και στην συνέχεια αποθηκεύονται όλα τα στοιχεία της αναφοράς στην Realtime Database τα στοιχεία της αναφοράς (Tag – Value).

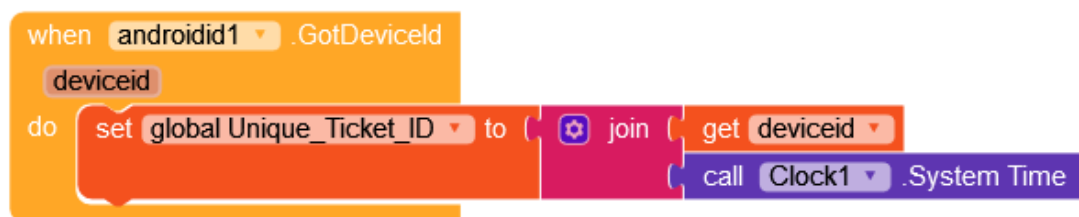


Εικόνα 56 Αποθήκευση στην Realtime Database



Εικόνα 57 Δομή Realtime Database

Κάθε αναφορά λαμβάνει ένα μοναδικό **Unique_Ticket_ID** που δημιουργείται από την μοναδική Mac Address της συσκευής με την χρήση της συνάρτησης “**GetDeviceId**” της επέκτασης “**androidId**” και το Timestamp της χρονικής στιγμής που γίνεται η αναφορά.

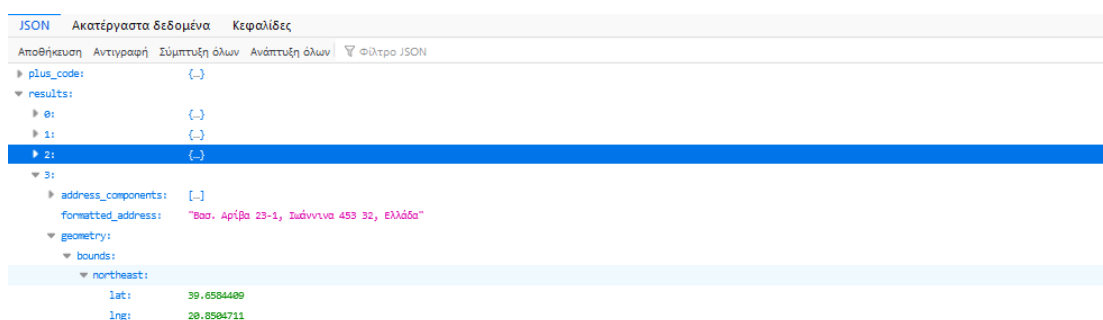


Εικόνα 58 Δημιουργία Unique ID

Η διεύθυνση που εμφανίζεται λαμβάνεται χρησιμοποιώντας το Geocoding API που είναι μια υπηρεσία του Google Maps Platform που μας παρέχει γεωκωδικοποίηση (Geocoding) και αντίστροφη γεωκωδικοποίηση (Reverse Geocoding) διευθύνσεων. Η αντίστροφη γεωκωδικοποίηση είναι η διαδικασία μετατροπής των γεωγραφικών συντεταγμένων σε μια διεύθυνση αναγνώσιμη από τον άνθρωπο. Η πρόσβαση στο Geocoding API γίνεται μέσω μιας διεπαφής HTTP Έτσι για να κάνουμε το Reverse Geocoding και να λάβουμε την διεύθυνση έχοντας γνωστές τις γεωγραφικές συντεταγμένες (Longitude , Latitude) του σημείου που βρισκόμαστε , καλούμε το URL

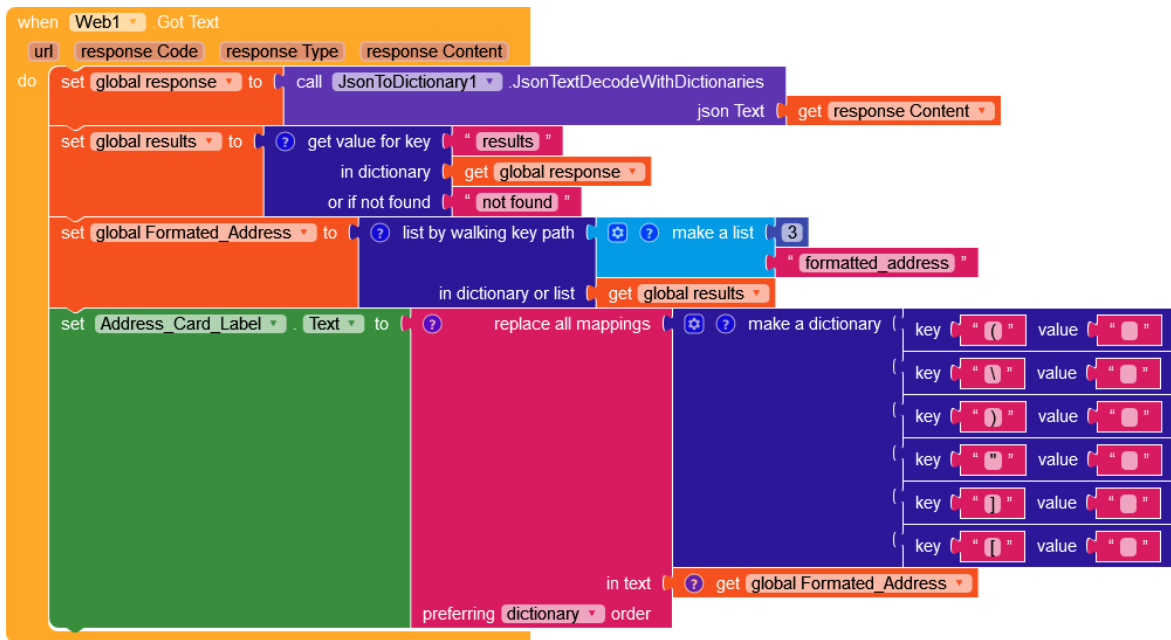
https://maps.googleapis.com/maps/api/geocode/json?latlng=40.714224,-73.961452&key=YOUR_API_KEY (*API_KEY έχει δημιουργηθεί με την ενεργοποίηση του Google Maps API).

το οποίο μας επιστρέφει ένα αντικείμενο JSON το οποίο περιέχει την διεύθυνση



Εικόνα 59 Address Json object

Ενώ η διεύθυνση λαμβάνετε με την χρήση της συνάρτησης **JsonToDicrionary** του στοιχείου **Dictionaries** του Kodular.



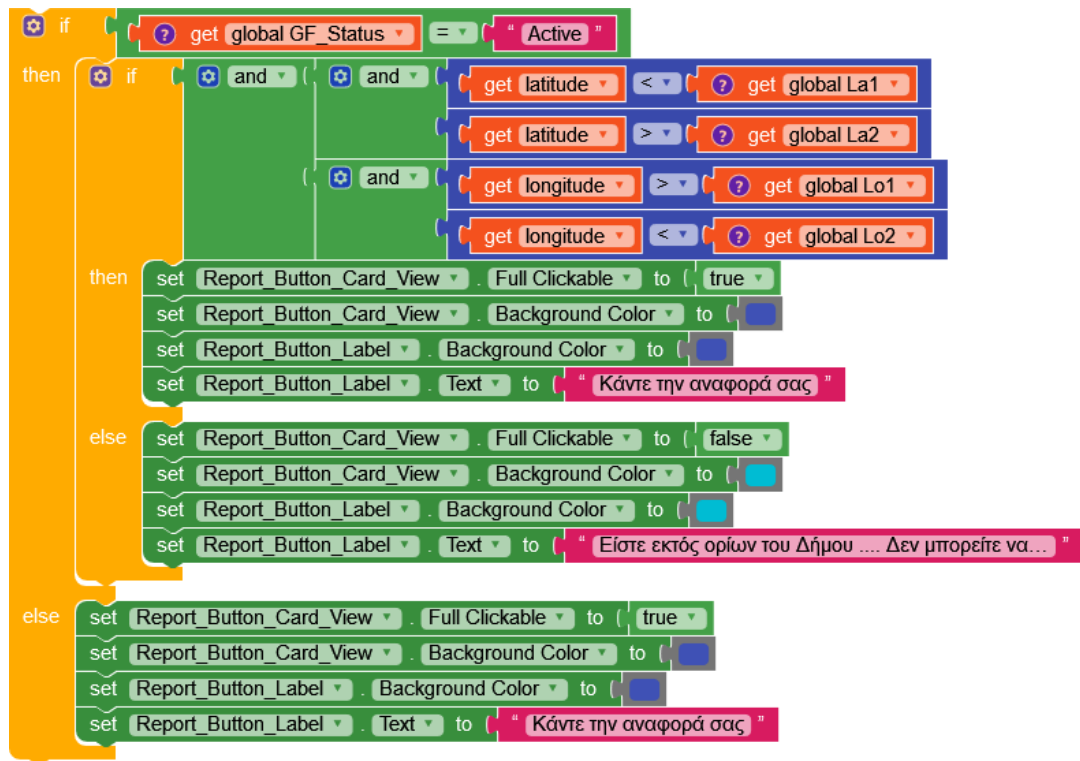
Εικόνα 60 Αποκωδικοποίηση διεύθυνσης

Κάθε αναφορά προβλήματος εμφανίζεται στον χάρτη με ένα σημάδι “marker”. Τα στοιχεία του marker είναι οι συντεταγμένες (Lat, Lot) , το χρώμα (κόκκινο για τα εκκρεμή , κίτρινο για τα υπό επίλυση και πράσινο για τα επιλυμένα) , τον τίτλο και το Snippet (πρόσθετο κείμενο που εμφανίζεται κάτω από τον τίτλο). Η καταχώρηση του marker γίνεται σε json format ώστε να είναι αργότερα εύκολη η ανάγνωση και η τοποθέτηση στον χάρτη.



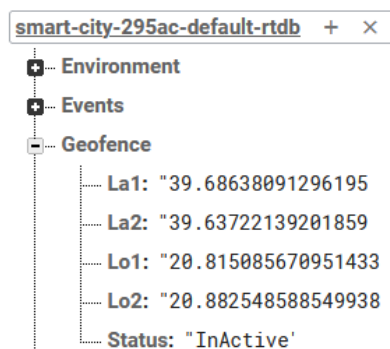
Εικόνα 61 Δημιουργία Marker

Για να αποφύγουμε αναφορές εκτός Δήμου χρησιμοποιούμε την τεχνική **Geofencing**. Αυτό επιτυγχάνεται εξετάζοντας αν οι γεωγραφικές συντεταγμένες του σημείου που βρισκόμαστε είναι εντός ενός ορθογωνίου σχήματος που είναι τα γεωγραφικά όρια του Δήμου. Επειδή όμως η χωροταξική διάταξη του Δήμου δεν έχει ορθογώνιο σχήμα μπορούμε να την «τεμαχίσουμε» σε μικρότερα ορθογώνια σχήματα και να κά-
νουμε διαδοχικούς ελέγχους σε όλα τα ορθογώνια που σχηματίστηκαν.



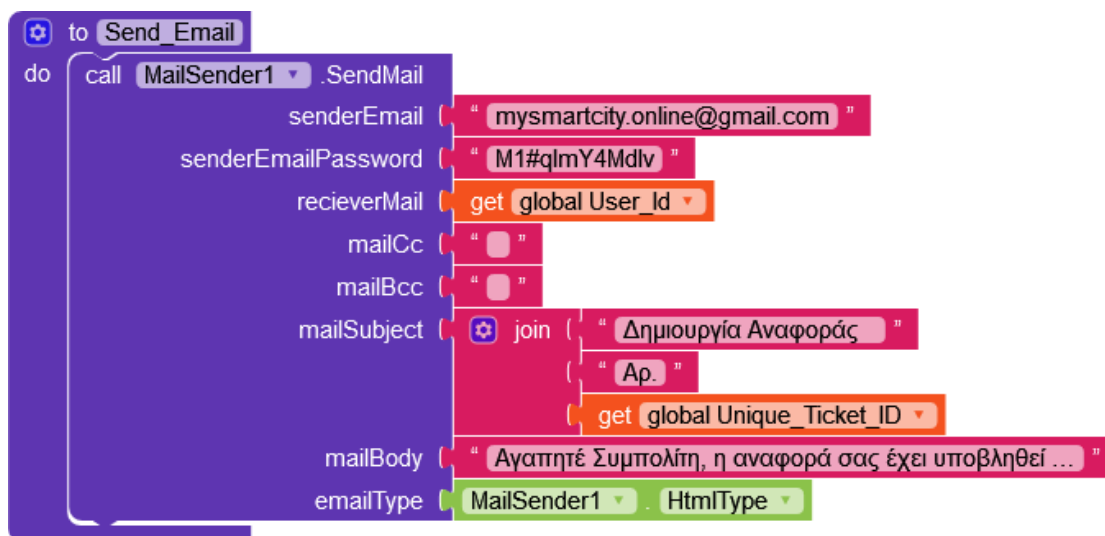
Εικόνα 62 Έλεγχος Γεωγραφικά ορίων Δήμου (Geofence)

Οι συντεταγμένες – όρια του Δήμου είναι αποθηκευμένες στην Firebase



Εικόνα 63 Γεωγραφικά όρια Δήμου

Η εφαρμογή μετά την αποθήκευση της αναφοράς στην Firebase η εφαρμογή στέλνει email επιβεβαίωσης στον χρήστη.



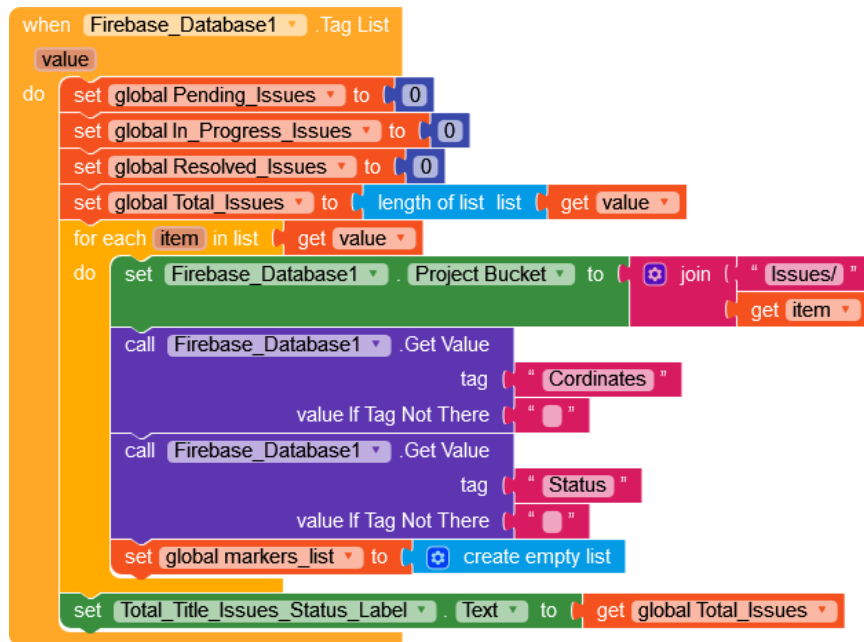
Εικόνα 64 Αποστολή Email

Η εμφάνιση των όλων των marker στον χάρτη γίνεται κάνοντας κλήση της συνάρτησης GetTagList της Firebase.



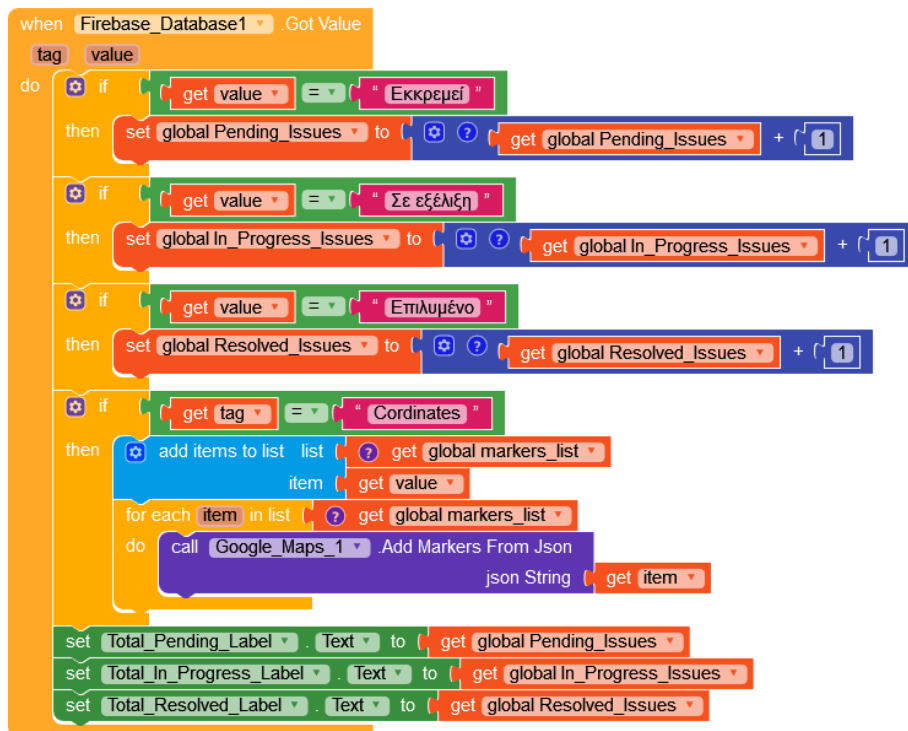
Εικόνα 65 Λήψη των Tags από την Realtime Database

Διαβάζουμε τις τιμές για το Tag που θέλουμε (Coordinates)



Εικόνα 66 Λήψη των συντεταγμένων των Marker

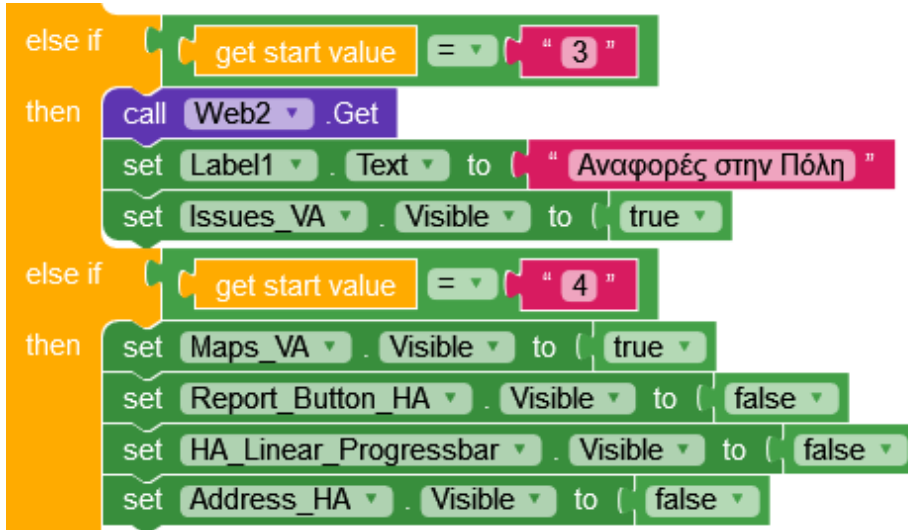
Και αφού λάβουμε τις τιμές οι marker σχηματίζονται από την συνάρτηση “Add Markers From Json” του στοιχείου Google Maps.



Εικόνα 67 Δημιουργία των Marker στον χάρτη

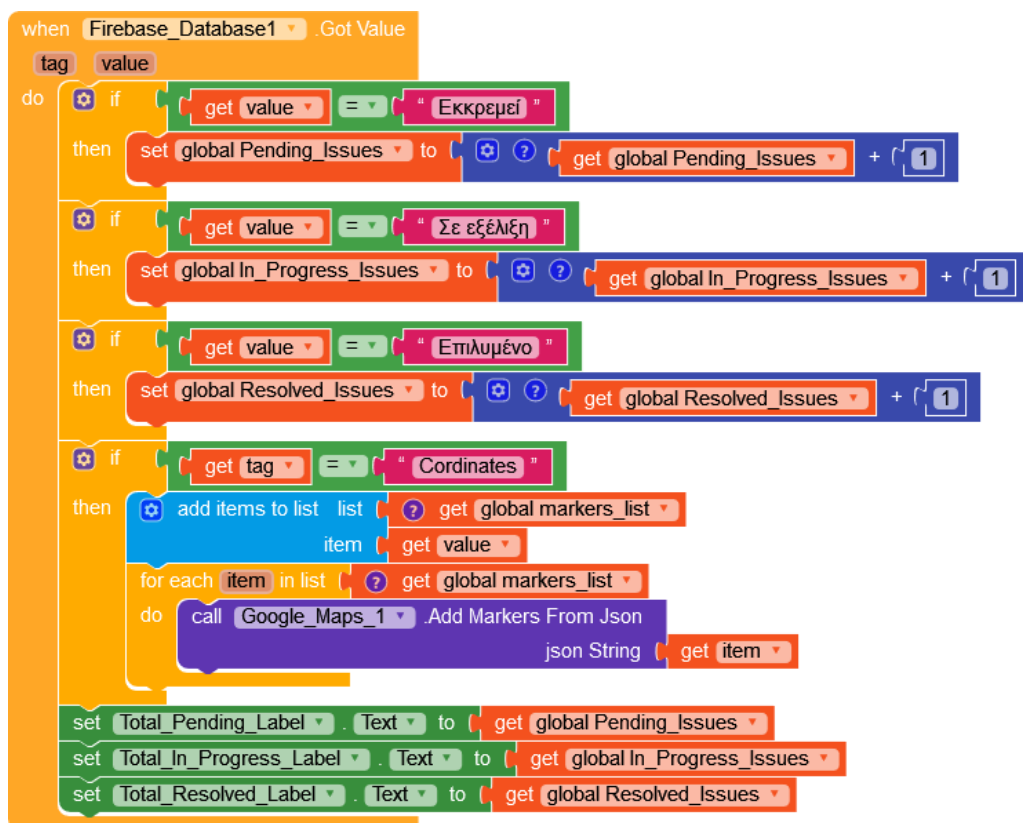
5.2.3.2. Λίστα Αναφορών

Για την προβολή της λίστας αναφορών (επιλογή 3 στο Menu,) γίνεται εμφάνιση του Vertical Arrangement (Issues_VA)



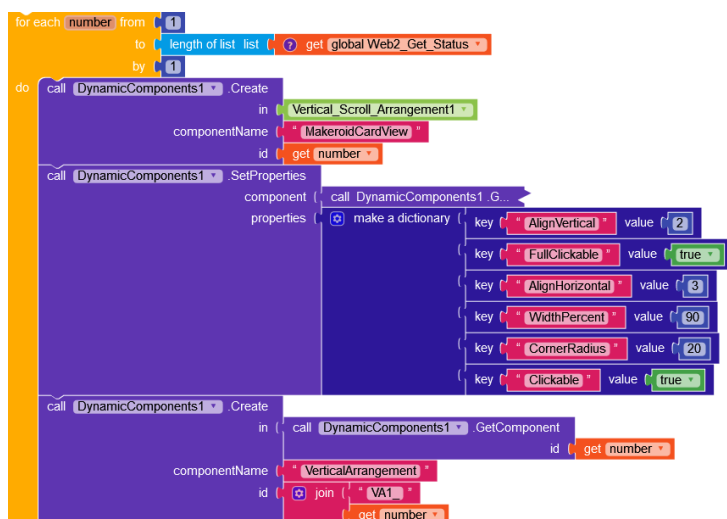
Εικόνα 68 Επιλογή «Λίστα αναφορών»

Ο Υπολογισμός των αναφορών γίνεται από την κλήση **"GetTagList"** και αφού λάβουμε τις τιμές αθροίζουμε τις αναφορές ανά κατηγορία.



Εικόνα 69 Υπολογισμός των Marker ανά κατηγορία

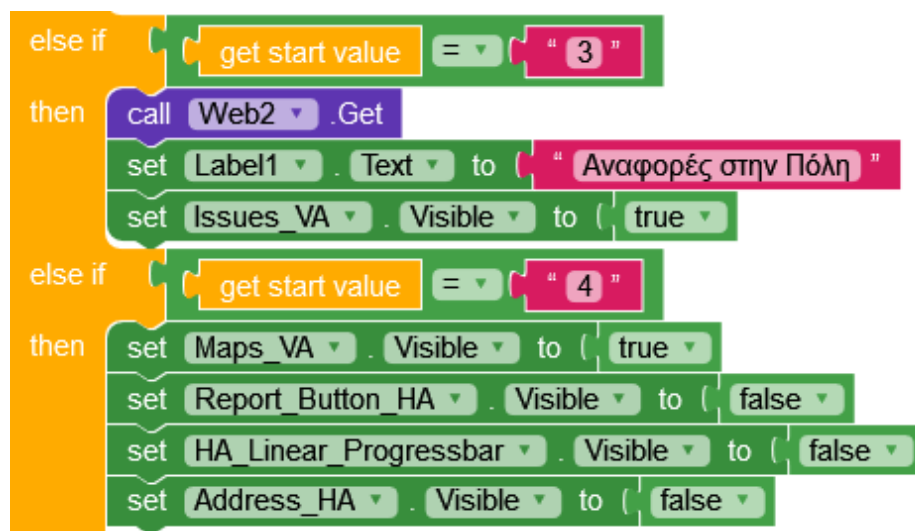
Οι αναφορές εμφανίζονται υπό μορφή κάρτας τα στοιχεία της οποίας σχηματίζονται δυναμικά με την χρήση του πρόσθετου “**DynamicComponents**” με αποτέλεσμα να έχουμε δημιουργήσει μια κάρτα με τα στοιχεία (ετικέτες, κείμενο , εικόνες κλπ.) που επιθυμούμε και σε όποια θέση θέλουμε.



Εικόνα 70 Δυναμική δημιουργία καρτών αναφοράς

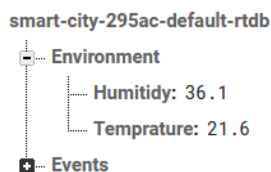
5.2.3.3. Περιβάλλον

Η εμφάνιση των περιβαλλοντικών συνθηκών (επιλογή 6 στο Menu,) γίνεται μέσα από Vertical Arrangement “Enviroment_VA”



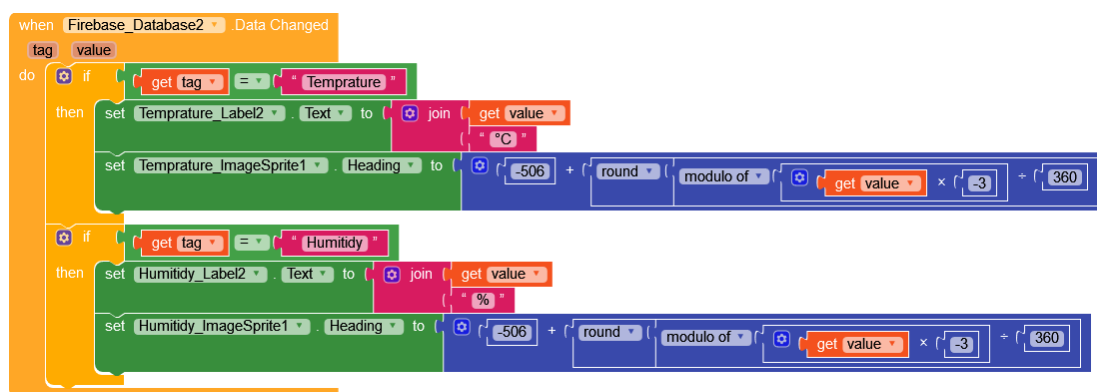
Εικόνα 71 Επιλογή «Περιβάλλον»

Την θερμοκρασία και η υγρασία την στέλνει στην Firebase ένα κύκλωμα μικροελεγκτή ESP8266 με αισθητήρα θερμοκρασίας -υγρασίας DHT22



Εικόνα 72 Θερμοκρασία και Υγρασία στην Realtime Database

Για κάθε αλλαγή στα Values των Tags, Humidity και Temperature αυτόματα ενημερώνονται και οι τιμές θερμοκρασίας και υγρασίας στην προβολή αυτή.

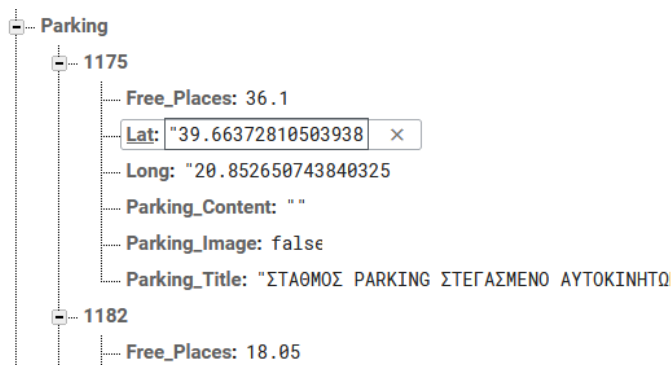


Εικόνα 73 Προβολή της Θερμοκρασίας και Υγρασίας

5.2.3.4. Parking

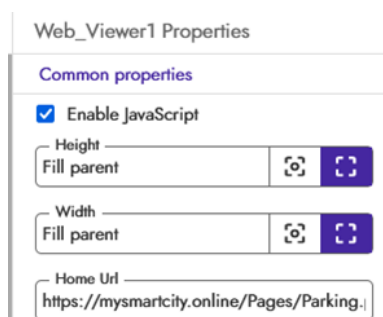
Η εμφάνιση των διαθέσιμων σημείων καθώς και των διαθέσιμων θέσεων στάθμευσης (επιλογή 7 στο Menu,) γίνεται μέσα από Vertical Arrangement “Parking_Map_VA”.

Τους διαθέσιμους σταθμούς στάθμευσής καθώς και οι διαθέσιμες θέσεις τις λαμβάνει από την Firebase. Οι θέσεις σε πραγματικό λαμβάνονται από ένα κύκλωμα μικροελεγκτή ESP8266 που ενημερώνει την Firebase.



Εικόνα 74 Στοιχεία Parking στην Realtime Database

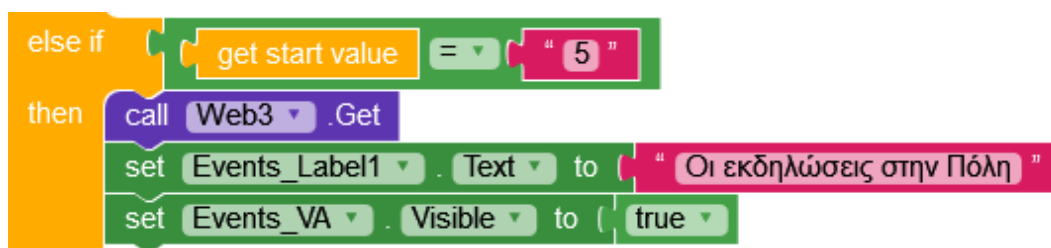
Ο χάρτης «φορτώνεται» από το στοιχείο Web_Viewer από σελίδα που υπάρχει στην ιστοσελίδα της εφαρμογής <https://mysmartcity.online/Pages/Parking.html>



Εικόνα 75 Προβολή χάρτη Parking

5.2.3.5. Εκδηλώσεις

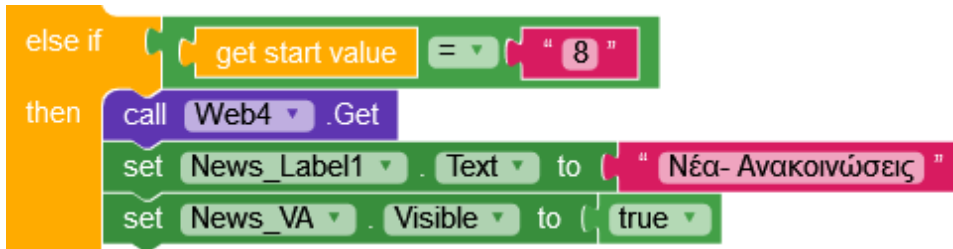
Η εμφάνιση των Εκδηλώσεων (επιλογή 5 στο Menu,) γίνεται μέσα από Vertical Arrangement «Events_VA»



Εικόνα 76 Επιλογή «Εκδηλώσεις»

5.2.3.6. Νέα-Ανακοινώσεις

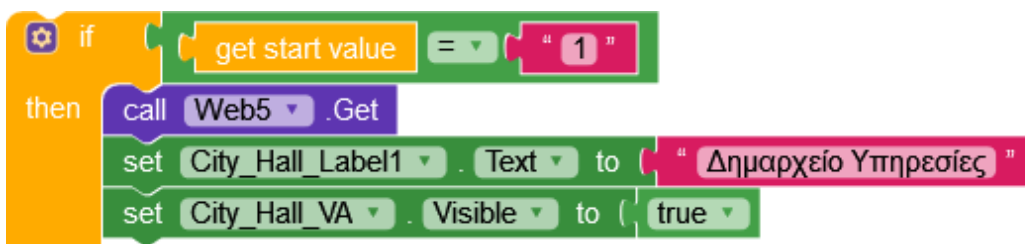
Η εμφάνιση των Νέων-Ανακοινώσεων (επιλογή 8 στο Menu,) γίνεται μέσα από Vertical Arrangement «News_VA»



Εικόνα 77 Επιλογή «Νέα-Ανακοινώσεις»

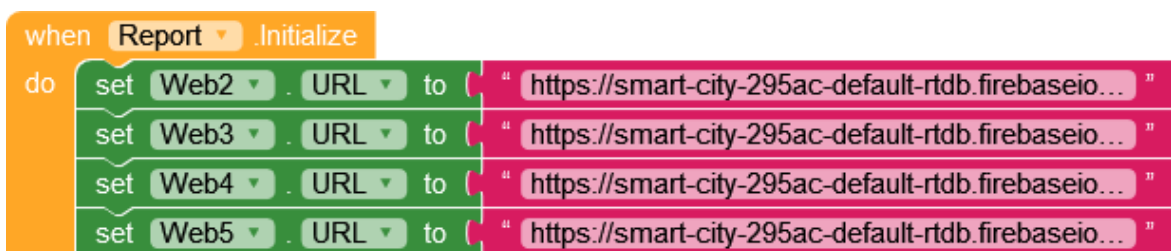
5.2.3.7. Σχετικά με τον Δήμο

Η εμφάνιση του «Σχετικά με τον Δήμο» (επιλογή 1 στο Menu,) γίνεται μέσα από Vertical Arrangement «City_Hall_VA»



Εικόνα 78 Επιλογή «Σχετικά με τον Δήμο»

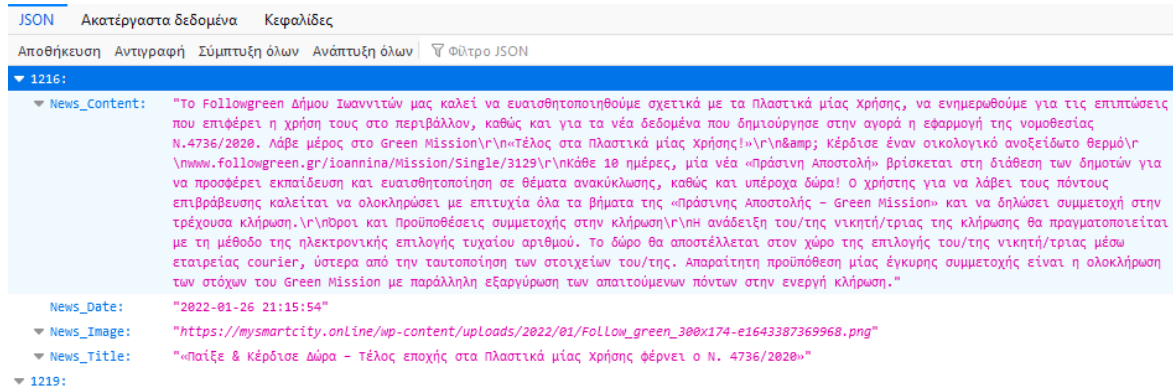
Για την εμφάνιση των «Εκδηλώσεων», την «Νέων» και του «Σχετικά με το Δήμο» χρησιμοποιείται το στοιχείο Web του Kodular



Εικόνα 79 Web Κλήση

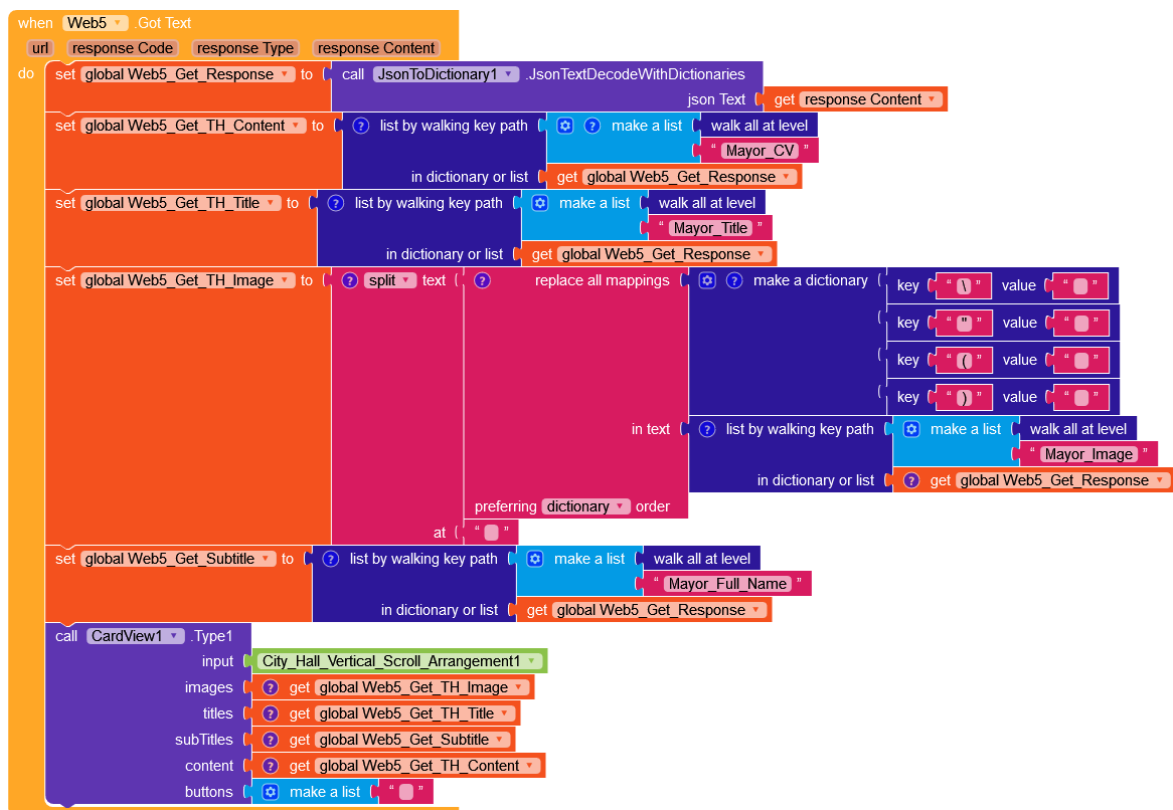
Είναι ένα μη ορατό στοιχείο που παρέχει λειτουργίες για αιτήματα HTTP GET, POST, PUT και DELETE. το οποίο μας επιστρέφει δεδομένα σε μορφή Json

<https://smart-city-295ac-default-rtdb.firebaseio.com/News/.json>



Εικόνα 80 Json Object

Αξιοποιώντας της συνάρτηση **JsonToDicrionary** του στοιχείου **Dictionaries** του Kodular και με το πρόσθετο **“CardView”** εμφανίζονται τα δεδομένα με την μορφής καρτέλας.



Εικόνα 81 Δυναμική δημιουργία κάρτας «Δήμος -Δημαρχείο»

Τα δεδομένα αντλούνται από τα σχετικά Tags από την Realtime Database της Firebase. Η αποθήκευση των δεδομένων γίνεται με την χρήση JavaScript κώδικα ταυτόχρονα με την δημιουργία αντίστοιχων post στην ιστοσελίδα της εφαρμογής.



Εικόνα 82 Στοιχεία Δήμος -Δημαρχείο στην Realtime Database

6. Ανάλυση και σχεδίαση της Web εφαρμογής “mysmartcity.online”

Η Web εφαρμογή είναι σχεδιασμένη έτσι ώστε οι χρήστες της (επισκέπτες) να μπορούν να βλέπουν τα προβλήματα που έχουν αναφερθεί με marker πάνω στον χάρτη , να βλέπουν την λίστα με τις τελευταίες αναφορές , τους σταθμούς στάθμευσης καθώς και τις διαθέσιμες θέσεις , τα Νέα , τις Ανακοινώσεις καθώς και εκδηλώσεις του Δήμου.

Οι διαχειριστές της εφαρμογής μπορούν να επεξεργαστούν τις αναφορές που έχουν υποβληθεί , να προσθέσουν ενέργειες και να αλλάξουν την κατάσταση τους από εκκρεμείς σε υπό εξεργασία ή ως επιλυμένες ανάλογα με την πρόοδο των εργασιών. Επίσης να δημιουργήσουν άρθρα (Post) για κάθε κατηγορίας (Νέα – Εκδηλώσεις-Δήμος) τα οποία με JavaScript κώδικα να ενημερώνουν την Firebase και από εκεί να τα διαβάζει η mobile εφαρμογή.

6.1. Εργαλεία ανάπτυξης της web εφαρμογής

Για την υλοποίηση της ιστοσελίδας της εφαρμογής της “Εξυπνης Πόλης αποκτήθηκε το Domain Name **mysmartcity.online** καθώς ο σχετικός χώρος φιλοξενίας. Το πακέτο φιλοξενίας που επιλέξαμε είναι βασισμένο σε Web Server NGINX , με βάση δεδομένων Mysql και την υποστήριξη της php. Για την ασφάλεια της ιστοσελίδας έχουμε ενεργοποιήσει το πρωτόκολλο ασφάλειας SSL (https) ώστε οι χρήστες να είναι ήσυχοι πως η ιστοσελίδα μας ανήκει και είναι νόμιμη και τα δεδομένα τους δε θα υποκλαπούν. Για την δημιουργία της ιστοσελίδας εγκαταστήσαμε ένα από τα πιο δημοφιλή σύστημα διαχείρισης περιεχομένου (cms) το **WordPress** .

Χρησιμοποιήθηκαν τα Plugin «**User Role Editor**» , «**Restrict Content**» και «**Nav Roles Menu**» , « **Custom Field Suite**». Το Theme που χρησιμοποιήθηκε είναι το «**Astra**» στην δωρεάν έκδοση του . Για να μην επηρεάζονται προσωπικές αλλαγές και τροποποιήσεις σε τυχόν αναβαθμίσεις του theme χρησιμοποιήθηκε η τεχνική του Child Theme. Το Astra είναι ένα «responsive» Theme ,δηλαδή προσαρμόζει το πλάτος του, καθώς και τα στοιχεία που περιέχει (εικόνες, γραφήματα, κείμενο κλπ.) ανάλογα με την συσκευή και φυσικά υποστηρίζει συσκευές όπως κινητά τηλέφωνα και tablet.

Για την αισθητική και εργονομική βελτίωση της ιστοσελίδας χρησιμοποιήθηκε το plugin Page Builder «**Elementor Pro**»

Για την σύνδεση της εφαρμογής με την Firebase χρησιμοποιήθηκαν τα παρακάτω εργαλεία προγραμματισμού Html , JavaScript ,Bootstrap , jQuery , CSS

Η **Html** είναι μία γλώσσα, δεν θα λέγαμε απόλυτα προγραμματισμού μιας και δεν περιέχει τις βασικές συνθήκες ελέγχων, επαναλήψεων, κλπ. Με αυτήν, φτιάχνουμε στην ουσία μία ιστοσελίδα και με τις εντολές της, τοποθετούμε όλα τα αντικείμενα (εικόνες, κείμενα, links, κ.λπ.) στο μέρος που θέλουμε, τα μορφοποιούμε και τα προβάλλουμε σε αυτήν με τον τρόπο που εμείς έχουμε "προγραμματίσει" με αυτήν.

Η **JavaScript** είναι μια γλώσσα προγραμματισμού που βασίζεται στο κείμενο και χρησιμοποιείται τόσο στην πλευρά του πελάτη όσο και στον διακομιστή, και σας επιτρέπει να κάνετε τις σελίδες Web πιο διαδραστικές. Η JavaScript προσδίδει στις σελίδες Web αλληλεπιδραστικά στοιχεία που δεσμεύουν έναν χρήστη.

Η **Bootstrap** είναι ένα ανοιχτού κώδικα front-end framework που παρέχει έτοιμα στοιχεία HTML, κώδικας CSS και μέθοδοι JS για τη δημιουργία responsive ιστοσελίδων . Διαθέτει πλήρως σχεδιασμένα στοιχεία διεπαφής (Πίνακες, Κουμπιά, Φόρμες, Λίστες, Μενού πλοήγηση κ.λπ.).

jQuery είναι μια βιβλιοθήκη επέκτασης JavaScript για την διαχείριση στοιχείων html, για χρήση animation , για χειρισμό συμβάντων και επικοινωνίας μέσω AJAX. Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι:

<https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js>

<https://www.gstatic.com/firebasejs/7.14.6/firebase-app.js>

<https://www.gstatic.com/firebasejs/7.14.6/firebase-analytics.js>

<https://www.gstatic.com/firebasejs/7.14.6/firebase-database.js>

<https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js>

<https://www.gstatic.com/firebasejs/4.12.1/firebase.js>

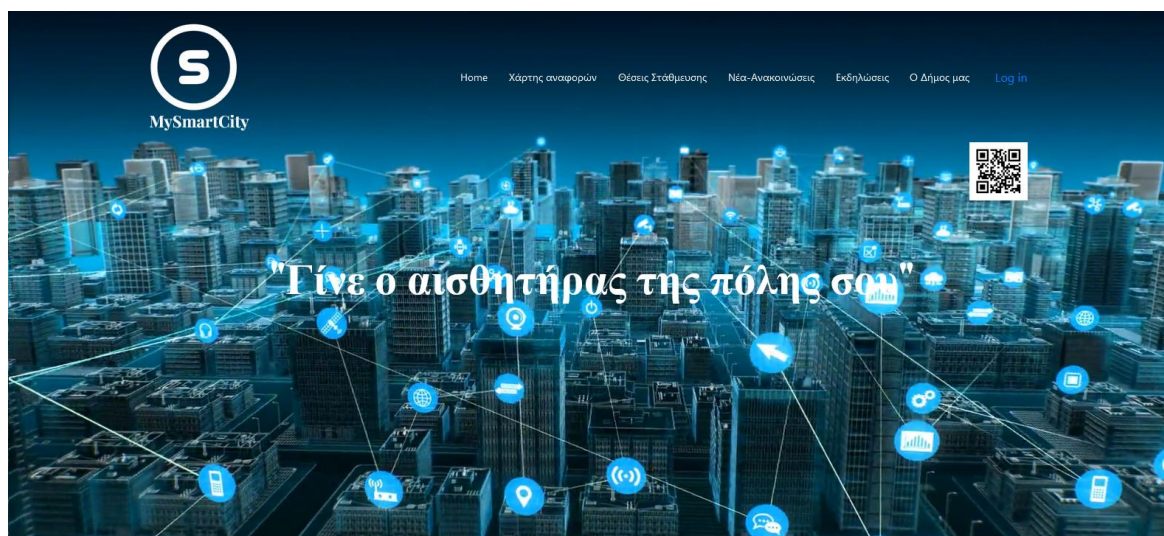
Η **CSS** (Cascading Style Sheets – διαδοχικά φύλλα ύφους ή επάλληλα φύλλα ύφους) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου.

(<https://el.wikipedia.org/wiki/CSS>).

6.2. Υλοποίηση της web εφαρμογής

6.2.1. Home

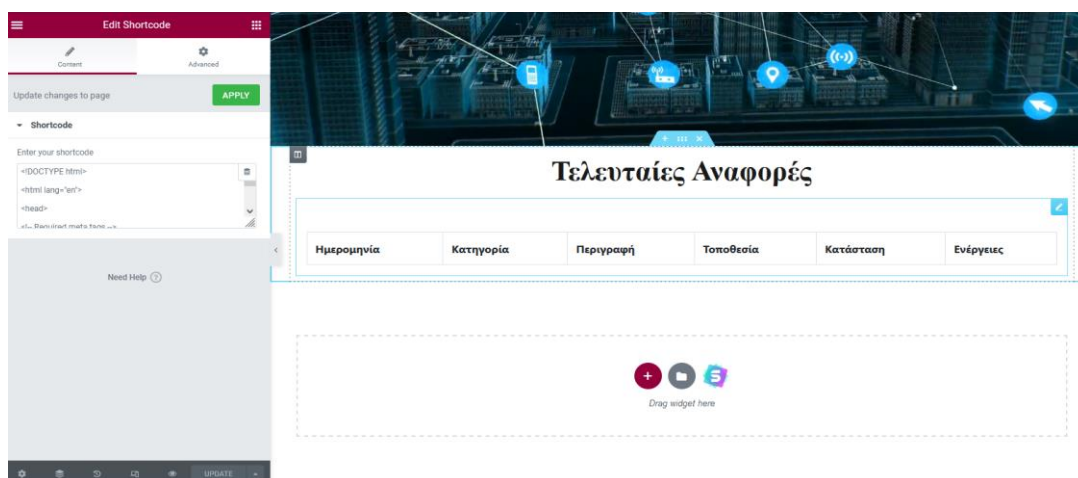
Στην αρχική σελίδα «Home» εμφανίζεται ένα εισαγωγικό video . Σε αυτή την σελίδα εμφανίζεται ένα QrCode μέσω του οποίου μπορεί να γίνει η λήψη της Mobile εφαρμογής.



Τελευταίες Αναφορές

Ημερομηνία	Κατηγορία	Περιγραφή	Τοποθεσία	Κατάσταση	Ενέργειες
3, Φεβ, 2022	Λακκούβες οδοστρώματος		2η Πάροδος Παπαθωμά Μέκαλη, Βελισσάριος 453 32, Ελλάδα	Εκκρεμεί	
3, Φεβ, 2022	Καθαρισμός κοινόχρηστων χώρων	4	Παπαθωμά Μέκαλη 64, Βελισσάριος 455 00, Ελλάδα	Εκκρεμεί	
3, Φεβ, 2022	Ανακύκλωση έλαιοψή κάδων	3	2η Πάροδος Παπαθωμά Μέκαλη, Βελισσάριος 453 32, Ελλάδα	Εκκρεμεί	
3, Φεβ, 2022	Ογκώδη αντικείμενα	2	2η Πάροδος Παπαθωμά Μέκαλη, Βελισσάριος 453 32, Ελλάδα	Εκκρεμεί	
3, Φεβ, 2022	Σκουπίδια έλαιοψή κάδων	1	2η Πάροδος Παπαθωμά Μέκαλη, Βελισσάριος 453 32, Ελλάδα	Εκκρεμεί	
4, Φεβ, 2022	Λακκούβες οδοστρώματος	τεστ	2η Πάροδος Παπαθωμά Μέκαλη, Βελισσάριος 453 32, Ελλάδα	Εκκρεμεί	
4, Φεβ, 2022	Φωτισμός δρόμων	γγη	2η Πάροδος Παπαθωμά Μέκαλη, Βελισσάριος 453 32, Ελλάδα	Εκκρεμεί	
4, Φεβ, 2022	Λακκούβες οδοστρώματος		JRXQ+R4 Βελισσάριος, Ελλάδα	Εκκρεμεί	
8, Φεβ, 2022	Λακκούβες οδοστρώματος	κατεστραμμένο οδόστρωμα	Λεωφ. Δωδώνης 77, Ιωάννινα 453 32, Ελλάδα	Εκκρεμεί	
8, Φεβ, 2022	Σκουπίδια έλαιοψή κάδων	χαλασμένοι κάδοι σκουπιδιών χωρίς κοπακι	Λεωκάδας 12, Βελισσάριος 453 32, Ελλάδα	Εκκρεμεί	

Στο πεδίο Τελευταίες αναφορές οι χρήστες της εφαρμογής μπορούν να δουν τη λίστα με τις 10 τελευταίες αναφορές.



Εικόνα 84 Τελευταίες αναφορές

Οι τελευταίες αναφορές εμφανίζονται με την χρήση Ο Html - JavaScript κώδικα (Παράρτημα Β , Last_10_Issues.html) που είναι εμφωλιασμένος στο Widget Shortcode της σελίδας **Home** του WordPress.

Όπως μπορούμε να δούμε για την σύνδεση με την Firebase χρησιμοποιούμε το SDK που έχουμε δημιουργία και την παραμετροποίηση της Firebase.

Τα στοιχεία τα διαβάζει από τη Firebase με την συνάρτηση

```
firebase.database().ref('Issues/').limitToLast(10).on('value', function (snapshot) {  
  var value = snapshot.val();
```

Εικόνα 85 Last_10_Issues



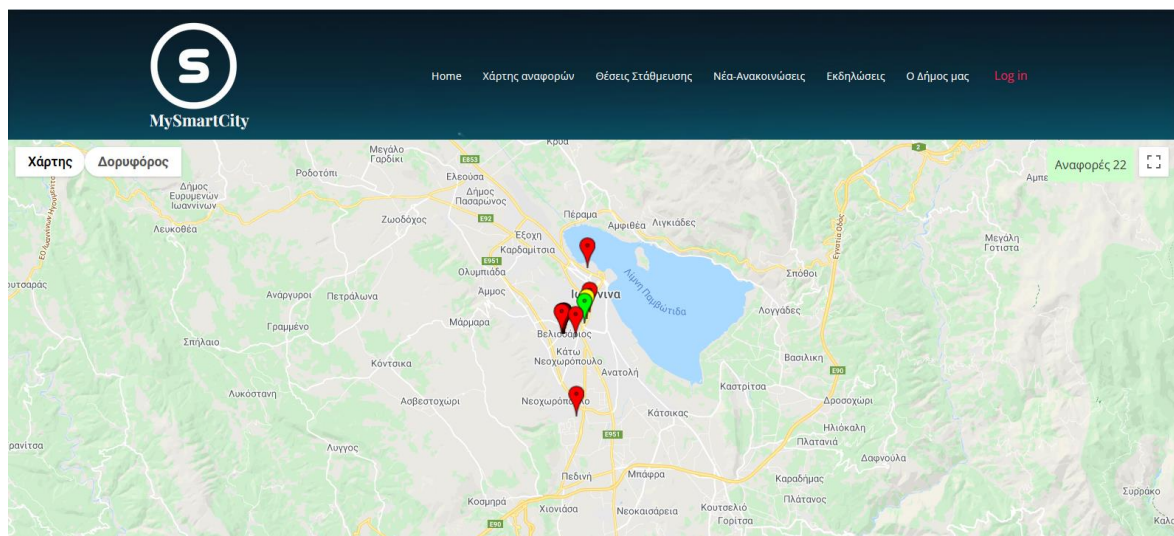
Εικόνα 86 Αναφορές στην Realtime Database

6.2.2. Χάρτης αναφορών

Από το κεντρικό μενού από την επιλογή **«Χάρτης αναφορών»** οι χρήστες μπορούν να δουν το χάρτη της περιοχής καθώς και τα σημάδια (marker) των αναφορών με χρώμα ανάλογο της κατάστασης τους. (κόκκινο για τα εκκρεμή, κίτρινο για τα υπό επεξεργασία και πράσινο για τα επιλυμένα).

Οι marker είναι τοποθετημένοι στο χάρτη στις γεωγραφικές συντεταγμένες που έχει καταχωρήσει ο χρήστης της mobile εφαρμογής.

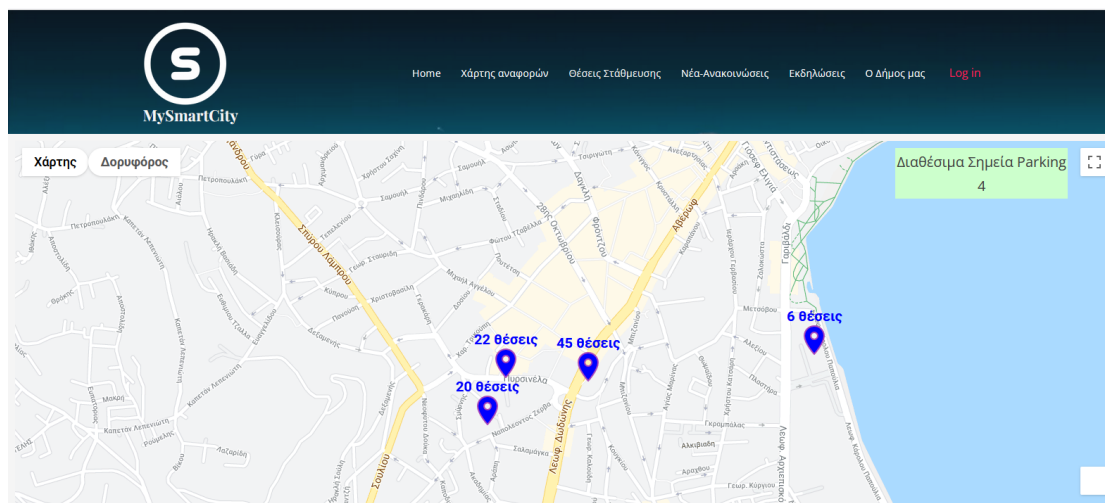
Αν με το δείκτη το mouse εστιάσουν πάνω στον marker εμφανίζεται η κατηγορία του προβλήματος.



Εικόνα 87 Χάρτης αναφορών

Και εδώ χρησιμοποιούμε Html -JavaScript κώδικα που είναι εμφωλιασμένος στο Widget Shortcode της σελίδας Maps του Wordpress (Παράρτημα Β , Last_10_Issues.html).

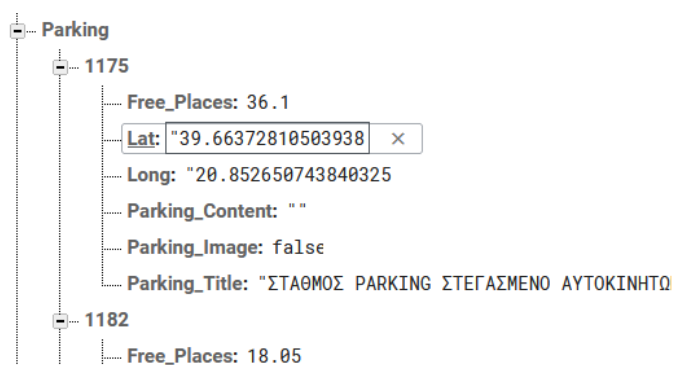
6.2.3. Θέσεις Στάθμευσης



Εικόνα 88 Θέσεις Στάθμευσης

Από το κεντρικό μενού από την επιλογή «Θέσεις Στάθμευσης» οι χρήστες μπορούν να δουν το χάρτη της περιοχής τους διαθέσιμους σταθμούς στάθμευσης καθώς και τις διαθέσιμες θέσεις.

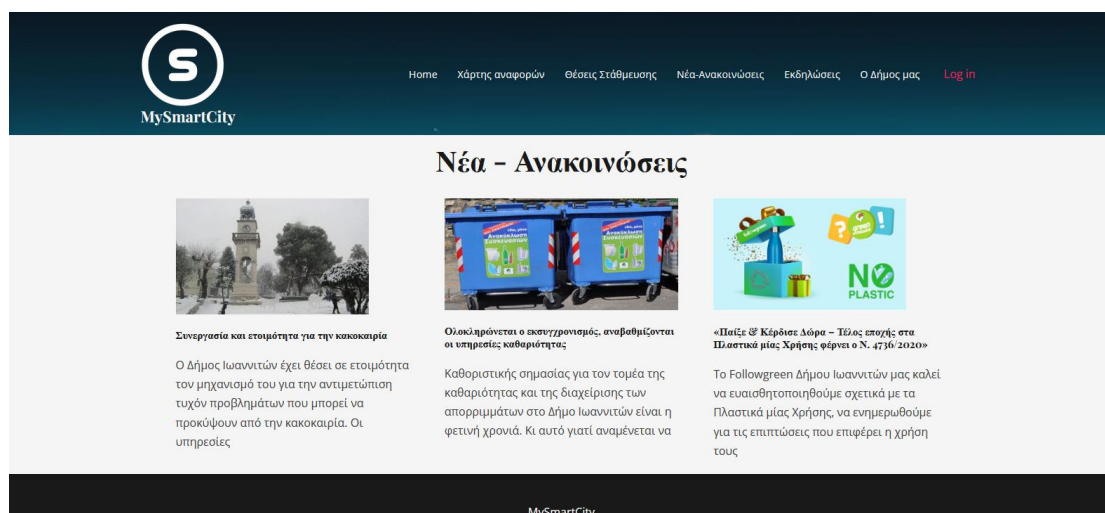
Οι σταθμοί σημειώνονται με marker και είναι τοποθετημένοι στο χάρτη στις γεωγραφικές συντεταγμένες που έχει καταχωρηθεί στην Firebase .



Εικόνα 89 Θέσεις Στάθμευσης στην Realtime Database

Για την εμφάνιση της σελίδας αυτής χρησιμοποιήθηκε ο κώδικας Html-JavaScript είναι εμφωλιασμενος στο Widget Shortcode της σελίδας Parking του WordPress. ((Παράρτημα Β , Parking.html)

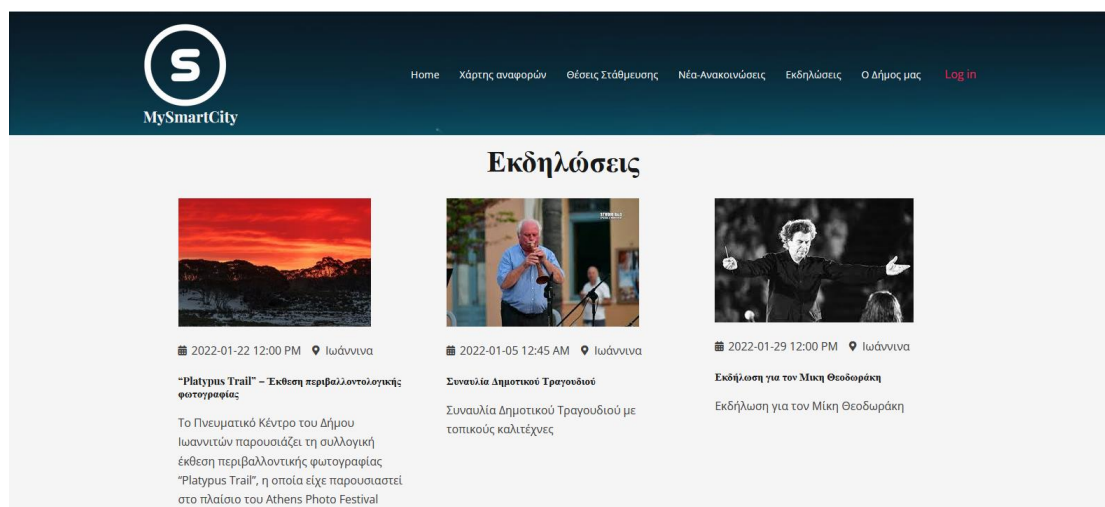
6.2.4. Νέα-Ανακοινώσεις



Εικόνα 90 Νέα - Ανακοινώσεις

Από το κεντρικό μενού από την επιλογή «**Νέα - Ανακοινώσεις**» » οι χρήστες μπορούν να δουν Νέα και Ανακοινώσεις του Δήμου προς τους πολίτες.

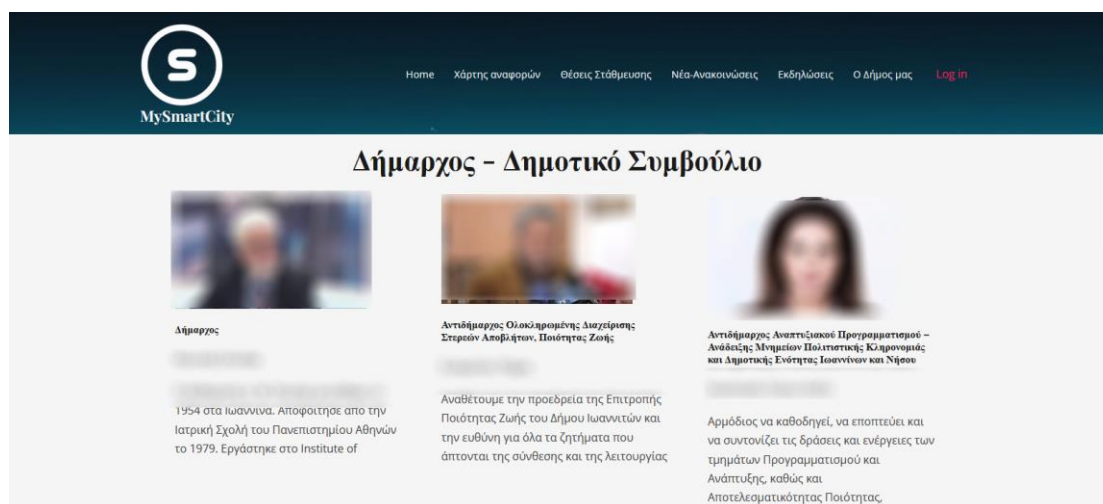
6.2.4.1. Εκδηλώσεις



Εικόνα 91 Εκδηλώσεις

Από το κεντρικό μενού από την επιλογή «**Εκδηλώσεις**» οι χρήστες μπορούν να δουν τις εκδηλώσεις που θα πραγματοποιηθούν στο Δήμο.

6.2.4.2. Ο Δήμος μας



Εικόνα 92 Ο Δήμος μας

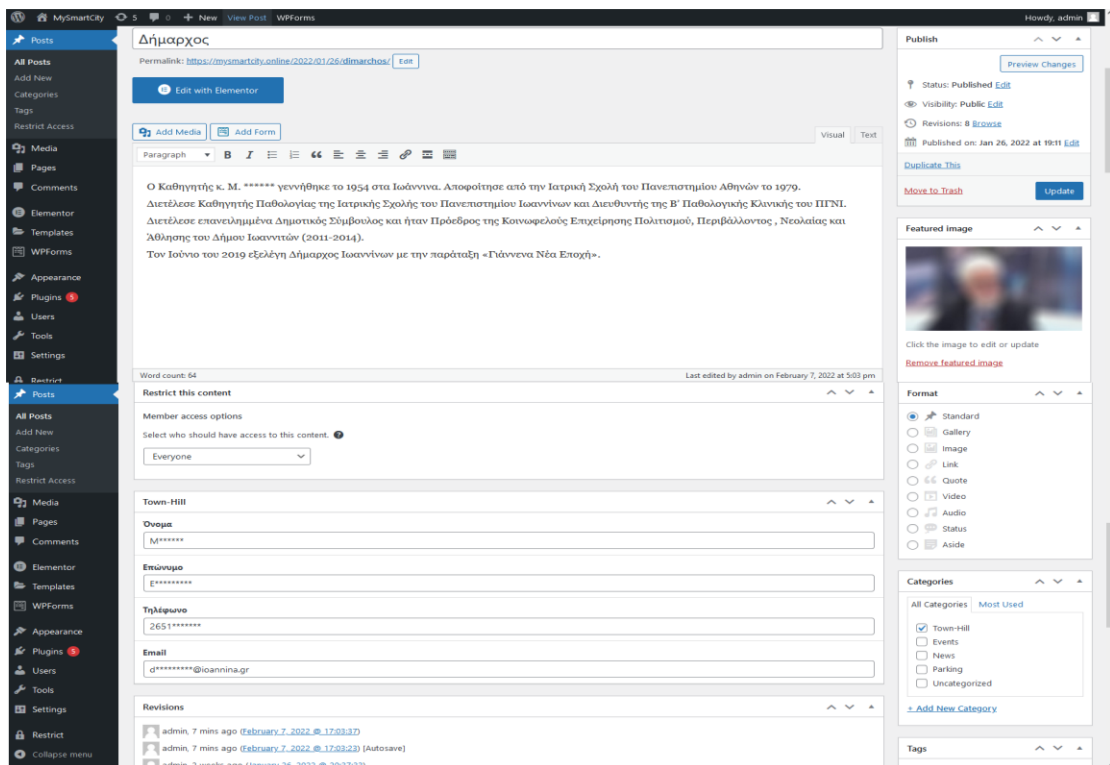
Από το κεντρικό μενού από την επιλογή «**Ο Δήμος μας**» οι χρήστες μπορούν να δουν πληροφορίες σχετικά με τον Δήμο, τον Δήμαρχο και Δημοτικό Συμβούλιο.

6.2.5. Διαχείριση εφαρμογής

Για την διαχείριση της εφαρμογής έχει δημιουργηθεί λογαριασμός με δικαιώματα **Subscriber** και με την χρήση των Plugin «**User Role Editor**», «**Restrict Content**» και «**Nav Roles Menu**» λαμβάνει τα κατάλληλα δικαιώματα πρόσβασης στο κρυφή επιλογή «**Αναφορές**» στο μενού. Ο λογαριασμός αυτός επίσης έχει δικαιώματα προσθήκης – επεξεργασίας – διαγραφής Post's.

6.2.5.1. Δημιουργία Posts (Ανακοινώσεις – Νέα, Events, Δήμος)

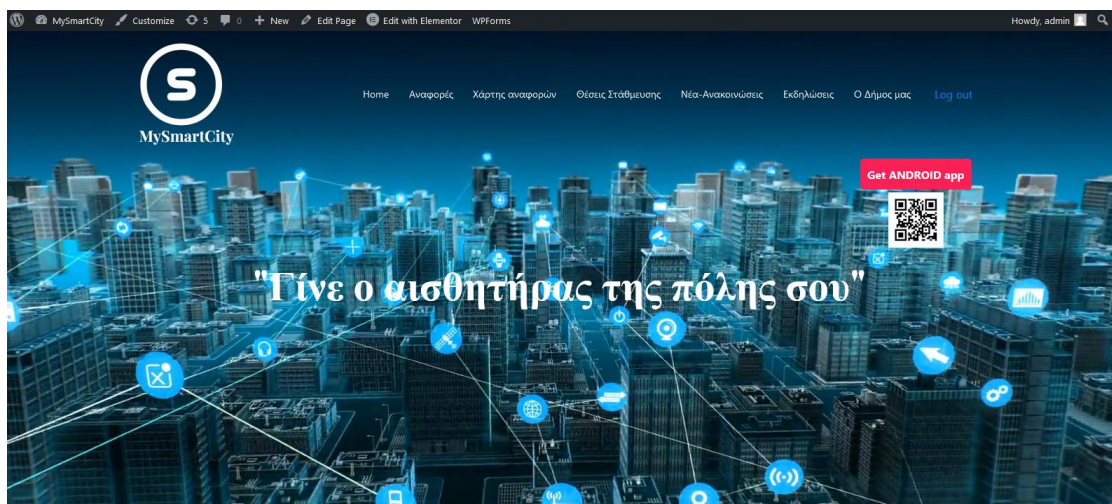
Το περιεχόμενων των τριών σελίδων (Νέα-Ανακοινώσεις, Εκδηλώσεις, Ο Δήμος μας) δημιουργούνται από τα άρθρα (Post's) που καταχωρεί ο διαχειριστής στο Back-end περιβάλλον του WordPress. Για την προσθήκη των περισσότερων πεδίων που απαιτήθηκαν στην εφαρμογή μας από αυτά που έχει το WordPress στην φόρμα των Post's χρησιμοποιήθηκε το Plugin "Custom Field Suite"



Εικόνα 93 Δημιουργία Post 's

Για την αποθήκευση των Post 's στην Firebase ώστε να τα διαβάζει η Mobile εφαρμογή χρησιμοποιήθηκε php κώδικας (Παράρτημα Β , Create_Post_in_Firebase.php) που είναι εμφωλιασμένος στο αρχείο **function.php** του θέματος.

6.2.5.2. Διαχείριση Αναφορών



Εικόνα 94 Αναφορές

6.2.5.2.1. Αναφορές

Από την σελίδα αυτή ο διαχειριστής μπορεί να ενημερωθεί για όλα τα προβλήματα που έχουν αναφερθεί , μπορεί να ενημερώσει το πεδίο **Ενέργειες** με τις εργασίες που έχουν γίνει για την αποκατάσταση του προβλήματος και να αλλάξει την **κατάσταση** από **Εκκρεμή** , σε **Εξέλιξη** ή σε **Επιλυμένο** ενημερώνοντας τα σχετικά πεδία στην Firebase.

The screenshot shows the 'SmartCity Management System' interface. It features a table titled 'Αναφορές' (Reports) with columns: Ημερομηνία, Κατηγορία, Περιγραφή, Τοποθεσία, Κατάσταση, Ενέργειες, and Action. An 'Update' modal is open, allowing the user to change the 'Κατάσταση' (Status) from 'Εκκρεμή' (Pending) to 'Εξέλιξη' (In Progress) or 'Επιλυμένο' (Resolved), and update the 'Ενέργειες' (Actions) field. The modal includes 'Close' and 'Update' buttons.

Ημερομηνία	Κατηγορία	Περιγραφή	Τοποθεσία	Κατάσταση	Ενέργειες	Action
16, Ιαν, 2022	Σκουπίδια έλκερη κάδων	Σκουπίδια	Λεωφ. Διαδύνος	Εκκρεμεί		<button>Update</button> <button>Delete</button>
16, Ιαν, 2022	Ανακύκλωση έλκερη κάδων			Εκκρεμεί	Γίνονται εργασίες αποκατάστασης	<button>Update</button> <button>Delete</button>
20, Ιαν, 2022	Φυτεσιές δρόμων			Εκκρεμεί	Γίνονται εργασίες αποκατάστασης	<button>Update</button> <button>Delete</button>
22, Ιαν, 2022	Λακκούφες οδοστρώματος			Εκκρεμεί		<button>Update</button> <button>Delete</button>
28, Ιαν, 2022	Λακκούφες οδοστρώματος	δοκιμη	ΙΒΚΟ+Q9 Βελισσάριος, Ελλάδα	Σε εξέλιξη	Σε εξέλιξη	<button>Update</button> <button>Delete</button>
31, Ιαν, 2022	Καθαρισμός κοινόχρηστων χώρων		ΙΒΚΟ+Q9 Βελισσάριος, Ελλάδα	Εκκρεμεί		<button>Update</button> <button>Delete</button>
1, Φεβ, 2022	Παζαρόδρια πλάκας	τραρ	2η Πάροδος Παπαθωμά	Εκκρεμεί		<button>Update</button> <button>Delete</button>

Εικόνα 95 Επεξεργασία αναφορών

Ο κώδικας που εκτελείται για την λειτουργία αυτή είναι εμφωλιασμένος στο Widget Shortcode στην σελίδα «**Αναφορές**» (Παράρτημα Β , Issues.html)

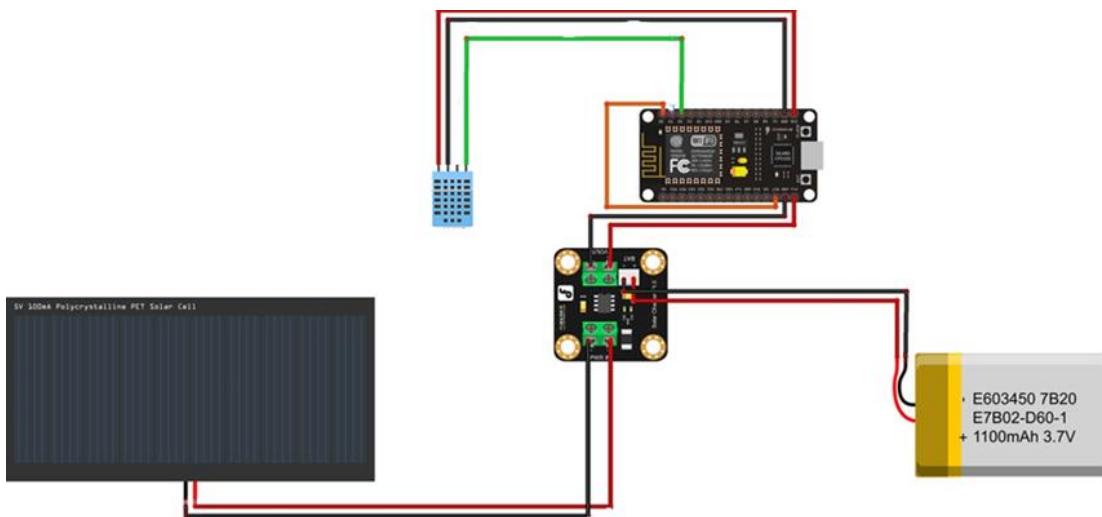
7. Επίγνωση πλαισίου αξιοποιώντας το Διαδίκτυο των Πραγμάτων

7.1. Μέτρηση περιβαλλοντικών συνθηκών.

Η μέτρηση των περιβαλλοντικών συνθηκών στη πόλη στηρίζεται σε ένα κύκλωμα Wi-Fi ESP8266 με τον αισθητήρα DHT22.

Η συγκεκριμένη κατασκευή αποτελεί ένα υλοποιημένο ενσωματωμένο σύστημα το οποίο, παίρνει μετρήσεις από έναν αισθητήρα θερμοκρασίας-υγρασίας και μέσω Wi-Fi επικοινωνίας, που επιτυγχάνεται με την βοήθεια εξαρτήματος, που λειτουργεί ως κάρτα δικτύου, στέλνει τα δεδομένα σε μία Cloud βάση δεδομένων (Firebase).

Για να κάνουμε το σύστημα εντελώς "αυτόνομο" ώστε να μπορεί να τοποθετηθεί σε σημεία χωρίς ηλεκτρικό ρεύμα η απαιτούμενη ισχύς παρέχεται από 1 ηλιακό συλλέκτη 6V που μπορεί να παρέχει 1W ισχύος. Η ενέργεια αποθηκεύεται σε μια μπαταρία ιόντων λιθίου πολυμερούς 3.7V με ικανότητα +/- 1000mAh. Για τη φόρτιση της μπαταρίας και την διαχείριση της ενέργειας χρησιμοποιείται το Solar Lipo Charger της DFRobot.



Εικόνα 96 Σχηματικό διάγραμμα ηλ. κυκλώματος IoT

7.1.1. Το Υλικό (Hardware)

7.1.1.1. ESP-8266 (NodeMcu)

Λειτουργίες

Το ESP8266 έχει πολλές εφαρμογές σε εφαρμογές IoT. Μερικές από τις λειτουργίες που χρησιμοποιείται το τσιπ:

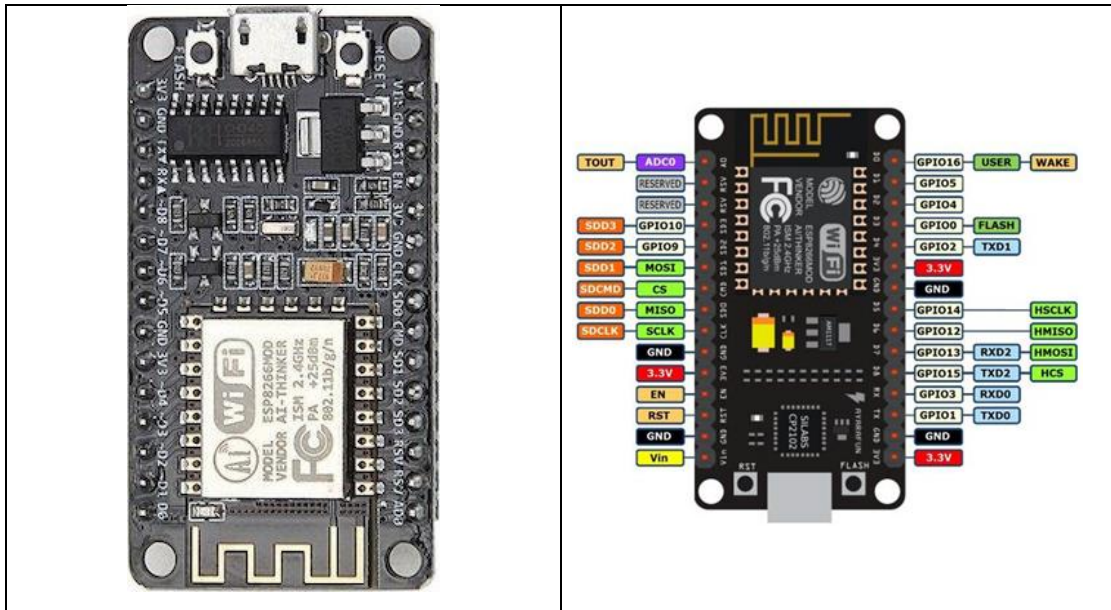
- Δικτύωση: Η κεραία Wi-Fi της μονάδας επιτρέπει στις ενσωματωμένες συσκευές να συνδέονται σε δρομολογητές και να μεταδίδουν δεδομένα
- Επεξεργασία δεδομένων: Περιλαμβάνει την επεξεργασία βασικών εισόδων από αναλογικούς και ψηφιακούς αισθητήρες για πολύ πιο σύνθετους υπολογισμούς με RTOS ή SDK χωρίς λειτουργικό σύστημα
- Συνδεσιμότητα P2P: Δημιουργία άμεσης επικοινωνίας μεταξύ ESP και άλλων συσκευών χρησιμοποιώντας συνδεσιμότητα IoT P2P
- Διακομιστής Web: Σελίδες πρόσβασης γραμμένες σε HTML ή γλώσσες ανάπτυξης

Εφαρμογές

Οι μονάδες ESP8266 βρίσκονται συνήθως στις ακόλουθες συσκευές IoT:

- Έξυπνες συσκευές ασφαλείας, συμπεριλαμβανομένων καμερών επιτήρησης και έξυπνων κλειδαριών
- Έξυπνες συσκευές ενέργειας, συμπεριλαμβανομένων των HVAC και των θερμοστατών
- Έξυπνες βιομηχανικές συσκευές, συμπεριλαμβανομένων των προγραμματιζόμενων λογικών ελεγκτών (PLC)
- Έξυπνες ιατρικές συσκευές, συμπεριλαμβανομένων των φορητών συστημάτων παρακολούθησης της υγείας

Το **NodeMcu** που θα χρησιμοποιήσουμε είναι ένα κιτ ανάπτυξης βασισμένο στο ESP8266 με firmware ανοιχτού κώδικα που μας βοηθά να αναπτύξουμε ένα προϊόν IoT σας σε λίγες γραμμές κώδικα Έχει προηγμένο API για το υλικό IO, το οποίο μπορεί να μειώσει δραματικά την περιττή εργασία για τη διαμόρφωση και το χειρισμό υλικού.



Εικόνα 97 NodeMcu

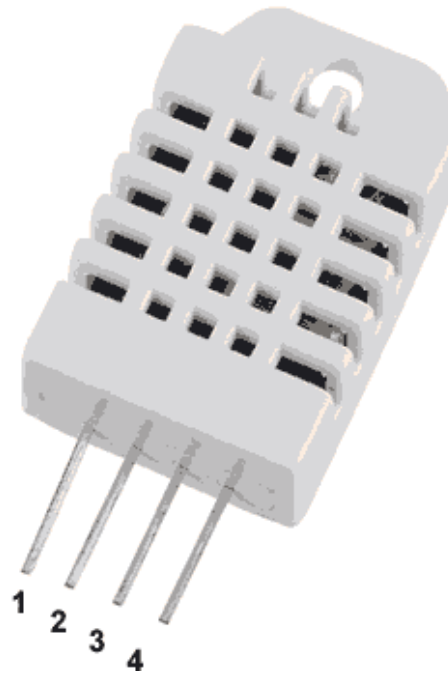
Τα Χαρακτηριστικά NodeMcu είναι:

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	Micro-USB: NodeMCU can be powered through the USB port 3.3V: Regulated 3.3V can be supplied to this pin to power the board GND: Ground pins Vin: External Power Supply
Control Pins	EN, RST	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you must find which pin is I2C.

Πίνακας 2 Τεχνικά χαρακτηριστικά NodeMcu

7.1.1.2. DHT22

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



Εικόνα 98 DHT22 Sensor

Το DHT22 χρησιμοποιεί αποκλειστική ψηφιακή τεχνική συλλογής σήματος και τεχνολογία ανίχνευσης θερμοκρασίας υγρασίας και μπορεί να εξάγει βαθμονομημένο ψηφιακό σήμα. Το μικρό μέγεθος & η χαμηλή κατανάλωση & η μεγάλη απόσταση μετάδοσης (20 μέτρα) επιτρέπουν AM2303 να είναι κατάλληλη σε όλα τα είδη σκληρών εφαρμογών.

Χαρακτηριστικά:

- Τάση τροφοδοσίας: 3,3-5 V
- Εύρος μέτρησης: υγρασία 0-100% RH και θερμοκρασία -40 έως +125 βαθμοί κελσίου
- Ακρίβεια: υγρασία +- 2% RH και θερμοκρασία +-0,5% RH
- Ο ρυθμός δειγματοληψίας δεν είναι περισσότερο από 0,5 Hz (μια φορά κάθε δύο δευτερόλεπτα)

7.1.1.3. Lipo Charger DFR0264

Αυτός ο ηλιακός φορτιστής lipo σχεδιάζεται για μια ενιαία μπαταρία λιθίου (3.7V) για την ευφυή φόρτιση, με την αντίστροφη προστασία πολικότητας εισαγωγής. Το

μέγιστο ρεύμα φόρτισης είναι 500mA και οι συνδέσεις είναι εύκολες (κοχλιωτό τερματικό για Solar/Power Out και JST PH για LiPo).



Εικόνα 99 Lipo Charger DFR0264

Ο φορτιστής χρησιμοποιείται στα έργα για αισθητήρες τηλεχειρισμού, ηλιακά έργα χαμηλής ισχύος, εκπαιδευτική χρήση

Χαρακτηριστικά:

- Τάση εισόδου: 4,4 ~ 6 V
- Ρεύμα φόρτισης: 500mA Max (Ανάλογα με την ικανότητα παροχής ηλιακής μπαταρίας)
- Τάση Διακοπής Φόρτισης: 4,2 V
- Θερμοκρασία λειτουργίας: -40 ~ 85°C
- Μέγεθος μονάδας: 33 mm * 33 mm * 12 mm
- Απαιτούμενη μπαταρία: Μπαταρία λιθίου 3.7V
- Ηλιακό Panel : 4.4V έως 6V, έως 500mA (ή 2 Watts)

(https://wiki.dfrobot.com/Solar_Charger_SKU_DFR0264)

7.1.1.4. Solar Panel



Εικόνα 100 Ηλιακός συλλέκτης

Υψηλός ρυθμός μετατροπής, υψηλή απόδοση παραγωγής σε χαμηλό φωτισμό
Κατάλληλο για φόρτιση κινητών τηλεφώνων και μικρών μπαταριών συνεχούς ρεύματος.

Χαρακτηριστικά:

Μέγεθος: 112*84 χιλ.

Τροφοδοσία: 6V 200mA 1,1W

7.1.2. Το Λογισμικό (Software)

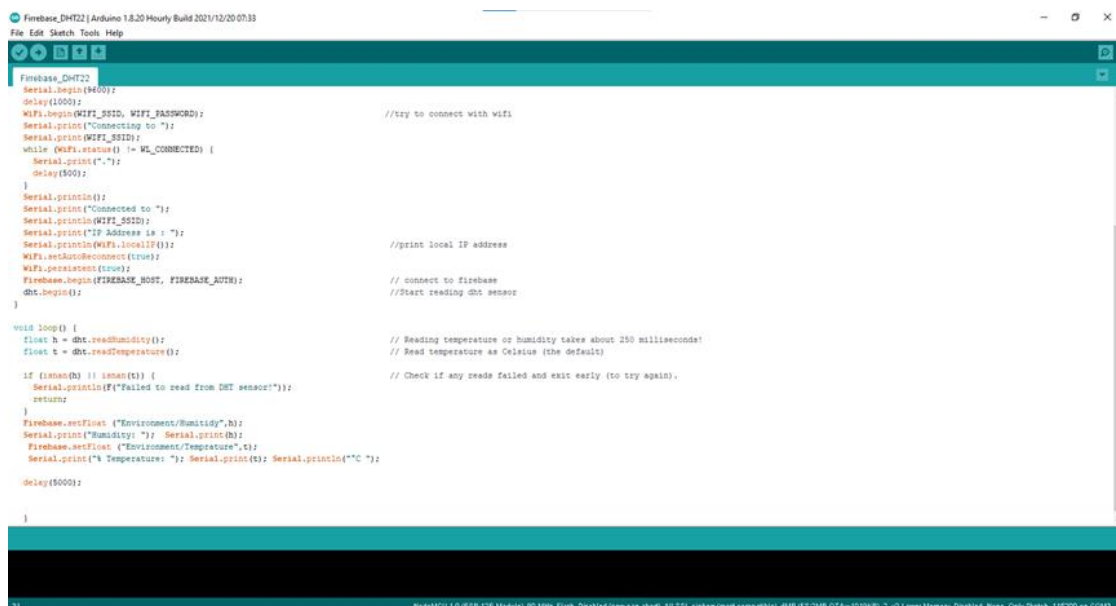
Ο Προγραμματισμός του μικροελεγκτή NodeMcu έγινε με την χρήση του ολοκληρωμένου περιβάλλοντος προγραμματισμού Arduino IDE.

7.1.2.1. Περιβάλλον προγραμματισμού Arduino IDE

Το Integrated Development Environment ή IDE του Arduino είναι η επίσημη εφαρμογή της εταιρείας και λειτουργεί ως γέφυρα μεταξύ της πλακέτας και υπολογιστή του χρήστη. Το Arduino IDE βασίζεται στη γλώσσα προγραμματισμού Processing και επομένως έχει καλή συμβατότητα με τα περισσότερα λειτουργικά συστήματα όπως τα Windows, το Linux και το Mac OS X. Κάθε πρόγραμμα που καλείται να γράψει ο χρήστης είναι γραμμένο στη γλώσσα προγραμματισμού Wiring. Μετά τη μετα-

φόρτωση του κώδικα, μετατρέπεται σε γλώσσα C. Αυτό το πλήρες περιβάλλον ανάπτυξης παρέχεται στους χρήστες δωρεάν και οι χρήστες μπορούν να το κατεβάσουν από την επίσημη σελίδα του Arduino.

Το Arduino IDE πέρα από τις διάφορες εκδόσεις πλακετών Arduino (Arduino Uno R3, Arduino Nano , Arduino Micro, Arduino Mega 2560) μέσα από τις πλούσιες βιβλιοθήκες υποστηρίζει και διάφορες ακόμη πλακέτες όπως ESP8266 -ESP-01S , NodeMcu κ, α) .



Εικόνα 101 Arduino IDE

Ο κώδικας της εφαρμογής θα μπορούσε να χωριστεί σε τρία τμήματα:

Στο 1^ο τμήμα ορίζονται οι βιβλιοθήκες που θα χρησιμοποιηθούν , τα στοιχεία σύνδεσης στο Wi-Fi καθώς και τα στοιχεία σύνδεσης με την Firebase

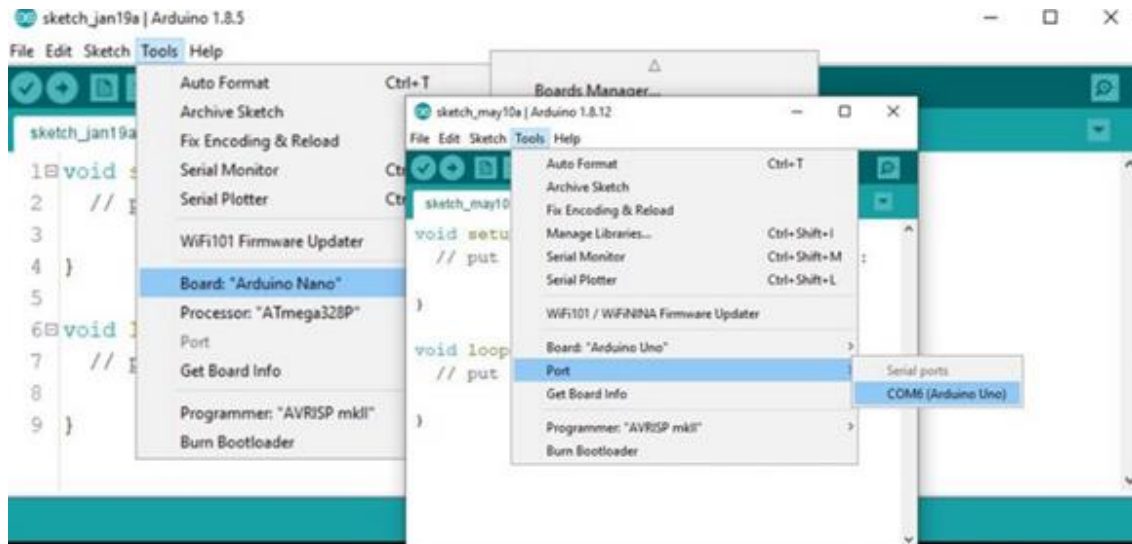
Έτσι , για να μπορέσουμε να χρησιμοποιήσουμε το ESP8266 θα πρέπει να κάνουμε χρήση της σχετικής βιβλιοθήκης μέσω της εντολής `#include <ESP8266WiFi.h>` .

Για την μπορέσει το NodeMcu να διαβάσει τιμές από τον αισθητήρα DHT22 θα πρέπει να κάνουμε χρήση της βιβλιοθήκης μέσω της εντολής `#include <DHT.h>` .

Για την σύνδεση με την Firebase κάνουμε χρήση της βιβλιοθήκης μέσω της εντολής `#include <FirebaseArduino.h>` .

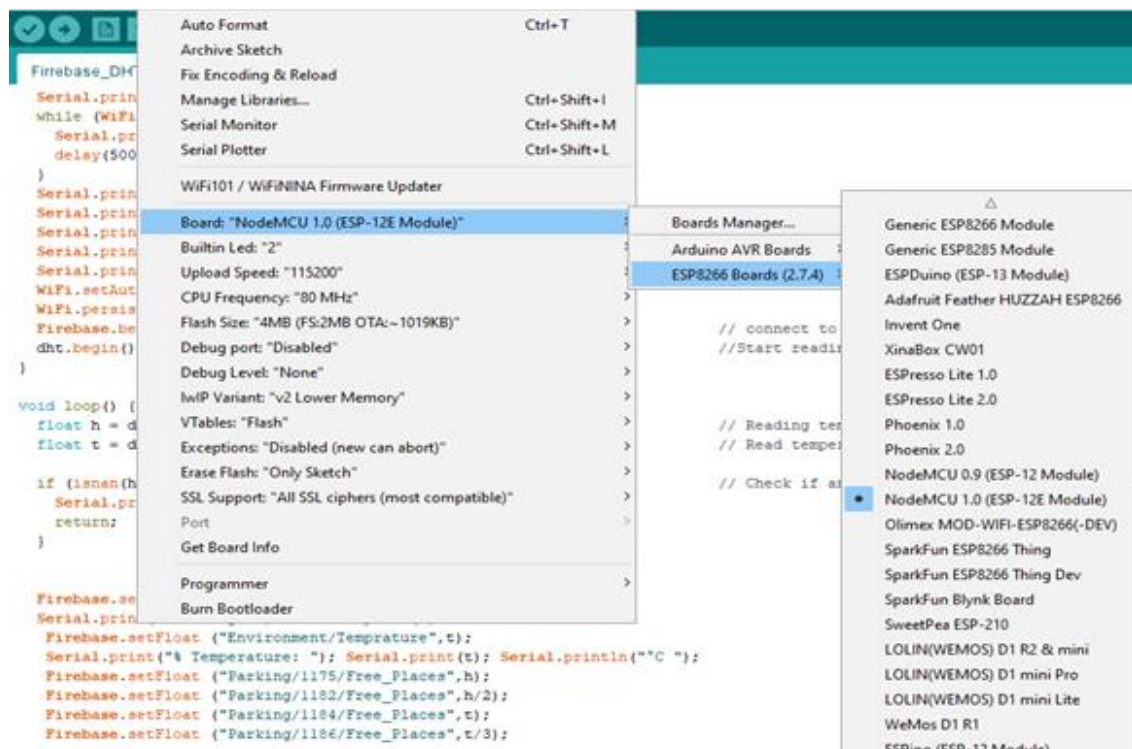
Στο 2^ο τμήμα γίνεται ο έλεγχος και η σύνδεση στο Wi-Fi , καθώς και η σύνδεση με την Firebase.

Στο 3^ο τμήμα γίνεται κάθε 5 λεπτά η ανάγνωση των τιμών θερμοκρασίας και υγρασίας από το αισθητήριο DHT22 και αποστολή στα αντίστοιχα tags της Firebase . Για τον προγραμματισμό του NodeMcu από το μενού πηγαίνουμε στα **Tools** μετά στο **Port** και επιλέγουμε την **σωστή θύρα (Port)**.



Εικόνα 102 Arduino IDE Port Settings

Στην συνέχεια από το μενού επιλέγουμε **Tools** και μετά επιλέγουμε το υπομενού **Board**, όπου εκεί θα πρέπει από την λίστα να επιλέξουμε την σωστή πλακέτα.



Εικόνα 103 Arduino IDE Board Select

Ο προγραμματισμός με τον κώδικα της εφαρμογής (Παράρτημα Γ , Firrebase_DHT22.ino) γίνεται από το μενού Sketch πατώντας **upload**

7.1.3. Υλοποίηση συσκευής IoT



Εικόνα 104 Άνω όψη συσκευής IoT

Το ηλεκτρονικό κύκλωμα του IoT τοποθετήθηκε σε Πλαστικό κουτί στεγανό IP66 διαστάσεων 11x15 cm και το ηλιακό Panel τοποθετήθηκε στο καπάκι του κουτιού.



Εικόνα 105 Εσωτερική εικόνα συσκευής IoT

Εσωτερικά τοποθετήθηκε ο μικροελεγκτής NodeMcu , ο ελεγκτής φόρτισης και η μπαταρία.



Εικόνα 106 Κάτω όψη συσκευής IoT

Αισθητήρας θερμοκρασίας- υγρασίας τοποθετήθηκε στην βάση του κουτιού για προστασία από την βροχή.

7.2. Θέσεις στάθμευσης

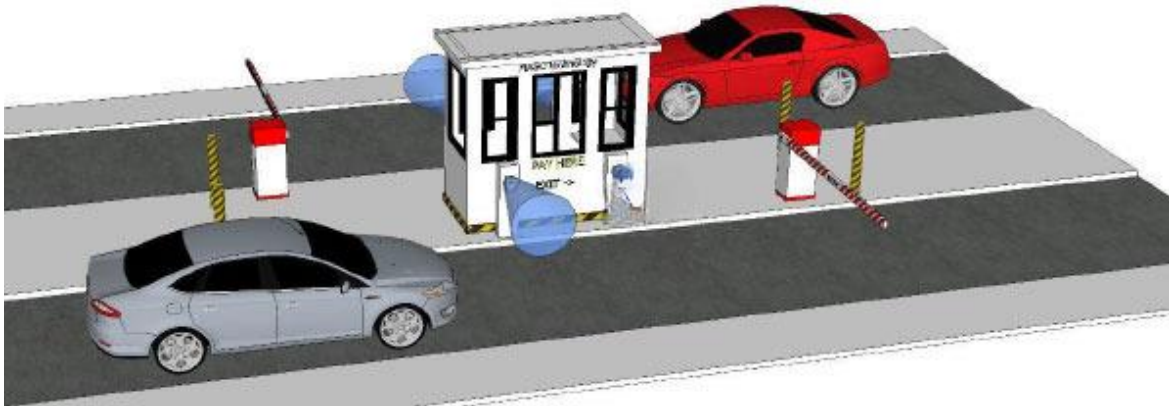
Για την προβολή των διαθέσιμων θέσεων στάθμευσης θα μπορούσε να χρησιμοποιηθεί οποιοδήποτε κύκλωμα υποστηρίζει σύνδεση με Google Firebase και με την παροχή των κατάλληλων credentials να έχει την δυνατότητα να «γράψει» σε αυτή τις διαθέσιμες θέσεις στάθμευσης.

Τέτοια ηλ. κυκλώματα IoT χρησιμοποιούν κάποιον αισθητήρα, όπως φωτός, αισθητήρα μαγνητικού πεδίου, αισθητήρα πίεσης ή αισθητήρα υπέρυθρων ακτινών για την ανίχνευση και απαρίθμηση των εισερχόμενων και εισερχόμενων αυτοκινήτων. Γνωρίζοντας λοιπόν την αρχική χωρητικότητα των σταθμών στάθμευσης μπορούμε να υπολογίσουμε τις διαθέσιμες θέσεις σταθμεύσεις. Κάθε αλλαγή στον αριθμό των διαθέσιμων θέσεων στάθμευσης αποστέλλεται στην Firebase και είναι διαθέσιμη στους χρήστες μέσω της mobile ή της web εφαρμογής

Ο κάθε σταθμός στάθμευσης λοιπόν θα μπορούσε να έχει την δική του υλοποίηση η οποία όμως θα πρέπει να είναι συμβατή με την δομή της Firebase που χρησιμοποιεί η εφαρμογή μας.

7.2.1. Το Υλικό (Hardware)

Στην υλοποίηση μας υποθέσαμε ότι ο έλεγχος εισόδου και εξόδου των αυτοκινήτων γίνεται από κάποια πύλη ή γκαραζόπορτα όπως αυτό συνήθως συμβαίνει στους πιο πολλούς οργανωμένους σταθμούς στάθμευσης.



Εικόνα 107 Έλεγχος εισόδου εξόδου αυτοκινήτων

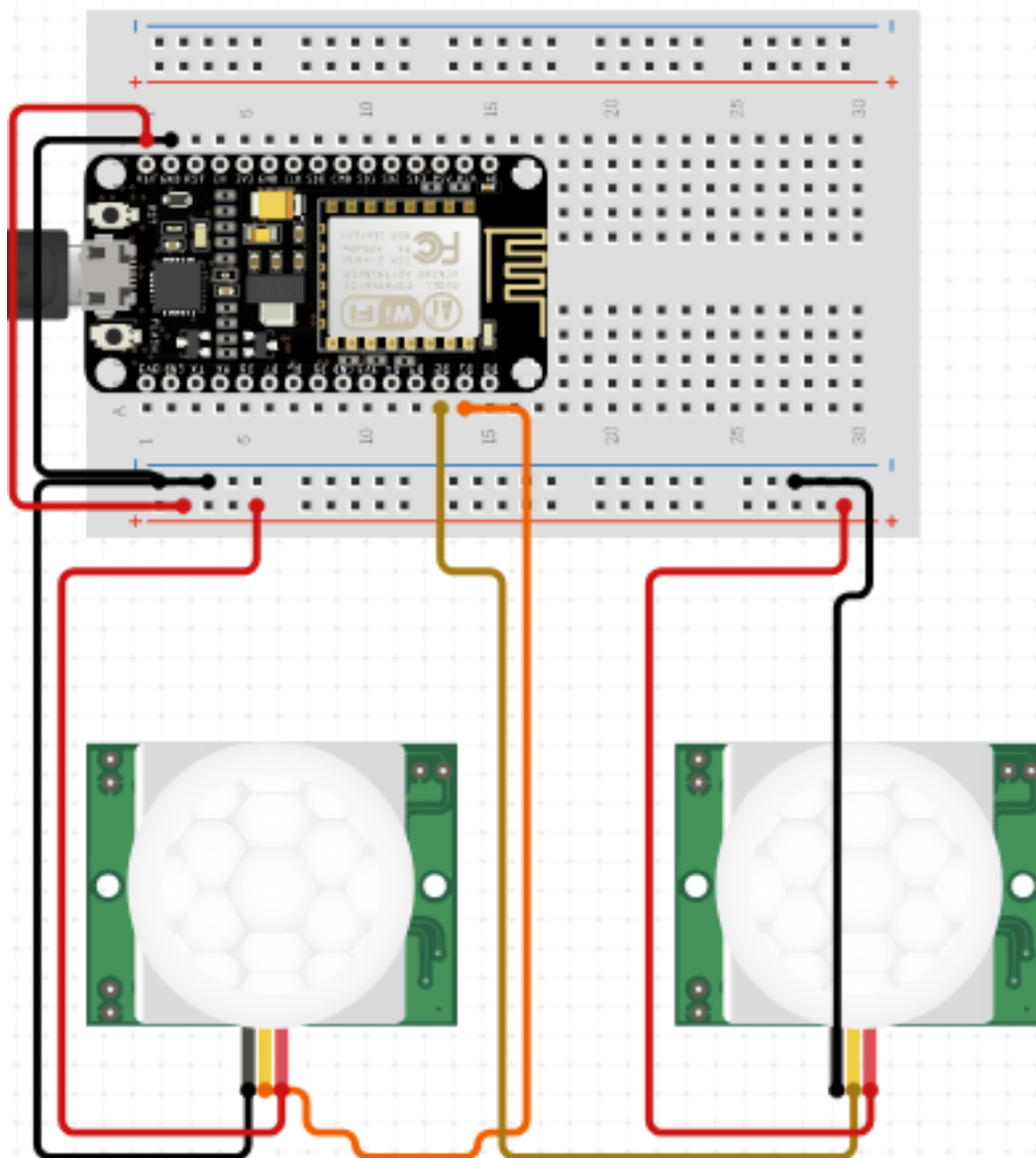
Πηγή (<https://www.tisparking.com/h-counter-vehicle-counting-parking-garage/>)

Το κύκλωμα που χρησιμοποιήσαμε αποτελείται από ένα Wi-Fi ESP8266 και 2 (δύο) αισθητήρες κίνησης Passive Infrared (PIR) HC-SR501.

Στην συγκεκριμένη κατασκευή οι αισθητήρες είναι τοποθετημένοι κατάλληλα, ένας σε κάθε κατεύθυνση ώστε να μπορούν να ανιχνεύουν τη είσοδο και την έξοδο των αυτοκινήτων αντίστοιχα από τον σταθμό στάθμευσης.

Στην συνέχεια γίνεται ο αριθμητικός υπολογισμός και μέσω Wi-Fi επικοινωνίας, στέλνει τα δεδομένα σε μία Cloud βάση δεδομένων (Firebase).

Το σύστημα επειδή θα τοποθετηθεί σε κάποιο ηλεκτροδοτούμενο οίκημα ή κατασκευή τροφοδοτείται από το ηλεκτρικό δίκτυο και δεν αιτείται κάποιο κύκλωμα αυτόνομης πηγής ηλ.ρεύματος όπως ηλιακός συλλέκτης με μπαταρία.

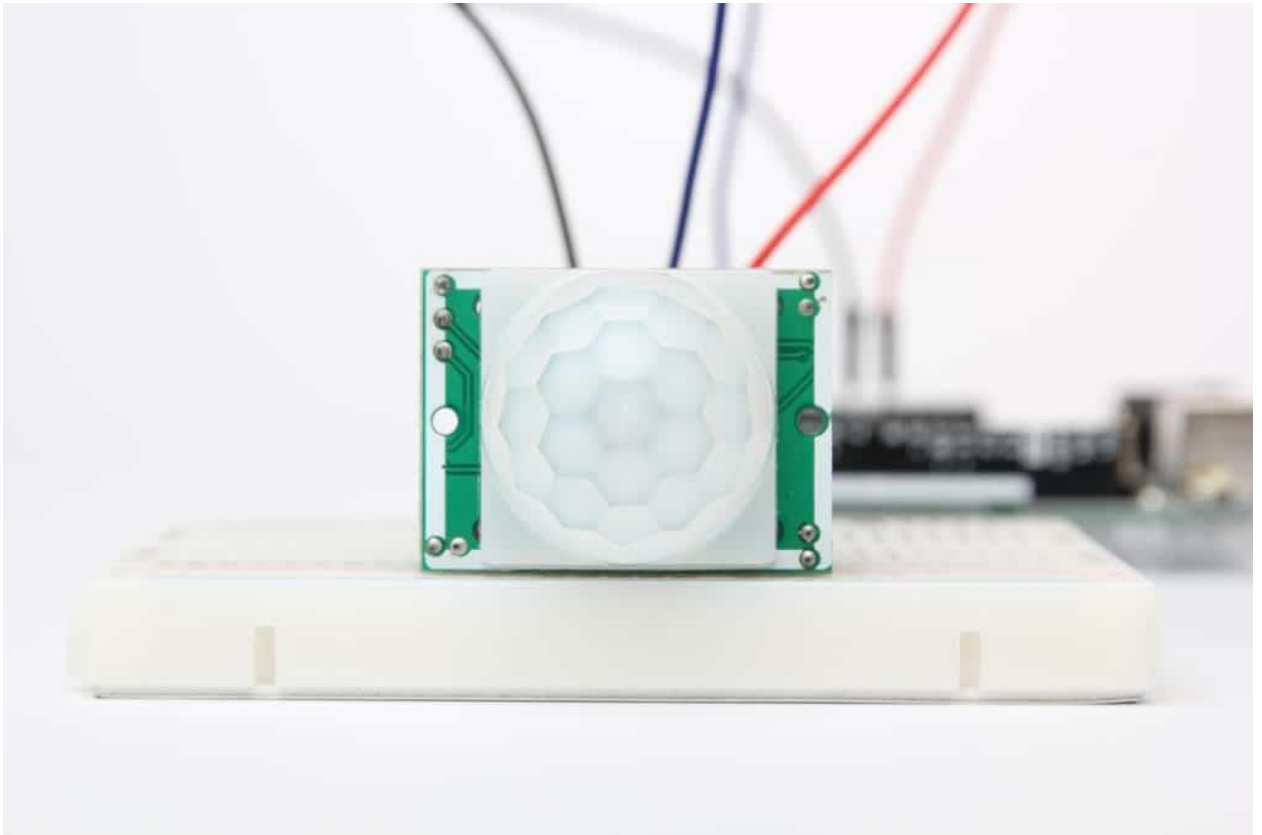


Είσοδος Αυτοκινήτων

Έξοδος Αυτοκινήτων

Εικόνα 108 Κύκλωμα IoT μέτρησης αυτοκινήτων

7.2.1.1. HC-SR501

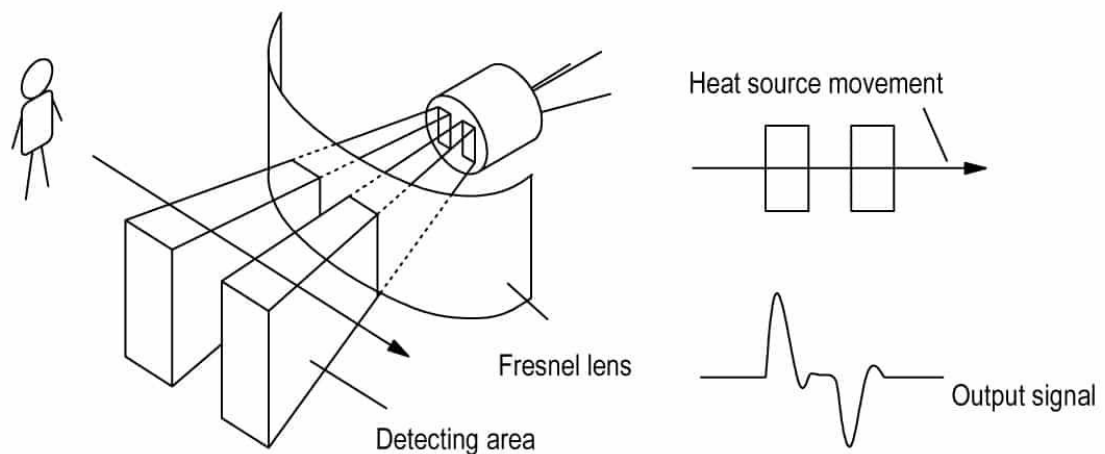


Εικόνα 109 Ο Αισθητήρας HC-SR501

(Πηγή: <https://www.makerguides.com/hc-sr501-arduino-tutorial/>)

Οι αισθητήρες κίνησης PIR αποτελούνται από δύο κύρια μέρη: ένα πυροηλεκτρικό αισθητήριο στοιχείο και έναν φακό Fresnel. Το πυροηλεκτρικό αισθητήριο στοιχείο μπορεί να ανιχνεύσει την υπέρυθρη ακτινοβολία. Όλα τα αντικείμενα με θερμοκρασία πάνω από το απόλυτο μηδέν (0 Kelvin / -273,15 °C) εκπέμπουν θερμική ενέργεια με τη μορφή υπέρυθρης ακτινοβολίας, συμπεριλαμβανομένων των ανθρώπινων σωμάτων.

Ο πυροηλεκτρικός αισθητήρας έχει δύο ορθογώνιες υποδοχές από υλικό που επιτρέπει τη διέλευση της υπέρυθρης ακτινοβολίας. Πίσω από αυτά, υπάρχουν δύο ξεχωριστά ηλεκτρόδια αισθητήρα υπερθύρων, το ένα υπεύθυνο για την παραγωγή θετικής εξόδου και το άλλο αρνητικής εξόδου. Τα δύο ηλεκτρόδια είναι καλωδιωμένα έτσι ώστε να αλληλοεξουδετερώνονται. Εάν το ένα μισό δει περισσότερη ή λιγότερη ακτινοβολία υπερθύρων από το άλλο, η έξοδος θα ταλαντευτεί ψηλά ή χαμηλά



Εικόνα 110 HC-SR501 Ανίχνευση κίνησης

Τεχνικά χαρακτηριστικά

Operating voltage	4.5 – 20 V
Quiescent current	50 μ A
Trigger	L single trigger / H repeating trigg
Delay time	3 – 300 s
Measuring range	3 – 7 m maximum
Measuring angle	< 110° cone angle

Πίνακας 3 HC-SR501 Τεχνικά χαρακτηριστικά

7.2.2. Το Λογισμικό (Software)

Ο Προγραμματισμός του μικροελεγκτή NodeMcu έγινε με την χρήση του ολοκληρωμένου περιβάλλοντος προγραμματισμού Arduino IDE.

7.2.2.1. Περιβάλλον προγραμματισμού Arduino IDE

Ο προγραμματισμός της συσκευής IoT έγινε μέσω του Integrated Development Environment ή IDE (Integrated Development Environment) του Arduino .

Ο κώδικας θα μπορούσε να χωριστεί σε τρία τμήματα:

Στο 1^ο τμήμα ορίζονται οι βιβλιοθήκες που θα χρησιμοποιηθούν , τα στοιχεία σύνδεσης στο Wi-Fi καθώς και τα στοιχεία σύνδεσης με την Firebase

Έτσι , για να μπορέσουμε να χρησιμοποιήσουμε το ESP8266 θα πρέπει να κάνουμε χρήση της σχετικής βιβλιοθήκης μέσω της εντολής `#include <ESP8266WiFi.h>` .

Για την σύνδεση με την Firebase κάνουμε χρήση της βιβλιοθήκης μέσω της εντολής `#include <DHT.h>` .

Στο 2^ο τμήμα γίνεται ο έλεγχος και η σύνδεση στο Wi-Fi , καθώς και η σύνδεση με την Firebase.

Στο 3^ο τμήμα γίνεται κάθε 2,5 sec, ανάγνωση της τάσης κάθε αισθητήρα PIR. Αν για τον αισθητήρα εισόδου έχουμε κατάσταση HIGH σημαίνει ότι ένα αυτοκίνητο μπήκε στο parking και αφαιρεί μία από τις συνολικές διαθέσιμες θέσεις . Αν όμως για τον αισθητήρα εξόδου έχουμε κατάσταση HIGH σημαίνει ότι ένα αυτοκίνητο βγήκε από parking και προσθέτει μία ακόμη διαθέσιμη θέση.

Ο κώδικας προγραμματισμού της IoT συσκευής μέτρησης θέσεων στάθμευσης βρίσκεται στο Παράτημα Δ , Parking.ino

8. Αξιολόγηση

Για την αξιολόγηση της χρηστικότητας της εφαρμογής "MySmartCity» δημιουργήθηκε ένα ερωτηματολόγιο σχεδιασμένο στο δωρεάν λογισμικό διαχείρισης έρευνας Google Forms.

Το ερωτηματολόγιο βασίστηκε στην κλίμακα αξιολόγησης SUS (System Usability Scale) η οποία εφευρέθηκε από τον John Brooke το 1986. Είναι ένας γρήγορος (quick and dirty, όπως χαρακτηρίστηκε από τον ίδιο), αλλά και ένας αξιόπιστος τρόπος μέτρησης και βαθμολόγησης της ευχρηστίας οποιουδήποτε συστήματος. Με αυτό μπορεί να μετρηθεί η ευχρηστία ιστοσελίδων, online καταστημάτων και online εφαρμογών, αλλά και κινητών τηλεφώνων, υπολογιστικών συστημάτων, λογισμικού, τηλεφωνικών κέντρων, ακόμα και τηλεφωνικών καταλόγων.

Η βαθμολογική κλίμακα του SUS ορίζεται ως ένας αριθμός από το 0 έως το 100. Όσο μεγαλύτερος είναι ο δείκτης SUS της εφαρμογής, τόσο πιο εύχρηστη θεωρείται η εφαρμογή. Ο βαθμός SUS προκύπτει από τις απαντήσεις του χρήστη που δοκιμάζει την εφαρμογή σε 10 ερωτήσεις που απαντώνται μετά την ολοκλήρωση κάθε τεστ με μια κλίμακα 1-5.

Η βαθμολόγηση των απαντήσεων είναι:

- Διαφωνώ απόλυτα: 1 σημείο
- Διαφωνώ: 2 βαθμοί
- Ουδέτερο: 3 βαθμοί
- Συμφωνώ: 4 βαθμοί
- Συμφωνώ απόλυτα: 5 βαθμοί

Οι ερωτήσεις που περιλαμβάνει το SUS είναι οι παρακάτω:

- Νομίζω ότι θα ήθελα να χρησιμοποιώ συχνά αυτό το σύστημα.
- Βρήκα το σύστημα χωρίς λόγο πολύπλοκο.
- Νομίζω ότι το σύστημα είναι εύκολο στη χρήση.
- Πιστεύω ότι θα χρειαζόμουν την υποστήριξη ενός τεχνικού προσώπου για να μπορέσω να χρησιμοποιήσω αυτό το σύστημα.
- Βρήκα ότι οι διάφορες λειτουργίες σε αυτό το σύστημα ήταν καλά ενσωματωμένες.

- Νόμιζα ότι υπήρχε υπερβολική ασυνέπεια σε αυτό το σύστημα.
- Φαντάζομαι ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν αυτό το σύστημα πολύ γρήγορα.
- Βρήκα το σύστημα πολύ δύσκαμπτο να χρησιμοποιηθεί.
- Ένιωσα πολύ σίγουρος χρησιμοποιώντας το σύστημα.
- Έπρεπε να μάθω πολλά πράγματα πριν προχωρήσω με αυτό το σύστημα

Με την βαθμολόγηση SUS μπορούμε να έχουμε άποψη για την απόδοση χρηστικότητας της αποτελεσματικότητας, της αποδοτικότητας, και της συνολικής ευκολίας χρήσης της εφαρμογής.

Η βαθμολογία υπολογίζεται $SUS = (X + Y) \times 2,5$

$X =$ Άθροισμα των σημείων για όλες τις ερωτήσεις με περιττό αριθμό - 5

$Y = 25 -$ Άθροισμα των σημείων για όλες τις ερωτήσεις με άρτιο αριθμό

Η μέση βαθμολογία SUS είναι 68.

Με βάση την βαθμολογία SUS η εφαρμογή αξιολογείται:

- | | |
|-------------|-----------|
| • > 80.3 | Πολύ καλή |
| • 68 – 80.3 | Καλή |
| • 68 | Εντάξει |
| • 51 – 68 | Κακή |
| • < 51 | Πολύ κακή |

8.1. Περιγραφή διαδικασίας αξιολόγησης

Οι επισκέπτες είχαν πρόσβαση στο ερωτηματολόγιο αυτό μέσω σχετικού συνδέσμου. που προωθήθηκε μέσω email αλλά και με την χρήση των μέσων κοινωνικής δικτύωσης.

Μέτρηση ευχρηστίας συστήματος MySmartCity

Η εφαρμογή MySmartCity είναι μια εφαρμογή για έξυπνες πόλεις αξιοποιώντας τεχνολογίες IOT. Επισκεφτείτε την ιστοσελίδα <https://mysmartcity.online>, κατεβάστε την εφαρμογή μέσω του QR Code και εγκαταστήστε τη σε κινητό τηλέφωνο με λειτουργικό σύστημα ANDROID.

Χρησιμοποιήστε την εφαρμογή και στη συνέχεια αν επιθυμείτε συμπληρώστε το ερωτηματολόγιο μέτρησης ευχρηστίας της εφαρμογής.

Η Βαθμολόγηση διενεργείται στο πλαίσιο της διπλωματικής εργασίας του φοιτητή Καφάλη Ανδρέα στο ΠΜΣ Ψηφιακές Εφαρμογές και Καινοτομία του Τμήματος Πληροφορικής του Ιονίου Πανεπιστημίου. Επισημαίνεται ότι το ερωτηματολόγιο και η επεξεργασία των απαντήσεων είναι ανώνυμα και πραγματοποιούνται από τον φοιτητή αποκλειστικά στο πλαίσιο της διπλωματικής του εργασίας.

antykaps@gmail.com [Εναλλαγή λογαριασμού](#)

* Απαιτείται

Διεύθυνση ηλεκτρονικού ταχυδρομείου *

Η διεύθυνσή σας ηλεκτρονικού ταχυδρομεί...

Νομίζω ότι θα ήθελα να χρησιμοποιώ συχνά αυτό το σύστημα. *

1 2 3 4 5

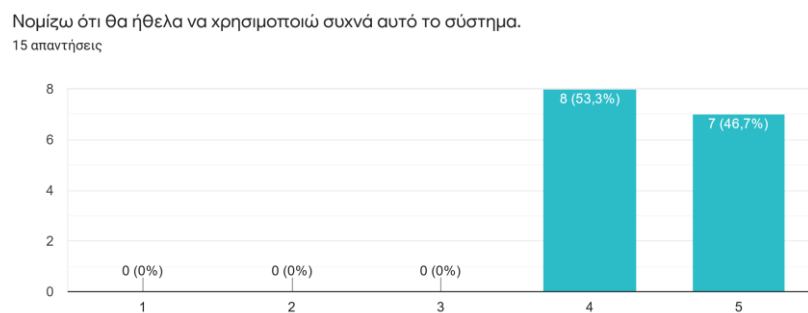
Διαφωνώ έντονα ☐ ☐ ☐ ☐ ☐ Συμφωνώ απόλυτα

Βρήκα το σύστημα χωρίς λόγο πολύπλοκο. *

Η έρευνα διενεργήθηκε στο χρονικό διάστημα από 11/02/2021- 18/02/2021.

Το σύνολο των απαντήσεων που λάβαμε ήταν 16.

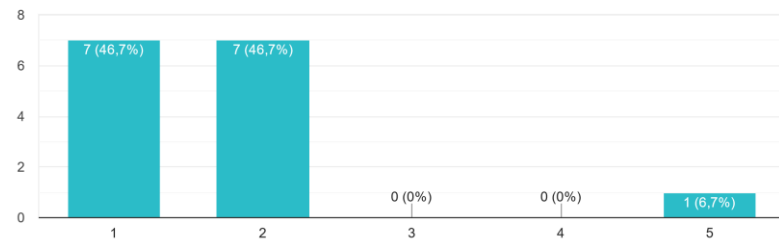
Οι κατανομή των απαντήσεων που λάβαμε είναι:



Εικόνα 111 Ερώτηση 1

Βρήκα το σύστημα χωρίς λόγο πολύπλοκο.

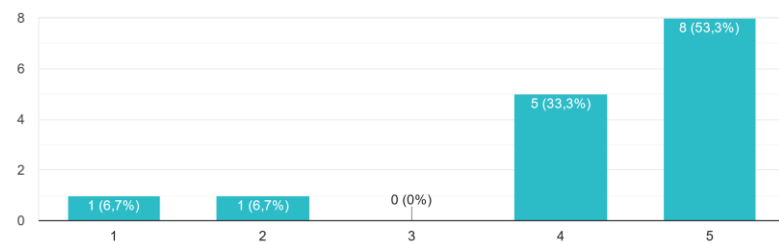
15 απαντήσεις



Εικόνα 112 Ερώτηση 2

Νόμιζα ότι το σύστημα ήταν εύκολο στη χρήση.

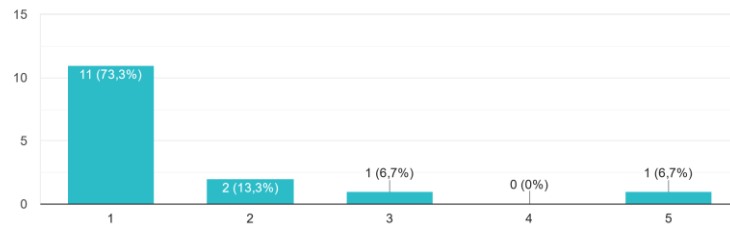
15 απαντήσεις



Εικόνα 113 Ερώτηση 3

Πιστεύω ότι θα χρειάζομαι την υποστήριξη ενός τεχνικού προσώπου για να μπορέσω να χρησιμοποιήσω αυτό το σύστημα

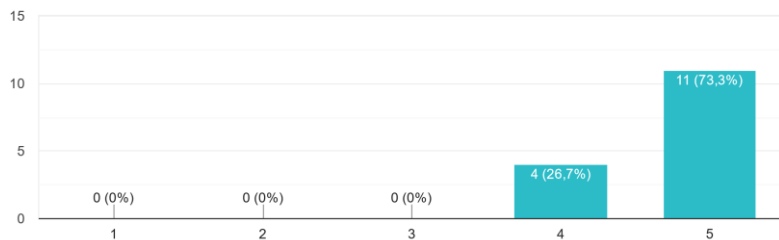
15 απαντήσεις



Εικόνα 114 Ερώτηση 4

Βρήκα ότι οι διάφορες λειτουργίες σε αυτό το σύστημα ήταν καλά ενσωματωμένες.

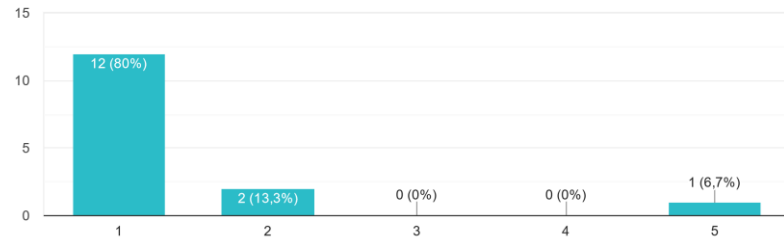
15 απαντήσεις



Εικόνα 115 Ερώτηση 5

Νόμιζα ότι υπήρχε υπερβολική ασυνέπεια σε αυτό το σύστημα.

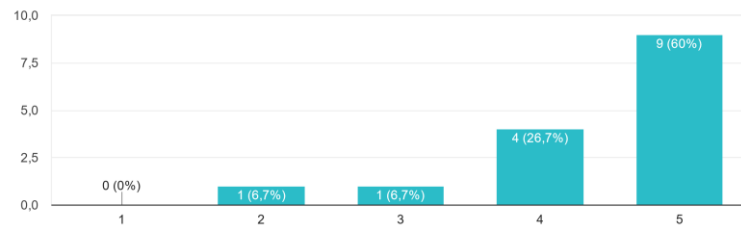
15 απαντήσεις



Εικόνα 116 Ερώτηση 6

Φαντάζομαι ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν αυτό το σύστημα πολύ γρήγορα.

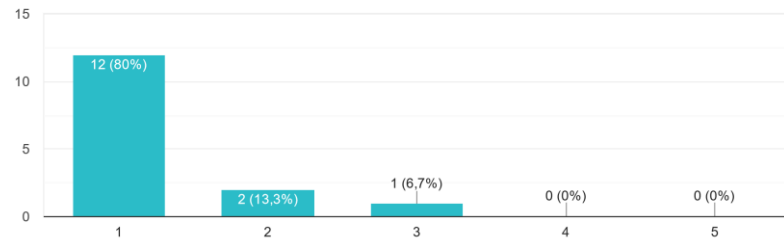
15 απαντήσεις



Εικόνα 117 Ερώτηση 7

Βρήκα το σύστημα πολύ δύσκαμπτο να χρησιμοποιηθεί.

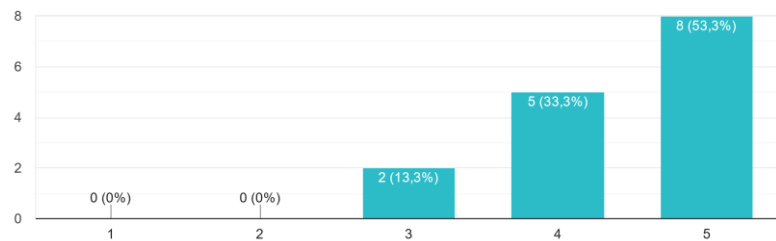
15 απαντήσεις



Εικόνα 118 Ερώτηση 8

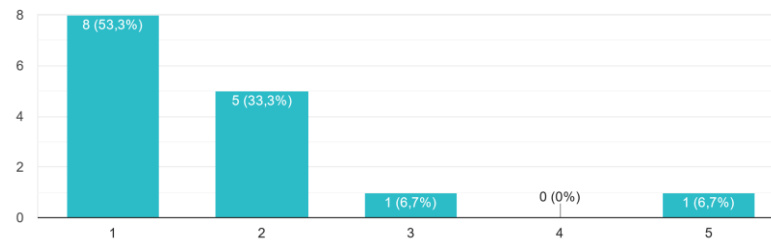
Ένιωσα πολύ σίγουρος χρησιμοποιώντας το σύστημα.

15 απαντήσεις



Εικόνα 119 Ερώτηση 9

Έπρεπε να μάθω πολλά πράγματα πριν προχωρήσω με αυτό το σύστημα
15 απαντήσεις



Εικόνα 120 Ερώτηση 10

8.2. Αποτελέσματα αξιολόγησης

Σύμφωνα με τον μαθηματικό τρόπο υπολογισμού του βαθμού ευχρηστίας SUS της εφαρμογής από τον παρακάτω πίνακα φαίνεται ο συνολικός βαθμός ευχρηστίας SUS της εφαρμογής.

	Ερ1	Ερ2	Ερ3	Ερ4	Ερ5	Ερ6	Ερ7	Ερ8	Ερ9	Ερ10	
Συμετέχων1	5	1	5	1	5	1	5	1	5	1	100
Συμετέχων2	5	2	5	1	5	1	5	1	5	5	87,5
Συμετέχων3	4	1	1	1	5	1	4	1	4	1	82,5
Συμετέχων4	5	1	4	1	5	1	5	1	5	1	97,5
Συμετέχων5	4	2	4	3	5	1	4	1	4	2	80
Συμετέχων6	4	2	4	1	4	1	4	3	4	3	75
Συμετέχων7	5	1	5	1	5	5	5	1	5	1	90
Συμετέχων8	5	1	5	1	5	1	5	1	5	2	97,5
Συμετέχων9	4	2	5	1	4	1	4	2	4	2	82,5
Συμετέχων10	4	1	4	1	5	1	5	1	5	1	95
Συμετέχων11	4	2	5	1	4	2	5	1	4	1	87,5
Συμετέχων12	5	5	5	2	5	1	5	1	5	1	87,5
Συμετέχων13	4	2	4	2	5	1	2	2	3	2	72,5
Συμετέχων14	5	1	5	5	5	1	3	1	5	2	82,5
Συμετέχων15	5	2	2	1	5	2	5	1	5	1	87,5
Συμετέχων16	4	1	5	1	4	1	5	1	3	1	90
									SuS Score		87,1875

Πίνακας 4 Βαθμολογία Ευχρηστίας Συστήματος

Ο βαθμός ευχρηστίας της εφαρμογής (SUS) υπολογίστηκε σε 87,18 . Ο Βαθμός αυτός είναι αρκετά υψηλός, και υποδηλώνει ότι η εφαρμογή «MySmartCity» είναι εύχρηστη , λειτουργική και δεν δημιουργεί ιδιαίτερα προβλήματα στους χρήστες.

9. Συμπεράσματα

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα συμπεράσματα που προέκυψαν με την ολοκλήρωση της παρούσας διπλωματικής εργασίας.

Η ανάπτυξη εφαρμογών σε Android λειτουργικά μας πρόσφερε πολλαπλά οφέλη. Αναπτυχθήκαν δεξιότητες σε σύγχρονα εργαλεία προγραμματισμού όπως το Kodular , AppInventor , Google Maps Api , php , JavaScript , Wordpress και χρησιμοποιήθηκαν Cloud τεχνολογίες αποθήκευσης δεδομένων όπως η Firebase . Επίσης μελετήθηκαν τα βασικά χαρακτηριστικά της τεχνολογίας IoT , η αρχιτεκτονική και τα πρωτόκολλα επικοινωνίας των συσκευών αυτών καθώς και τα πλεονεκτήματα που προσφέρουν στην υλοποίηση λύσεων «έξυπνων πόλεων».

Από την πλευρά του προγραμματισμού , η ανάπτυξη της εφαρμογής μέσω της πλατφόρμας KODULAR αν και αρχικά φάνηκε μια εύκολη διαδικασία μιας και η πλατφόρμα έχει ενσωματώσει πάρα πολλά στοιχεία και πλέον υπάρχουν πάρα πολλά extensions για αυτή, στην πορεία της υλοποίησης της αντιμετωπίσαμε κάποια προβλήματα που στην πλειονότητά τους επιλύθηκαν όπως, ότι η εφαρμογή δεν «προλάβει» να διαβάσει τις διαστάσεις της οθόνης κατά τον σχεδιασμό των μενού αλλά επιλύθηκε με την χρήση του στοιχείου clock με στόχο την εισαγωγή καθυστέρησης. Επίσης η δημιουργία Custom Markers δεν έγινε εφικτή λόγω μη υποστήριξης της λειτουργίας αυτής από τις βιβλιοθήκες του Kodular.

Θα μπορούσε ακόμη να θεωρήσει κάποιος ως αρνητικό στοιχείο το γεγονός ότι η εφαρμογή δεν «τρέχει» σε συσκευές με λειτουργικό IOS . Αυτό δυστυχώς συμβαίνει διότι η πλατφόρμα KODULAR δεν το υποστηρίζει .

Παρόλα αυτά τα μικρά προβλήματα που εμφανίστηκαν κατά την υλοποίηση της εργασίας , τα οποία όμως αντιμετωπίστηκαν με επιτυχία , όπως προκύπτει και από την αξιολόγηση των χρηστών η «MySmartCity» κατάφερε να γίνει μια εύχρηστη , λειτουργική εφαρμογή η οποία αφενός δεν δημιουργεί ιδιαίτερα προβλήματα στους χρήστες , και αφετέρου με την χρήση της συμβάλλει ουσιαστικά στην ενίσχυση της βιωσιμότητας, αλλά και αποδοτικότητας του αστικού περιβάλλοντος, μέσα από τη ενεργή συμμετοχή των πολιτών, των δημοτικών αρχών και των τοπικών επιχειρήσεων στην καθημερινή ζωή στην πόλη.

Μέσα από την εκπόνηση της εργασίας κατανοήθηκαν έννοιες όπως « έξυπνη πόλη » , το Διαδίκτυο των Πραγμάτων καθώς και η σημασία της χρήσης των έξυπνων συ-

σκευών (smartphones) και των εφαρμογών τους στην διαμόρφωση «έξυπνων πόλεων» και «έξυπνων πολιτών».

9.1. Μελλοντικές επεκτάσεις

Βασική μελλοντική επέκταση είναι η δυνατότητα εκτέλεσης της εφαρμογής σε λειτουργικό . Αυτό είναι εφικτό να επιλυθεί χρησιμοποιώντας το Thinkable , μια paid πλατφόρμα ανάπτυξης εφαρμογών Android η οποία είναι συμβατή με τον block κώδικα του Kodular η οποία όμως έχει την δυνατότητα να παράγει εκτελέσιμες εφαρμογές για ANDROID αλλά και για IOS συσκευές.

Όπως έχουμε ήδη αναφέρει η τεράστια διάδοση των κινητών συσκευών , και η όλο και αυξανόμενη κάλυψη με Internet (Wi-Fi – 3G -5G) δίνουν άπειρες δυνατότητες μελλοντικής ανάπτυξης – βελτίωσης της εφαρμογής μας. Ενδεικτικά θα αναφέρουμε μερικές από αυτές:

Αναπτύσσοντας ένα δίκτυο αισθητήρων IoT στην πόλη η εφαρμογή θα μπορούσε να απεικονίζει σε πραγματικό χρόνο τις διαθέσιμες θέσεις στάθμευσης στους δρόμους της πόλης.

Σε ελεγχόμενες θέσεις στάθμευσης (με πληρωμή) θα μπορούσε να γίνεται ηλεκτρονική αγορά χρόνου στάθμευσης.

Σε συνεργασία με το αστικό ΚΤΕΛ θα ήταν εφικτή η ενημέρωση των χρηστών για τα δρομολόγια καθώς και την θέση των λεωφορείων καθώς και τον εκτιμώμενο χρόνο άφιξης.

Και πάλι με την βοήθεια συστήματος IoT θα μπορούσε να ενσωματωθεί λειτουργία που θα δείχνει την κίνηση στην πόλη.

Θα μπορούσε ακόμη να υλοποιηθεί ένα σύστημα Έξυπνου φωτισμού. Αισθητήρες τοποθετημένοι στα φωτιστικά σώματα στον δρόμο θα ανιχνεύουν την ένταση του φυσικού φωτός και την παρουσία ανθρώπων ή αυτοκινήτων για να προσαρμόσουν το φωτισμό στους δημόσιους χώρους στο επιθυμητό επίπεδο.

Βιβλιογραφία

- Albino, V., Berardi, U., & Dangelico, R. M. (2015). Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, 22(1), 3–21.
<https://doi.org/10.1080/10630732.2014.942092>
- Anthopoulos, Leonidas. (2019). The smart city of Trikala. 10.1016/B978-0-12-816169-2.00007-9.
- Anthopoulos L. and Vakali A., (2012), Urban Planning and Smart Cities: Interrelations and Reciprocities. In Alvarez, F. et al., (Eds.), *Future Internet Assembly 2012: From Promises to Reality*, 4th FIA book, The Future Internet, Lecture Notes in Computer Science, Volume 7281, 2012, pp 178-189, Springer. Berlin, Heidelberg, online: http://link.springer.com/chapter/10.1007/978-3-642-30241-1_16, [10.10.2016]
- Boyd Cohen, "What exactly is a smart city" Sep. 2012; Accessed 15/12/2021 from:
(<https://www.fastcompany.com/1680538/what-exactly-is-a-smart-city>)
- Caragliu, A., Del Bo, C. and Nijkamp, P. (2009), "Smart Cities in Europe", Serie Research Memoranda 0048, VU University Amsterdam, Faculty of Economics, Business Administration and Econometrics
- Christopoulou, E., Ringas, D., Garofalakis, J. (2014). The Vision of the Sociable Smart City. In: Streitz, N., Markopoulos, P. (eds) *Distributed, Ambient, and Pervasive Interactions. DAPI 2014. Lecture Notes in Computer Science*, vol 8530. Springer, Cham. https://doi.org/10.1007/978-3-319-07788-8_50
- Deakin, M.; Al Waer, H. (2011) "From Intelligent to Smart Cities". *Journal of Intelligent Buildings International: From Intelligent Cities to Smart Cities*. 3 (3).
- Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Pichler-Milanovic, N. and Meijers, E. (2007), *Smart Cities: Ranking of European Medium-Sized Cities*, Vienna, Austria: Centre of Regional Science (SRF), Vienna University of Technology.
- Johnny Ryan. 2010. *A History of the Internet and the Digital Future* (1st. ed.). Reaktion Books, London,
- McLaren, D., and Agyeman, J. 2015. *Sharing Cities: A Case for Truly Smart and Sustainable Cities*. MIT Press
- Mora, Luca & Bolici, Roberto. (2017). *How to Become a Smart City: Learning from Amsterdam*. 10.1007/978-3-319-44899-2_15.
- Schaffers, H., Komninos, N., & Pallot, M., (2012), *Fireball White Paper: Smart Cities as Innovation Ecosystems Sustained by the Future Internet*, Fireball Community
- Tripathy, B. K., Anuradha, J., & Taylor & Francis (Londyn). (2018). *Internet of things (IoT): Technologies, applications, challenges, and solutions*. Boca Raton, FL: CRC Press - Taylor & Francis Group.

Van Dijk A., (2015), 'Smart Cities, How rapid advances in technology are reshaping our economy and society' [online], Deloitte, Accessed 15/12/2021 from:

(<https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/public-sector/deloitte-nl-ps-smart-cities-report.pdf>)

Vinod Kumar T.D. Ed. 2017. Smart economy in smart cities. Springer

Βογιατζής, Ν., Κολοκοτρώνης Δ., (2017) «Συλλογή, μελέτη και αξιολόγηση δράσεων και εφαρμογών «έξυπνων πόλεων. Αποτύπωση της υφιστάμενης κατάστασης στους Δήμους της Ελλάδας» Εκδόσεις Πανεπιστήμιο Αιγαίου Ανακτήθηκε 10/1/2022 από

(https://hellenicus.lib.aegean.gr/bitstream/handle/11610/17392/Διπλωματική_Έξυπνες_Πόλεις_Βογιατζής_Κολοκοτρώνης_2017.pdf).

Καμπουράκης Β (2020) Ανακτήθηκε 10/1/2022 από

(https://maredu.hcg.gr/modules/document/file.php/MAK265/Ηλεκτρονικοί_Υπολογιστές/Internet_of_Things.pdf)

Μυλωνοπούλου Ζ (2018) «Η έννοια της «Έξυπνης Πόλης» ως προσέγγιση της σύγχρονης αστικής ανάπτυξης μέσα από τη μελέτη επιλεγμένων διεθνών και ελληνικών παραδειγμάτων», Ανακτήθηκε 10/1/2022 από (https://amitos.library.uop.gr/xmlui/bitstream/handle/123456789/3988/595-2017_ΜΥΛΩΝΟΠΟΥΛΟΥ_ΖΩΗ.pdf)

Μαλούτα Θ., Μίχου Ε., (2019) «Το Διαδίκτυο των Πραγμάτων (IoT) Συγκριτική παρουσίαση και αξιολόγηση των σημαντικότερων τεχνολογιών υλοποίησης», » Ανακτήθηκε 10/1/2022 από (<https://pergamon.lib.uoa.gr/uoa/dl/frontend/file/lib/default/data/2866409/>)

Στρουτσιώτη Α, Λαγού Ο, Τζανή Α., (2019) «Εφαρμογές Έξυπνης Πόλης» Ανακτήθηκε 10/1/2022 από (http://repository.library.teimes.gr/xmlui/bitstream/handle/123456789/7651/ΕΦΑΡΜΟΓΕΣ_ΕΞΥΠΝΗΣ_ΠΟΛΗΣ..pdf).

Τζιούφα Π., (201) «INTERNET OF THINGS-RFID ΚΑΙ ΠΡΟΣΩΠΙΚΑ ΔΕΔΟΜΕΝΑ: ΘΕΜΑΤΑ ΑΣΦΑΛΕΙΑΣ ΚΑΙ ΑΠΟΡΡΗΤΟΥ ΣΤΟ ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ (IoT)» Ανακτήθηκε 10/2/2022 από (<https://core.ac.uk/download/pdf/292408725.pdf>)

Amsterdam SmartCity, Ανακτήθηκε 10/2/2022 από (<https://smartercommunities.media/truly-smart-city-five-lessons-amsterdam/>)

AM2302 / DHT22 Datasheet, Ανακτήθηκε 3/3/2022 από

(<https://www.electroschematics.com/am2302-dht22-datasheet/>)

Arduino dht22 library Ανακτήθηκε 10/2/2022 από

(<https://www.arduino.cc/reference/en/libraries/dht-sensor-library-for-esp/>)

Arduino IDE, Ανακτήθηκε 3/3/2022 από (<https://www.arduino.cc/en/software>)

Copenhagen Smartups, 2020 ,Ανακτήθηκε 10/2/2022 από

(<https://www.smartecocity.com/category/eucities/dn-copenhagen/>)

Derivation of centripetal acceleration using polar coordinates Ανακτήθηκε 2/2/2022 από

<https://thecuriousastronomer.wordpress.com/2014/09/15/derivation-of-centripetal-acceleration-using-polar-coordinates/>

ESP8266 Solar Weather Station , Ανακτήθηκε 2/2/2022 από (

<https://www.hackster.io/Tiobel/esp8266-solar-weather-station-1ea3ec>)

ESP8266 for IoT: A Complete Guide, Ανακτήθηκε 2/2/2022 (από [https://www.nabto.com/esp8266-](https://www.nabto.com/esp8266-for-iot-complete-guide/)

[for-iot-complete-guide/](https://www.nabto.com/esp8266-for-iot-complete-guide/))

Elementor WordPress Website Builder ,Ανακτήθηκε 5/12/2021 από (<https://elementor.com/>)

Firebase Realtime Database Arduino Library for ESP8266, Ανακτήθηκε 10/2/2022 από

(<https://github.com/mobizt/Firebase-ESP8266>)

Google Cloud Platform, Ανακτήθηκε 5/12/2021 από (<https://console.cloud.google.com/>)

Google Firebase, Ανακτήθηκε 5/12/2021 από (<https://console.firebase.google.com/>)

Google Map Platform, Geogoding , Ανακτήθηκε 5/12/2021 από

(<https://developers.google.com/maps/documentation/geocoding/start>)

HC-SR501 PIR Motion Sensor Ανακτήθηκε 5/3/2022 από (<https://www.makerguides.com/hc-sr501-arduino-tutorial/>)

Kodular Platform, Ανακτήθηκε 5/12/2021 από(<https://creator.kodular.io/>)

Lipo Charger DFR0264, Ανακτήθηκε 3/3/2022 από (

https://wiki.dfrobot.com/Solar_Charger_SKU_DFR0264)

NodeMCU ESP8266, Ανακτήθηκε 2/2/2022 από ([https://components101.com/development-](https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet)

[boards/nodemcu-esp8266-pinout-features-and-datasheet](https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet))

OneSignal, Ανακτήθηκε 5/12/2021 από (<https://onesignal.com/>)

Smart City Series: the Barcelona Experience, Ανακτήθηκε 10/2/2022 από (<https://www.e-zigurat.com/blog/en/smart-city-barcelona-experience>)

Smartphone Sensors Ανακτήθηκε 10/1/2022 από
(<https://github.com/Apt3kStudio/Phone/wiki/Sensors-on-Smartphones>)

Solar Panel 6V 1W, Ανακτήθηκε 3/3/2022 από(<https://topelectronics.gr/electronics/solar/6v-200ma-1.1w-mini-solar-panel-112x84mm/>)

System Usability Scale (SUS) Ανακτήθηκε 15/2/2022 από (<https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>)

Wordpress Blog Tool, Publishing Platform, and CMS Ανακτήθηκε 5/12/2021 από
(<https://wordpress.com>)

Διαδίκτυο των πραγμάτων Ανακτήθηκε 15/2/2022 από
(https://el.wikipedia.org/wiki/Διαδίκτυο_των_πραγμάτων)

Δήμος Ιωαννιτών Ανακτήθηκε 10/2/2022 από (www.ioannina.gr)

Η εφαρμογή Novoville Ανακτήθηκε 15/1/2022 από
<https://play.google.com/store/apps/details?id=com.novoville&hl=el&gl=US>)

Η εφαρμογή SenceCity Ανακτήθηκε 10/1/2022 από
(<https://play.google.com/store/apps/details?id=gr.upatras.ece.nam.sensecity&hl=el&gl=US>)

Εισαγωγή στα προγράμματα έξυπνων πόλεων μέσω βασικών τεχνολογιών ανάπτυξης ιστού. Ανακτήθηκε 10/2/2022 από([https://discover-project.eu/uploads/text/3/documents_el/7.Εισαγωγή στα προγράμματα έξυπνων πόλεων μέσω βασικών τεχνολογιών ανάπτυξης ιστού.pdf](https://discover-project.eu/uploads/text/3/documents_el/7.Εισαγωγή_στα_προγράμματα_έξυπνων_πόλεων_μέσω_βασικών_τεχνολογιών_ανάπτυξης_ιστού.pdf))

Η ΙΣΤΟΡΙΑ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ : ΧΡΟΝΟΛΟΓΙΚΗ ΕΞΕΛΙΞΗ Ανακτήθηκε 10/2/2022 από
(<https://blogs.e-me.edu.gr/hive-pli-roforikigxanthi/2020/05/27/η-ιστορια-του-διαδικτυου--χρονολο/>)

ΙΩΑΝΝΙΝΑ SMART CITY – Επιχειρησιακό Σχέδιο του Δήμου Ιωαννιτών για την «Έξυπνη Πόλη» Ανακτήθηκε 10/2/2022 από (<http://www.diavouleusi.eu/diabouleyseis/ioannina-smart-city>)

Κατανομή των (IoT) σε όλο τον κόσμο Ανακτήθηκε 10/1/2022 από (<https://www.statista.com/statistics/869335/world-internet-of-things-projects-by-segment-enterprise/>)

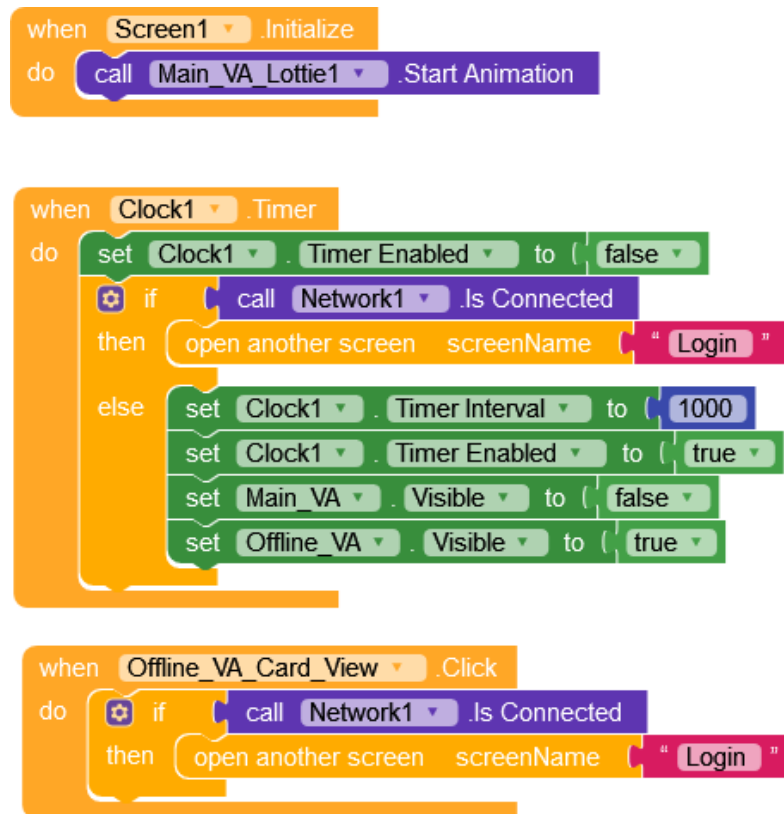
Πρωτόκολλα επικοινωνίας IoT Ανακτήθηκε 10/1/2022 από (<https://hashstudioz.com/blog/top-iot-communication-protocols-2020/>)

Τι είναι το IoT και πώς θα αλλάξει τη ζωή σου Ανακτήθηκε 10/1/2022 από(<https://www.cosmotesmartliving.gr/τι-είναι-το-iot-και-πώς-θα-αλλάξει-τη-ζωή-σο/>)

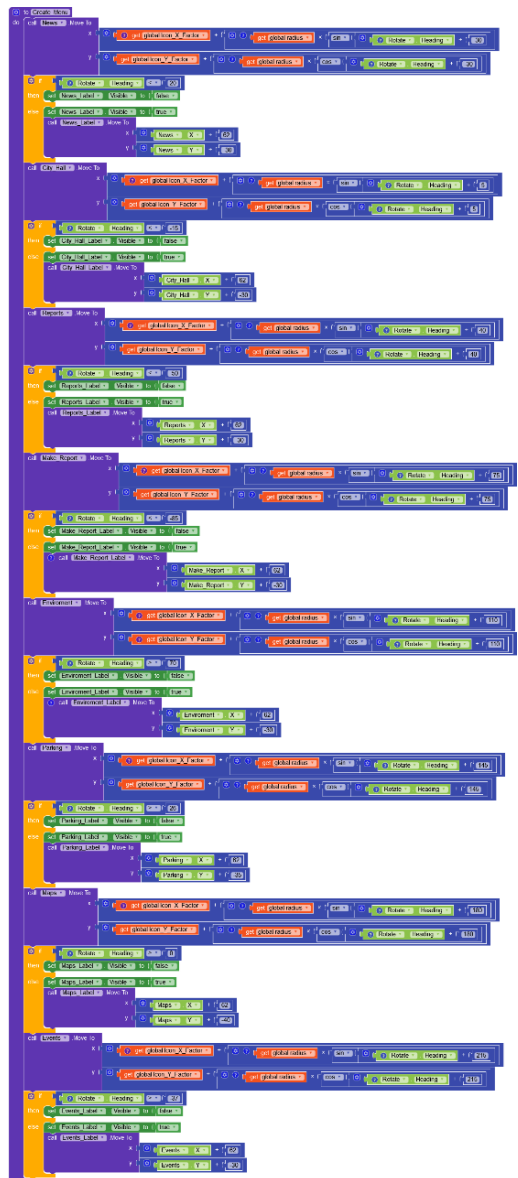
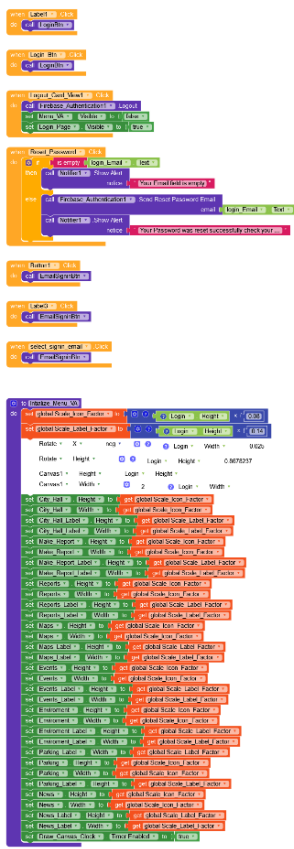
Τρίκαλα - Σύστημα Έξυπνης Στάθμευσης Ανακτήθηκε 10/2/2022 από (<https://trikalacity.gr/systima-exypnis-stathmeysis/>)

Τρίκαλα -Σύστημα παρακολούθησης περιβαλλοντικών συνθηκών Ανακτήθηκε 10/2/2022 από (<https://trikalacity.gr/systima-parakolythisis-perivallontikon-synthikon/>)

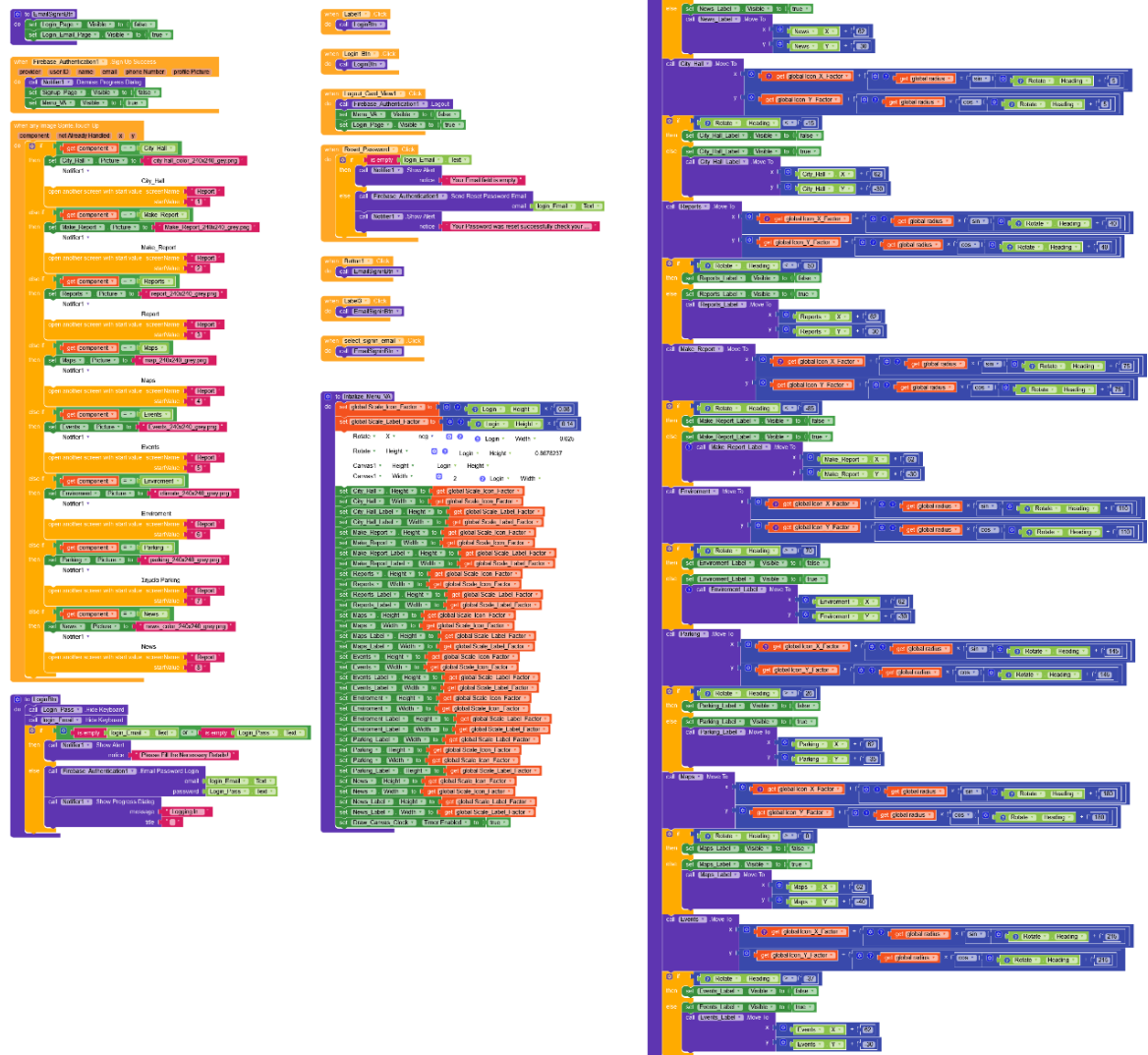
Παράρτημα Α: «Ο Block Κώδικας της Mobile εφαρμογής»



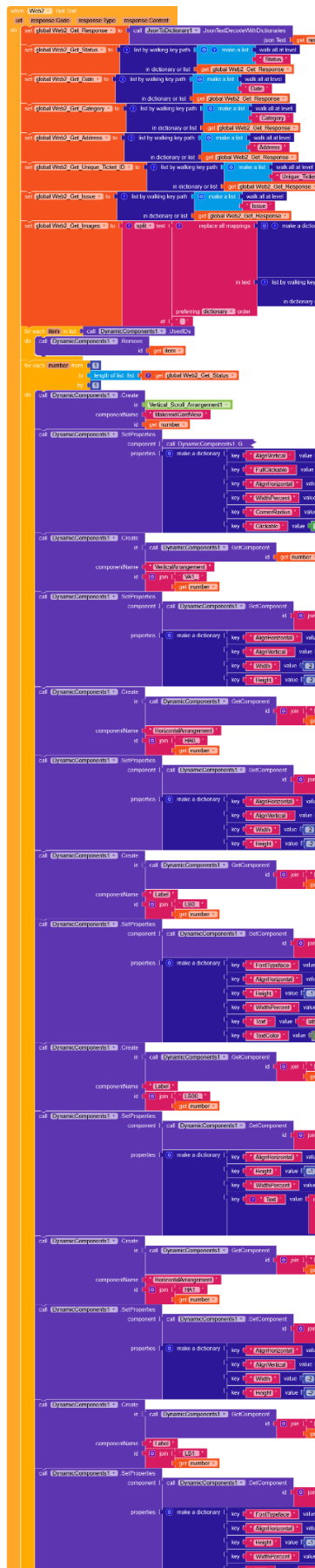
Block κώδικας της Screen 1



Block Κώδικας Login Screen (b)



Block Κώδικας Report Screen (b)



Block Κώδικας Report Screen(c)

Παράρτημα Β: «Κώδικας Web εφαρμογής»

```
<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<title>SmartCity Managment System</title>
<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js" integri-
ty="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" cros-
sorigin="anonymous"></script>
</head>
<body>
<div class="container" style="margin-top: 50px;">
<table class="table table-bordered">
<tr>
<th>Ημερομηνία</th>
<th>Κατηγορία</th>
<th>Περιγραφή</th>
<th>Τοποθεσία</th>
<th>Κατάσταση</th>
<th>Ενέργειες</th>
</tr>
<tbody id="tbody">
</tbody>
</table>
</div>
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.14.6/firebase-app.js"></script>
<!-- TODO: Add SDKs for Firebase products that you want to use
https://firebase.google.com/docs/web/setup#available-libraries -->
<script>
src="https://www.gstatic.com/firebasejs/7.14.6/firebase-analytics.js"></script>
<script>
src="https://www.gstatic.com/firebasejs/7.14.6/firebase-database.js"></script>
<script>
// Initialize Firebase
var config = {
apiKey: "AIzassSyBIros6H-Ao1jU5fdCss6s5joaXVdYdz6ccKgTS2w",
authDomain: "smart-city-295ac.firebaseio.com",
databaseURL: "https://smart-city-295ac-default-rtdb.firebaseio.com",
projectId: "smart-city-295ac",
storageBucket: "smart-city-295ac.appspot.com",
messagingSenderId: "52993427272",
appId: "1:52993427272:web:25dee83c31ef9734ee957d",
measurementId: "G-328BDXCK4D"
};
firebase.initializeApp(config);
firebase.analytics();
var database = firebase.database();
var lastIndex = 0;
// Get Data
firebase.database().ref('Issues/').limitToLast(10).on('value', function (snapshot) {
var value = snapshot.val();
```

```

var htmls = [];
$.each(value, function (index, value) {
if (value) {
htmls.push('<tr>\
<td>' + (value.Date).replace (/"/g, '"') + '</td>\
<td>' + (value.Category).replace (/"/g, '"') + '</td>\
<td>' + (value.Issue).replace (/"/g, '"') + '</td>\
<td>' + (value.Address).replace (/"/g, '"') + '</td>\
<td>' + (value.Status).replace (/"/g, '"') + '</td>\
<td>' + (value.Actions).replace (/"/g, '"') + '</td>\
</tr>');
}
lastIndex = index;
});
$('#tbody').html(htmls);
$('#submitUser').removeClass('disabled');
});
</script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
</body>
</html>

```

Last_10_Issues.html

```

<!DOCTYPE html>
<html>
<head>

<style>
#map {
height: 800px;
}

html, body {
height: 100%;
margin: 0;
padding: 0;
}

#over_map {
position: absolute;
top: 10px;
left: 89%;
z-index: 99;
background-color: #ccffcc;
padding: 10px;
}
</style>
</head>
<body>
<div id="map"></div>
<div id="over_map">
<div>
<span>Αναφορές </span><span id="events">0</span>

```

```

</div>
</div>
<!-- jQuery CDN -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<!-- Firebase -->
<script src="https://www.gstatic.com/firebasejs/4.12.1/firebase.js"></script>
<script>
  // Replace your Configuration here..
  var config = {
    apiKey: "AlstopzaSyBIro6H-Ao1jU5fadC5joaXgjlVdYdz6KgTS2w",
    authDomain: "smart-city-295ac.firebaseio.com",
    databaseURL: "https://smart-city-295ac-default-rtdb.firebaseio.com",
    projectId: "smart-city-295ac",
    storageBucket: "smart-city-295ac.appspot.com",
    messagingSenderId: "52993427272",
    appId: "1:52993427272:web:25dee83c31ef9734ee957d",
    measurementId: "G-328BDXCK4D"
  };
  firebase.initializeApp(config);
</script>
<script>
  // counter for online events...
  var events_count = 0;
  // markers array to store all the markers
  var markers = [];
  var map;
  function initMap() { // Google Map Initialization...
    map = new google.maps.Map(document.getElementById('map'), {
      zoom: 12,
      center: new google.maps.LatLng(39.61430338869748, 20.852007662975016),
      mapTypeId: 'roadmap',
      styles: [
        {
          "featureType": "poi",
          "stylers": [
            { "visibility": "off" }
          ]
        }
      ]
    });
    // This Function will create a Marker icon with angle and add/display that marker on the map
    function Add_Event(data) {
      var tmpStr = data.val().Coordinates;
      const string = tmpStr.substring(3, tmpStr.length-3);
      const regex = /\s*([^\s:]+)\s*:\s*([^\s:]+)/g;
      const res = Object.fromEntries(Array.from(string.matchAll(regex), (m) => [m[1], m[2]]));
      var marker_color = res.color;
      if (marker_color==0) {
        var fill_color = "#FF0000";
      }
      else if (marker_color==60) {
        var fill_color = "#FFFF00";
      }
      else if (marker_color==120) {
        var fill_color = "#00FF00";
      }
      var iconBase = 'https://maps.google.com/mapfiles/kml/shapes/';
    }
  }
}

```

```

    var icon = {path: 'M 0,0 C -2,-20 -10,-22 -10,-30 A 10,10 0 1,1 10,-30 C 10,-22 2,-20 0,0 z M -2,-
30 a 2,2 0 1,1 4,0 2,2 0 1,1 -4,0',
    scale: 1,
    fillColor: fill_color, //<-- Marker Color, you can change it
    fillOpacity: 1,
    strokeWeight: 1,
    anchor: new google.maps.Point(0, 5),
    rotation: data.val().angle //<-- Marker angle
  };
  var uluru = { lat: parseFloat(res.lat), lng: parseFloat(res.lng) };
  var marker = new google.maps.Marker({
    position: uluru,
    icon: icon,
    map: map,
    title: res.title
  });
  markers[data.key] = marker; // add marker in the markers array...
  document.getElementById("events").innerHTML = events_count;
}
// get firebase database reference...
var events_Ref = firebase.database().ref('Issues/');
// this event will be triggered when a new object will be added in the database...
events_Ref.on('child_added', function (data) {
  events_count++;
  Add_Event(data);--
});
// this event will be triggered on location change of any Marker...
events_Ref.on('child_changed', function (data) {
  markers[data.key].setMap(null);
  Add_Event(data);
});
// If any event goes offline then this event will get triggered and we'll remove the marker of that
Marker...
events_Ref.on('child_removed', function (data) {
  markers[data.key].setMap(null);
  events_count--;
  document.getElementById("events").innerHTML = events_count;
});
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIsszdddaSyBdddCiwcdpjwe7qddVU4yU-
PWqxyUC-mTpQCrfGcnYs&callback=initMap">
</script>
</body>
</html>

```

Issues_Map.html

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      #map {
        height: 640px;
      }
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
      #over_map {
        font-size: 20px;
        position: absolute;
        top: 10px;
        left: 80%;
        z-index: 99;
        background-color: #ccffcc;
        padding: 2px;
      }
    </style>
  </head>
  <body>
    <div id="map"></div>
    <div id="over_map">
      <div>
        <div align="center"><span>Διαθέσιμα Σημεία Parking <br>
        </span><span id="events">0</span>
        </div>
      </div>
    </div>
    <!-- jQuery CDN -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <!-- Firebase -->
    <script src="https://www.gstatic.com/firebasejs/4.12.1/firebase.js"></script>
    <script>
      // Replace your Configuration here..
      var config = {
        apiKey: "AIzaSddfhjkmyBlRo6H-Ao1jU5fdgjkdddddC5joaXdddVdYdzddd6lllKgTS2w",
        authDomain: "smart-city-295ac.firebaseio.com",
        databaseURL: "https://smart-city-295ac.firebaseio.com",
        projectId: "smart-city-295ac",
        storageBucket: "smart-city-295ac.appspot.com",
        messagingSenderId: "52993427272",
        appId: "1:52993427272:web:25dee83c31ef9734ee957d",
        measurementId: "G-328BDXCK4D"
      };
      firebase.initializeApp(config);
    </script>
    <script>
      // counter for online events...
      var events_count = 0;
      // markers array to store all the markers
      var markers = [];
      var map;
      function initMap() { // Google Map Initialization...
        map = new google.maps.Map(document.getElementById('map'), {
          zoom: 16,

```

```

        center: new google.maps.LatLng(39.664134078368036, 20.851975476635868),
        mapTypeId: 'roadmap',
        styles: [
            {
                "featureType": "poi",
                "stylers": [
                    { "visibility": "off" }
                ]
            }
        ]
    });
}
// This Function will create a Marker icon with angle and add/display that marker on the map
function Add_Event(data) {
    var Parking_Title = data.val().Parking_Title;
    const contentString =
        '<div id="content">' +
        '<div id="siteNotice">' +
        "</div>" +
        '<h1 id="firstHeading" class="firstHeading">Uluru</h1>' +
        '<div id="bodyContent">' +
        "<p><b>Uluru</b>, also referred to as <b>Ayers Rock</b>, is a large " +
        "sandstone rock formation in the southern part of the " +
        +
        "Aboriginal people of the area. It has many springs, waterholes, " +
        "rock caves and ancient paintings. Uluru is listed as a World " +
        "Heritage Site.</p>" +
        '<p>Attribution: Uluru, <a'
        href="https://en.wikipedia.org/w/index.php?title=Uluru&oldid=297882194">' +
        "https://en.wikipedia.org/w/index.php?title=Uluru</a> " +
        "(last visited June 22, 2009).</p>" +
        "</div>" +
        "</div>";
    const infowindow = new google.maps.InfoWindow({
        content: contentString,
    });
    var uluru = { lat: JSON.parse(data.val().Lat), lng: JSON.parse(data.val().Long) };

    //var uluru = { lat: parseFloat(res.lat), lng: parseFloat(res.lng) };
    var number = parseInt(data.val().Free_Places) + " θέσεις";
    const MAP_MARKER = 'M12 2C8.13 2 5 5.13 5 9c0 5.25 7 13 7 13s7-7.75 7-13c0-3.87-3.13-7-7-7zm0 9.5c-1.38 0-2.5-1.12-2.5-2.5s1.12-2.5 2.5-2.5 2.5-1.12 2.5-2.5 2.5-1.12 2.5-2.5 2.5z';
    var marker = new google.maps.Marker({
        position: uluru,
        map: map,
        animation: google.maps.Animation.DROP,
        label: {
            text: "" + number,
            color: "blue",
            fontSize: "20px",
            fontWeight: "bold"
        },
        title: "Διαθέσιμες " + parseInt(data.val().Free_Places) + " θέσεις στάθμευσης",
        icon: {
            path: MAP_MARKER,
            scale: 2,
            fillColor: 'blue',
            fillOpacity: 1,
            strokeColor: '#AB47BC',

```

```

        anchor: { x: 12, y: 24 },
        labelOrigin: new google.maps.Point(12, -5)
    },
    });
    marker.addListener("click", () => {
        infoWindow.open({
            anchor: marker,
            map,
            shouldFocus: false,
        });
    });
    markers[data.key] = marker; // add marker in the markers array...
    document.getElementById("events").innerHTML = events_count;
}
// get firebase database reference...
var events_Ref = firebase.database().ref('Parking/');
// this event will be triggered when a new object will be added in the database...
events_Ref.on('child_added', function (data) {
    events_count++;
    Add_Event(data);
});
// this event will be triggered on location change of any Marker...
events_Ref.on('child_changed', function (data) {
    markers[data.key].setMap(null);
    Add_Event(data);
});
// If any event goes offline then this event will get triggered and we'll remove the marker of that
Marker...
events_Ref.on('child_removed', function (data) {
    markers[data.key].setMap(null);
    events_count--;
    document.getElementById("events").innerHTML = events_count;
});
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AddIzafSyBfjvnmoawzzCipjwess7qVU4yh
hU-PWhhhqxkkkyUC-mTpQCrCnYs&callback=initMap">
</script>
</body>
</html>

```

Parking.html

```

<?php
/**
 * Custom code antreas
 */
//send post to firebase
function Publish_Post($post_id,$post )
{
    $post_id = $post->ID;

    // from here send the data to firebase
    //$Event_date = $post->Event_Date;
    $Event_time_date = $post->Event_Time_Date;

```

```

$Event_Content =$post->post_content;
$Event_Title = $post->post_title;
$Place = $post->location;
$Event_thumbnail_id = get_post_thumbnail_id( $post );
$Event_thumbnail_url = wp_get_attachment_image_url( $Event_thumbnail_id, 'post-thumbnail' );
//Read Parking Data
$Lat = $post->lat;
$Long = $post->long;
$Free_Places=$post->free_places;
$Parking_thumbnail_id = get_post_thumbnail_id( $post );
$Parking_thumbnail_url = wp_get_attachment_image_url( $Parking_thumbnail_id, 'post-thumbnail' );
// Town-Hill
$Mayor_Title =$post->post_title;
$Mayor_CV = $post->post_content;
$Mayor_fname =$post-> mayor_fname;
$Mayor_lname =$post-> mayor_lname;
$Mayor_thumbnail_id = get_post_thumbnail_id( $post );
$Mayor_thumbnail_url = wp_get_attachment_image_url( $Mayor_thumbnail_id, 'post-thumbnail' );
$Phone =$post->phone;
$Email = $post->email;
//News
$News_Title =$post->post_title;
$News_Content = $post->post_content;
$News_Date = $post->post_date;
$News_thumbnail_id = get_post_thumbnail_id( $post );
$News_thumbnail_url = wp_get_attachment_image_url( $News_thumbnail_id, 'post-thumbnail' );
// Constants
//$FIREBASE = "https://esp8226-a14e4-default-rtdb.firebaseio.com/";
$FIREBASE = "https://smart-city-295ac-default-rtdb.firebaseio.com/";
// $NODE_DELETE = "/Event/".$post_id.".json";
// $NODE_GET = "Events.json";
// $NODE_PATCH = ".json";
$NODE_PUT_Events = "/Events/".$post_id.".json";
$NODE_PUT_Parking = "/Parking/".$post_id.".json";
$NODE_PUT_Town_Hill = "/Town-Hill/".$post_id.".json";
$NODE_PUT_News = "/News/".$post_id.".json";
$category_detail=get_the_category($post->ID);//$post->ID
foreach($category_detail as $cd){
    $Category = $cd->cat_name;
}
// Initialize cURL
$curl = curl_init();
if ($Category == "Events") {
    $data = array(
        "Event_Content" => $Event_Content,
        "Event_Title" => $Event_Title ,
        "Image" => $Event_thumbnail_url,
        "Date_Time_Place" => $Event_time_date." ".$Place
    );
}
// JSON encoded
$json = json_encode( $data );
// Create
curl_setopt( $curl, CURLOPT_URL, $FIREBASE . $NODE_PUT_Events );
curl_setopt( $curl, CURLOPT_CUSTOMREQUEST, "PUT" );
curl_setopt( $curl, CURLOPT_POSTFIELDS, $json );
}
if ($Category == "Parking") {
    $data = array(

```

```

        "Parking_Content" => $Event_Content,
        "Parking_Title" => $Event_Title ,
        "Parking_Image" => $Parking_thumbnail_url,
        "Lat" => $Lat,
        "Long" => $Long,
        "Free_Places" => $Free_Places
    );
    // JSON encoded
    $json = json_encode( $data );

    // Create
    curl_setopt( $curl, CURLOPT_URL, $FIREBASE . $NODE_PUT_Parking );
    curl_setopt( $curl, CURLOPT_CUSTOMREQUEST, "PUT" );
    curl_setopt( $curl, CURLOPT_POSTFIELDS, $json );
    $data = array(
        "Mayor_Title" => $Mayor_Title,
    if ($Category == "Town-Hill") {

        "Mayor_Full_Name" => $Mayor_fname." ".$Mayor_lname,
        "Mayor_CV" => $Mayor_CV,
        "Mayor_Image" => $Mayor_thumbnail_url,
        "Mayor_Image" => $Mayor_thumbnail_url,
        "Phone" => $Phone,
        "Email" => $Email,
    );
    // JSON encoded
    $json = json_encode( $data );
    // Create
    curl_setopt( $curl, CURLOPT_URL, $FIREBASE . $NODE_PUT_Town_Hill );
    curl_setopt( $curl, CURLOPT_CUSTOMREQUEST, "PUT" );
    curl_setopt( $curl, CURLOPT_POSTFIELDS, $json );
    }
    if ($Category == "News") {
    $data = array(
        "News_Title" => $News_Title,
        "News_Content" => $News_Content,
        "News_Image" => $News_thumbnail_url,
        "News_Date" => $News_Date,
    );
    // JSON encoded
    $json = json_encode( $data );
    // Create
    curl_setopt( $curl, CURLOPT_URL, $FIREBASE . $NODE_PUT_News);
    curl_setopt( $curl, CURLOPT_CUSTOMREQUEST, "PUT" );
    curl_setopt( $curl, CURLOPT_POSTFIELDS, $json );
    }
    // Get return value
    curl_setopt( $curl, CURLOPT_RETURNTRANSFER, true );

    // Make request
    // Close connection
    $response = curl_exec( $curl );
    curl_close( $curl );
    }
    add_action( 'publish_post', 'Publish_Post' , 10, 2 );

?>

```

Create_Post_in_Firebase.php

```

<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<title>SmartCity Managment System</title>
<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js" integri-
ty="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0=" cros-
sorigin="anonymous"></script>
</head>
<body>
<div class="container" style="margin-top: 50px;">
<h4 class="text-center">SmartCity Managment System</h4><br>
<h5>Αναφορές</h5>
<table class="table table-bordered">
<tr>
<th>Ημερομηνία</th>
<th>Κατηγορία</th>
<th>Περιγραφή</th>
<th>Τοποθεσία</th>
<th>Κατάσταση</th>
<th>Ενέργειες</th>
<th width="180" class="text-center">Action</th>
</tr>
<tbody id="tbody">
</tbody>
</table>
</div>
<!-- Update Model -->
<form action="" method="POST" class="users-update-record-model form-horizontal">
<div id="update-modal" data-backdrop="static" data-keyboard="false" class="modal fade" tabin-
dex="-1" role="dialog" aria-labelledby="custom-width-modalLabel"
aria-hidden="true">
<div class="modal-dialog modal-dialog-centered" style="width:55%;">
<div class="modal-content" style="overflow: hidden;">
<div class="modal-header">
<h4 class="modal-title" id="custom-width-modalLabel">Update</h4>
<button type="button" class="close" data-dismiss="modal"
aria-hidden="true">
</button>
</div>
<div class="modal-body" id="updateBody">
</div>
<div class="modal-footer">
<button type="button" class="btn btn-light"
data-dismiss="modal">Close
</button>
<button type="button" class="btn btn-success updateUser">Update
</button>
</div>
</div>
</div>
</div>
</div>
</form>
<!-- Delete Model -->

```

```

<form action="" method="POST" class="users-remove-record-model">
<div id="remove-modal" data-backdrop="static" data-keyboard="false" class="modal fade" tabin-
dex="-1" role="dialog" aria-labelledby="custom-width-modalLabel"
aria-hidden="true" style="display: none;">
<div class="modal-dialog modal-dialog-centered" style="width:55%;">
<div class="modal-content">
<div class="modal-header">
<h4 class="modal-title" id="custom-width-modalLabel">Delete</h4>
<button type="button" class="close remove-data-from-delete-form" data-dismiss="modal" aria-
hidden="true">
</button>
</div>
<div class="modal-body">
<p>Do you want to delete this record?</p>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-default waves-effect remove-data-from-delete-form"
data-dismiss="modal">Close
</button>
<button type="button" class="btn btn-danger waves-effect waves-light deleteRecord">Delete
</button>
</div>
</div>
</div>
</div>
</div>
</form>
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.14.6/firebase-app.js"></script>
<!-- TODO: Add SDKs for Firebase products that you want to use
https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/7.14.6/firebase-analytics.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.14.6/firebase-database.js"></script>
<script>
// Initialize Firebase
var config = {
apiKey: "AIzaSySBldRodd6H-Ao1jUgg5fdC5jojaXVdYkdz6KgTkS2w",
authDomain: "smart-city-295ac.firebaseio.com",
databaseURL: "https://smart-city-295ac-default-rtdb.firebaseio.com",
projectId: "smart-city-295ac",
storageBucket: "smart-city-295ac.appspot.com",
messagingSenderId: "52993427272",
appId: "1:52993427272:web:25dee83c31ef9734ee957d",
measurementId: "G-328BDXCK4D"
};
firebase.initializeApp(config);
firebase.analytics();
var database = firebase.database();
var lastIndex = 0;
// Get Data
firebase.database().ref('Issues/').on('value', function (snapshot) {
var value = snapshot.val();
var htmls = [];
$.each(value, function (index, value) {
if (value) {
htmls.push('<tr>\n
<td>' + (value.Date).replace (/" /g, " ) + '</td>\n
<td>' + (value.Category).replace (/" /g, " ) + '</td>\n
<td>' + (value.Issue).replace (/" /g, " ) + '</td>\n
<td>' + (value.Address).replace (/" /g, " ) + '</td>\n

```

```

<td>' + (value.Status).replace (/"/g, '"') + '</td>\
<td>' + (value.Actions).replace (/"/g, '"') + '</td>\
<td><button data-toggle="modal" data-target="#update-modal" class="btn btn-info updateData" data-id="' + index + '">Update</button>\
<button data-toggle="modal" data-target="#remove-modal" class="btn btn-danger removeData" data-id="' + index + '">Delete</button></td>\
</tr>');
}
lastIndex = index;
});
$('#tbody').html(htmls);
$('#submitUser').removeClass('disabled');
});
// Add Data
$('#submitUser').on('click', function () {
var values = $('#addUser').serializeArray();
var text2 = '\
var date = values[0].value;
var time = values[1].value;
var Event_content = values[2].value;
var Event_title = values[3].value;
var Place = values[4].value;
var userID = lastIndex + 1;
console.log(values);
firebase.database().ref('Events/' + userID).set({
Date: text2+date+text2,
Time: text2+time+text2,
Event_Content:text2+Event_content+text2,
Event_Title:text2+Event_title+text2,
Place:text2+Place+text2,
});

// Reassign lastID value
lastIndex = userID;
$('#addUser input').val('');
});
// Update Data
var updateID = 0;
$('body').on('click', '.updateData', function () {
updateID = $(this).attr('data-id');
firebase.database().ref('Issues/' + updateID).on('value', function (snapshot) {
var values = snapshot.val();
//var tempvalues = (values.Category).substring(1, (values.Category).length-1)
var updateData = '<div class="form-group">\
<div class="col-md-12">\
<select name="Status" class="form-control" id="status" name="status" value="' + (values.Status).replace (/"/g, '"') + '" required autofocus >\
<option>Εκκρεμεί</option>\
<option>Σε εξέλιξη</option>\
<option>Επιλυμένο</option>\
</select>\
</div>\
</div>\
<div class="form-group">\
<label for="Actions" class="col-md-12 col-form-label">Ενέργειες</label>\
<div class="col-md-12">\
<input id="Actions" type="text" class="form-control" name="actions" value="' + (values.Actions).replace (/"/g, '"') + '" required autofocus>\
</div>\

```

```

</div>\
<div class="form-group">\
<div class="col-md-12">\
<div><input id="Hidden_Address" type="hidden" class="form-control" name="hidden_address" value="" + (values.Address).replace (/"/g, '"') + "" required autofocus>\
</div>\
</div>\
<div class="form-group">\
<div class="col-md-12">\
<div><input id="Hidden_Category" type="hidden" class="form-control" name="hidden_category" value="" + (values.Category).replace (/"/g, '"') + "" required autofocus>\
</div>\
<div class="form-group">\
<div class="col-md-12">\
<div><input id="Hidden_Cordinates" type="hidden" class="form-control" name="hidden_cordinates" value="" + (values.Cordinates).replace (/"/g, '"') + "" required autofocus>\
</div>\
<div class="form-group">\
<div class="col-md-12">\
<div><input id="Hidden_Date" type="hidden" class="form-control" name="hidden_date" value="" + (values.Date).replace (/"/g, '"') + "" required autofocus>\
</div>\
</div>\
<div class="form-group">\
<div class="col-md-12">\
<div><input id="Hidden_Image" type="hidden" class="form-control" name="hidden_image" value="" + (values.Image).replace (/"/g, '"') + "" required autofocus>\
</div>\
<div class="form-group">\
<div class="col-md-12">\
<div><input id="Hidden_Issue" type="hidden" class="form-control" name="hidden_issue" value="" + (values.Issue).replace (/"/g, '"') + "" required autofocus>\
</div>\
<div class="form-group">\
<div class="col-md-12">\
<div><input id="Hidden_Marker_Id" type="hidden" class="form-control" name="hidden_marker_id" value="" + (values.Marker_Id).replace (/"/g, '"') + "" required autofocus>\
</div>\
<div class="form-group">\
<div class="col-md-12">\
<div><input id="Hidden_Priority" type="hidden" class="form-control" name="hidden_priority" value="" + (values.Priority).replace (/"/g, '"') + "" required autofocus>\
</div>\
<div class="form-group">\
<div class="col-md-12">\
<div><input id="Hidden_Unique_Ticket_ID" type="hidden" class="form-control" name="hidden_unique_ticket_id" value="" + (values.Unique_Ticket_ID).replace (/"/g, '"') + "" required autofocus>\
</div>\
<div class="form-group">\
<div class="col-md-12">\
<div><input id="Hidden_UserId" type="hidden" class="form-control" name="hidden_userid" value="" + (values.UserId).replace (/"/g, '"') + "" required autofocus>\
</div>';
$('#updateBody').html(updateData);
});
});
$('.updateUser').on('click', function () {
    var text2 = '\n';

```

```

var values = $(".users-update-record-model").serializeArray();
if ((values[0].value) == "Εκκρεμεί"){
    temp_cord = (values[4].value);
    temp_cord = temp_cord.replace('color:60','color:0');
    temp_cord = temp_cord.replace('color:120','color:0');
}
else if ((values[0].value) == "Σε εξέλιξη"){
    temp_cord = (values[4].value);
    temp_cord = temp_cord.replace('color:0','color:60');
    temp_cord = temp_cord.replace('color:120','color:60');
}
else if ((values[0].value) == "Επιλυμένο"){
    temp_cord = (values[4].value);
    temp_cord = temp_cord.replace('color:0','color:120');
    temp_cord = temp_cord.replace('color:60','color:120');
}
var postData = {
    Status: text2 + values[0].value + text2,
    Actions: text2 + values[1].value + text2,
    Address: text2 + values[2].value + text2,
    Category: text2 + values[3].value + text2,
    Coordinates: text2 + temp_cord + text2,
    Date: text2 + values[5].value + text2,
    Image: text2 + values[6].value + text2,
    Issue: text2 + values[7].value + text2,
    Marker_Id: text2 + values[8].value + text2,
    Priority: text2 + values[9].value + text2,
    Unique_Ticket_ID: text2 + values[10].value + text2,
    UserId: text2 + values[11].value + text2,
};
var updates = {};
updates['Issues/' + updateID] = postData;
firebase.database().ref().update(updates);
$("#update-modal").modal('hide');
});
// Remove Data
$("body").on('click', '.removeData', function () {
var id = $(this).attr('data-id');
$("body").find('.users-remove-record-model').append('<input name="id" type="hidden" value="' + id + '">');
});
$('.deleteRecord').on('click', function () {
var values = $(".users-remove-record-model").serializeArray();
var id = values[0].value;
firebase.database().ref('Issues/' + id).remove();
$("body").find('.users-remove-record-model').find("input").remove();
$("#remove-modal").modal('hide');
});
$('.remove-data-from-delete-form').click(function () {
$("body").find('.users-remove-record-model').find("input").remove();
});
</script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
</body>
</html>

```

Issues.html

Παράρτημα Γ: «Ο Κώδικας της IoT συσκευής θερμοκρασίας -υγρασίας»

```
#include <ESP8266WiFi.h>                // esp8266 library
#include <FirebaseArduino.h>             // firebase library
#include <DHT.h>                         // dht11 temperature and humidity sensor library
#define FIREBASE_HOST "smart-city-295ac-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "RyFSa8oz2LaalB0cDo1QBdfXspvNy35plhIwdfhjrdFz"
#define WIFI_SSID "proxim_2.4"
#define WIFI_PASSWORD "a25a80zs2Yd903001d877s6" //password of wifi ssid
#define DHTPIN D4                       // what digital pin we're connected to
#define DHTTYPE DHT22                   // select dht type as DHT 11 or DHT22
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  delay(1000);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); //try to connect with wifi
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP()); //print local IP address
  WiFi.setAutoReconnect(true);
  WiFi.persistent(true);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // connect to firebase
  dht.begin(); //Start reading dht sensor
}

void loop() {
  float h = dht.readHumidity(); // Reading temperature or humidity takes
  about 250 milliseconds!
  float t = dht.readTemperature(); // Read temperature as Celsius (the default)

  if (isnan(h) || isnan(t)) { // Check if any reads failed and exit early (to try
  again).
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  Firebase.setFloat ("Environment/Humitidy",h);
  Serial.print("Humidity: "); Serial.print(h);
  Firebase.setFloat ("Environment/Temperture",t);
  Serial.print("% Temperature: "); Serial.print(t); Serial.println("°C ");

  delay(300000);
}
```

Firrebase_DHT22.ino

Παράρτημα Δ: «Ο Κώδικας της IoT συσκευής μέτρησης θέσεων στάθμευσης»

```
#include <ESP8266WiFi.h> // esp8266 library
#include <FirebaseArduino.h> // firebase library

#define FIREBASE_HOST "smart-city-295ac-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "RyFaS8oz2sfLalB0cDxo1QBffXspvNy35plfhlwdhfjrdFz"
#define WIFI_SSID "TP-LINK_9A72"
#define WIFI_PASSWORD "a25f80ssd2Y001g8776" //password of wifi ssid

int pinin=14;
int pinin1=12;
int count=0;

void setup() {

  pinMode(pinin,INPUT);
  pinMode(pinin1,INPUT);
  Serial.begin(115200);

  delay(1000);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); //try to connect with wifi
  Serial.print("Connecting to ");
  Serial.print(WIFI_SSID);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Connected to ");
  Serial.println(WIFI_SSID);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP()); //print local IP address
  WiFi.setAutoReconnect(true);
  WiFi.persistent(true);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // connect to firebase
}

void loop()
{
  if(digitalRead(pinin)==1)
  {
    delay(2500);

    count++;

    Serial.print("PEOPLE INSIDE: ");
    Serial.println(count);
    Firebase.setFloat ("Parking/1175/Free_Places", count);
  }

  else
  if(digitalRead(pinin1)==1)
```

```
{  
  
  delay(2500);  
  
  count--;  
  
  Serial.print("PEOPLE INSIDE: ");  
  Serial.println(count);  
  Firebase.setFloat ("Parking/1175/Free_Places", count);  
}  
  
}
```

Parking.ino