

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

*ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ*



Διπλωματική εργασία

Σύστημα απομακρυσμένης διαχείρισης και ελέγχου για θερμοκήπιο εσωτερικού χώρου

Κωνσταντόπουλος Κωνσταντίνος

Εξεταστική επιτροπή:

Καθ. Δόλλας Απόστολος (Επιβλέπων)

Καθ. Μπλέτσας Άγγελος

Αν. Καθ. Μανιά Αικατερίνη

Χανιά 2020

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Απόστολο Δόλλα, επιβλέποντα της παρούσας διπλωματικής εργασίας και καθηγητή μου, για την υποστήριξη του καθ' όλη την διάρκεια της διπλωματικής μου εργασίας, καθώς και τον καθηγητή κ. Άγγελο Μπλέτσα και την αν. καθηγήτρια κ. Αικατερίνη Μανιά, μέλη της τριμελούς επιτροπής για την συμβολή τους. Θα ήθελα επιπλέον να ευχαριστήσω τους γονείς μου για την αστείρευτή τους συμπαράσταση και στήριξη, καθώς επίσης και τον αγώνα τους, ώστε να καταφέρω να ολοκληρώσω τις σπουδές μου.

Περίληψη

Η παρούσα διπλωματική εργασία αφορά την σχεδίαση, ανάπτυξη και εξέταση απόδοσης, συστήματος διαχείρισης «έξυπνου» θερμοκηπίου εσωτερικού χώρου. Το σύστημα διαχείρισης αυτό, εξυπηρετεί τρεις σκοπούς. Πρώτον, παρέχει στον χρήστη την δυνατότητα παραμετροποίησης των περιβαλλοντικών συνθηκών της καλλιέργειάς του, δεύτερον παρέχει την δυνατότητα ελέγχου των συνθηκών ανάπτυξης μέσω γραφημάτων για όλο το φάσμα των τριών βασικών περιβαλλοντικών τιμών, και τρίτον, μέσω μηχανισμού παρακολούθησης (monitoring) της διαδικασίας καλλιέργειας, διασφαλίζει στον χρήστη την ορθή λειτουργία του θερμοκηπίου και συνεπώς την ομαλή ανάπτυξη της καλλιέργειάς του. Το σύστημα αποτελείται από έναν συνδυασμό backend και frontend. Το backend, συνίσταται από μία βάση δεδομένων και ένα api για την εξυπηρέτηση χρηστών και την διαχείριση θερμοκηπίων, εισάγοντας ταυτόχρονα εποπτικές διεργασίες ορθής λειτουργίας. Το frontend, είναι μία εφαρμογή σε περιβάλλον android, που εξυπηρετεί την εύχρηστη αλληλεπίδραση και παροχή πληροφοριών στον χρήστη, καθώς και την λήψη ειδοποιήσεων σε όλες τις συνδεδεμένες συσκευές του, σε περιπτώσεις αστοχίας στην λειτουργία κάποιου θερμοκηπίου.

Abstract

This thesis concerns the design, development and performance testing of a "smart" indoor greenhouse management system. This management system serves three purposes. Firstly, it enables the user to customize the environmental conditions of his crop, secondly, it provides the possibility of controlling the growth conditions through graphs for the whole range of the three basic environmental values, and thirdly, through a mechanism for monitoring the cultivation process, it provides to the user useful information regarding the correct operation of the greenhouse and therefore facilitates the smooth development of the crop. The system consists of a combination of backend and frontend. The backend consists of a database and an api for user service and greenhouse management, while introducing proper operation monitoring processes. The frontend, is an application in android environment, that serves the easy to use interaction and supply of information to the user, as well as receiving notifications on all its connected devices, in cases of a greenhouse's operation failure.

Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ.....	12
1.1 Θερμοκήπιο εσωτερικού χώρου LEKO	12
1.1.1 Περιγραφή Leko	12
1.1.2 Τα δύο υποσυστήματα.....	13
1.1.3 Επικοινωνία Υποσυστημάτων	14
1.1.4 Κάρτα διασύνδεσης ESP8266-01 και TCP/IP	14
1.2 Σκοπός Διπλωματικής Εργασίας	16
1.3 Οργάνωση διπλωματικής εργασίας	16
2. ΣΧΕΤΙΚΗ ΕΡΕΥΝΑ.....	17
2.1 IoT 17	
2.1.1 Αρχιτεκτονικές δικτύων IoT	18
2.1.2 Επίπεδα IoT	18
2.1.3 Πρωτόκολλα IoT	19
2.1.4 Θέματα ασφαλείας στις IoT εφαρμογές.....	20
2.2 Γεωργία ακριβείας.....	21
2.2.1 Εφαρμογές IoT στην γεωργία	21
2.3 Έρευνα Τεχνολογιών	22
2.3.1 Σχεσιακή & Μη-σχεσιακή λογική βάσεως δεδομένων	22
2.3.2 Αρχιτεκτονικές API & Server Scripting Languages	23
2.3.3 Τεχνικές API authentication	24
2.3.4 Αποστολή Notifications σε android	25
2.4 Λειτουργικό σύστημα Android	25
2.4.1 Αρχιτεκτονική Android.....	25
2.4.2 Δραστηριότητα android.....	26
2.4.3 Χρήσιμες βιβλιοθήκες android	28
2.5 Εργαλεία ανάπτυξης	29
2.5.1 Android studio	29
2.5.2 WampServer	29
2.5.3 Jmeter 29	
2.5.4 Postman 30	
2.5.5 Wireshark 30	
3. ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ ΚΑΙ ΣΧΕΔΙΑΣΤΙΚΕΣ ΕΠΙΛΟΓΕΣ.....	31
3.1 Επισκόπηση συστήματος	31
3.2 Ανάλυση συστήματος θερμοκηπίου	31

3.2.1	Μοντελοποίηση συστήματος θερμοκηπίου	31
3.3	Ανάλυση συστήματος διαχείρισης.....	32
3.3.1	Μοντελοποίηση συστήματος διαχείρισης	32
3.3.2	Απαιτήσεις χρήστη	33
3.3.3	Υποσυστήματα συστήματος διαχείρισης.....	34
3.4	Σχεδιαστικές επιλογές.....	34
3.4.1	Διεπαφή χρήστη	34
3.4.2	Διεπαφή θερμοκηπίου	35
3.4.3	Διαδικασία σύνδεσης του θερμοκηπίου με το σύστημα.....	37
3.5	Σχεδίαση διαδικασιών συστήματος.....	38
3.6	Λειτουργικές απαιτήσεις και θέματα αξιοπιστίας συστήματος	40
4.	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ.....	41
4.1	Αρχιτεκτονική backend συστήματος	41
4.1.1	Αρχιτεκτονική API	42
4.2	Τελική αρχιτεκτονική συστήματος	43
5.	ΣΧΕΔΙΑΣΗ ΣΥΣΤΗΜΑΤΟΣ	44
5.1	Σχεδίαση Βάσης Δεδομένων.....	44
5.1.1	Σχεδίαση οντοτήτων	45
5.1.2	Τελικό σχήμα βάσης.....	49
5.2	Σχεδίαση API	51
5.2.1	API Documentation.....	52
5.2.2	Αιτήματα Χρήστη	55
5.2.3	Αιτήματα θερμοκηπίου.....	57
5.2.4	Monitoring συστήματος.....	57
5.2.5	Ροή λειτουργίας συστήματος Notification.....	59
5.2.6	Υλοποίηση API authentication	60
5.3	Εφαρμογή Android.....	61
5.3.1	Άδειες χρήσης εφαρμογής	62
5.3.2	Εφαρμογή LekoApp.....	62
5.3.3	Επικοινωνία εφαρμογής με Server.....	64
5.3.4	Γραφικές παραστάσεις εβδομαδιαίων αναφορών	65
5.3.5	Λήψη ειδοποιήσεων.....	66
6.	Επιβεβαίωση λειτουργίας	67
6.1	Εγγραφή χρήστη.....	67
6.2	Είσοδος χρήστη	70
6.2.1	Κεντρική οθόνη LekoApp	71

6.2.2	Αποθήκευση διακριτικού συσκευής.....	71
6.3	Διαχείριση θερμοκηπίων Leko	71
6.3.1	Προσθήκη θερμοκηπίου	73
6.3.2	Λίστα θερμοκηπίων	75
6.4	Επικοινωνία με θερμοκήπιο Leko	76
6.5	Εβδομαδιαίες αναφορές θερμοκηπίων Leko.....	78
6.5.1	Γραφική παράσταση περιβαλλοντικών τιμών.....	82
6.5.2	Γραφική παράσταση συμβάντων	84
6.6	Ειδοποιήσεις χρήστη	85
6.6.1	Ειδοποιήσεις εσφαλμένων περιβαλλοντικών τιμών.....	87
6.6.2	Ειδοποιήσεις σε πολλαπλές συσκευές.....	88
6.6.3	Ειδοποιήσεις αδράνειας.....	89
6.6.4	Ειδοποιήσεις σε συγκεκριμένο θερμοκήπιο από τον browser	90
6.7	API Authentication	91
7.	API Performance Test.....	93
7.1	Πλάνο Testing – Φάσεις Πειράματος	93
7.2	Περιβάλλον Testing	94
7.3	Εργαλεία Testing	95
7.4	Εκτέλεση Performance Test	95
7.4.1	Εκτέλεση 1 ^{ου} κύκλου.....	96
7.4.2	Εκτέλεση 2 ^{ου} κύκλου.....	96
7.4.3	Εκτέλεση 3 ^{ου} κύκλου.....	97
7.5	Αποτελέσματα.....	97
7.5.1	Response Times ανά transaction.....	100
7.5.2	Errors ανά transaction	102
7.5.3	Bytes & Request Throughput ανά transaction.....	104
7.5.4	Packet transfer ανά κύκλο	107
7.5.5	Χρήση CPU & Μνήμης	108
7.6	Απολογισμός Αποτελεσμάτων	109
8.	Συμπεράσματα & Μελλοντικές βελτιώσεις	110
8.1	Συμπεράσματα και γνώση που αποκτήθηκε.....	110
8.2	Μελλοντικές Βελτιώσεις	110
8.2.1	Μετατροπή σε MQTT ή CoAP	110
8.2.2	Data Analysis	110
8.2.3	Ενεργειακή αναφορά θερμοκηπίου	111
8.2.4	Ειδοποίηση για κλάδεμα ή συγκομιδή.....	111

8.2.5	Ειδοποίηση δεξαμενής νερού	111
8.2.6	Διασύνδεση μεταξύ των χρηστών	111
8.2.7	Τροποποίηση παραμέτρων θερμοκηπίου από τον χρήστη	111
9.	ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ.....	113

Ευρετήριο εικόνων

ΕΙΚΟΝΑ 2-1. IOT USERS TREND[51].	18
ΕΙΚΟΝΑ 4-1. ΔΙΑΔΙΚΑΣΙΑ ΣΥΝΔΕΣΗΣ ΘΕΡΜΟΚΗΠΙΟΥ ΛΕΚΟ ΜΕ ΤΟΝ WEBSERVER.	37
ΕΙΚΟΝΑ 5-1. LOGO ΕΦΑΡΜΟΓΗΣ.	61
ΕΙΚΟΝΑ 6-1. ΟΘΟΝΗ LOGIN	67
ΕΙΚΟΝΑ 6-2. ΟΘΟΝΗ ΕΓΓΡΑΦΗΣ	68
ΕΙΚΟΝΑ 6-3. ΣΤΟΙΧΕΙΑ ΕΓΓΡΑΦΗΣ ΧΡΗΣΤΗ	68
ΕΙΚΟΝΑ 6-4. ΜΗΝΥΜΑ ΕΠΙΤΥΧΗΜΕΝΗΣ ΕΓΓΡΑΦΗΣ	69
ΕΙΚΟΝΑ 6-5. TABLE USERS	69
ΕΙΚΟΝΑ 6-6. LOGIN ΕΓΓΕΓΡΑΜΜΕΝΟΥ ΧΡΗΣΤΗ	70
ΕΙΚΟΝΑ 6-7. ΚΕΝΤΡΙΚΗ ΟΘΟΝΗ ΕΦΑΡΜΟΓΗΣ	71
ΕΙΚΟΝΑ 6-8. ΠΙΝΑΚΑΣ USER_DEVICES	71
ΕΙΚΟΝΑ 6-9. ΛΕΚΟ MANAGER	72
ΕΙΚΟΝΑ 6-10. ΤΡΟΠΟΠΟΙΗΣΗ ΣΤΟΙΧΕΙΩΝ ΠΡΟΦΙΛ ΧΡΗΣΤΗ	72
ΕΙΚΟΝΑ 6-11. ΠΡΟΣΘΗΚΗ ΛΕΚΟ	73
ΕΙΚΟΝΑ 6-12. ΛΙΣΤΑ ΤΥΠΩΝ ΘΕΡΜΟΚΗΠΙΟΥ	73
ΕΙΚΟΝΑ 6-13. ΠΡΟΣΘΗΚΗ ΘΕΡΜΟΚΗΠΙΟΥ ΤΥΠΟΥ 1	74
ΕΙΚΟΝΑ 6-14. ΚΕΝΤΡΟ ΕΛΕΓΧΟΥ ΜΕΤΑ ΤΗΝ ΕΙΣΑΓΩΓΗ	74
ΕΙΚΟΝΑ 6-15. ΠΙΝΑΚΑΣ ΛΕΚΟ	75
ΕΙΚΟΝΑ 6-16. ΟΘΟΝΗ ΛΙΣΤΑΣ ΛΕΚΟΣ ΤΟΥ ΧΡΗΣΤΗ	76
ΕΙΚΟΝΑ 6-17. ΠΡΟΣΟΜΟΙΩΣΗ ΘΕΡΜΟΚΗΠΙΟΥ ΛΕΚΟ	77
ΕΙΚΟΝΑ 6-18. ΛΕΚΟ ΑΝΑΝΕΩΣΗ ΤΙΜΩΝ	77
ΕΙΚΟΝΑ 6-19. ΠΙΝΑΚΑΣ HISTORY	77
ΕΙΚΟΝΑ 6-20. ΠΙΝΑΚΑΣ EVENTS	78
ΕΙΚΟΝΑ 6-21. ΟΘΟΝΗ ΛΕΚΟ	78
ΕΙΚΟΝΑ 6-22. ΑΝΑΦΟΡΑ ΤΙΜΩΝ ΕΠΕΙΤΑ ΑΠΟ ΤΗΝ ΠΡΩΤΗ ΕΓΓΡΑΦΗ	79
ΕΙΚΟΝΑ 6-23. ΑΝΑΦΟΡΑ ΕΝΕΡΓΟΠΟΙΗΤΩΝ ΕΠΕΙΤΑ ΑΠΟ ΤΗΝ ΠΡΩΤΗ ΕΓΓΡΑΦΗ	79
ΕΙΚΟΝΑ 6-24. ΠΙΝΑΚΑΣ HISTORY	81
ΕΙΚΟΝΑ 6-25. ΠΙΝΑΚΑΣ EVENTS	81
ΕΙΚΟΝΑ 6-26. ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ ΠΕΡΙΒΑΛΛΟΝΤΙΚΩΝ ΤΙΜΩΝ ΕΒΔΟΜΑΔΑΣ	82
ΕΙΚΟΝΑ 6-27. ΗΜΕΡΗΣΙΕΣ ΑΝΑΦΟΡΕΣ ΕΒΔΟΜΑΔΑΣ	83
ΕΙΚΟΝΑ 6-28. ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ ΣΥΜΒΑΝΤΩΝ ΕΒΔΟΜΑΔΑΣ	84
ΕΙΚΟΝΑ 6-29. ΗΜΕΡΗΣΙΑ ΑΝΑΦΟΡΑ ΣΥΜΒΑΝΤΩΝ ΕΒΔΟΜΑΔΑΣ	85
ΕΙΚΟΝΑ 6-30. ΛΙΣΤΑ ΘΕΡΜΟΚΗΠΙΩΝ ΠΟΥ ΠΡΟΣΤΕΘΗΚΑΝ	86
ΕΙΚΟΝΑ 6-31. ΕΚΤΕΛΕΣΗ ΡΟΥΤΙΝΑΣ CHECKΛΕΚΟΣ	87
ΕΙΚΟΝΑ 6-32. NOTIFICATIONS ΕΣΦΑΛΜΕΝΩΝ ΤΙΜΩΝ	88
ΕΙΚΟΝΑ 6-33. ΔΙΑΚΡΙΤΙΚΑ ΕΓΓΡΑΦΗΣ ΣΥΣΚΕΥΩΝ ΣΤΟΝ ΠΙΝΑΚΑ USER_DEVICES	88
ΕΙΚΟΝΑ 6-34. ΤΑΥΤΟΧΡΟΝΗ ΑΠΟΣΤΟΛΗ ΕΙΔΟΠΟΙΗΣΗΣ ΣΤΙΣ ΕΝΕΡΓΕΣ ΣΥΣΚΕΥΕΣ ΤΟΥ ΧΡΗΣΤΗ	89
ΕΙΚΟΝΑ 6-35. ΕΙΔΟΠΟΙΗΣΗ ΧΡΗΣΤΗ ΓΙΑ ΑΔΡΑΝΕΙΑ ΘΕΡΜΟΚΗΠΙΟΥ	90
ΕΙΚΟΝΑ 6-36. ΑΠΟΣΤΟΛΗ ΕΙΔΟΠΟΙΗΣΗΣ ΓΙΑ ΣΥΓΚΕΚΡΙΜΕΝΟ ΛΕΚΟ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	90
ΕΙΚΟΝΑ 6-37. API TOKEN – REGISTER USER	91
ΕΙΚΟΝΑ 6-38. API TOKEN – LOGIN USER	91
ΕΙΚΟΝΑ 6-39. API AUTHENTICATION FAILED	92
ΕΙΚΟΝΑ 6-40. API AUTHENTICATION SUCCESSFUL	92

Ευρετήριο σχημάτων

ΣΧΗΜΑ 1-1. TOP LEVEL ΜΟΝΤΕΛΟ FRONT-END ΣΥΣΤΗΜΑΤΟΣ & ΣΥΣΤΗΜΑΤΟΣ ΘΕΡΜΟΚΗΠΙΩΝ	14
ΣΧΗΜΑ 1-2. ΕΠΙΚΟΙΝΩΝΙΑ SERVER ΜΕ ΤΟ ΣΥΣΤΗΜΑ ΤΩΝ ΘΕΡΜΟΚΗΠΙΩΝ ΜΕΣΩ MODEM/ROUTER.....	14
ΣΧΗΜΑ 1-3. TCP/IP DATA ENCAPSULATION[57].....	15
ΣΧΗΜΑ 2-1. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ANDROID.....	26
ΣΧΗΜΑ 2-2. ΚΥΚΛΟΣ ΖΩΗΣ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ[56].	28
ΣΧΗΜΑ 3-1. ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΘΕΡΜΟΚΗΠΙΟΥ TOP LEVEL.	31
ΣΧΗΜΑ 3-2. ΕΙΣΟΔΟΙ – ΕΞΟΔΟΙ ΣΥΣΤΗΜΑΤΟΣ ΘΕΡΜΟΚΗΠΙΟΥ.	32
ΣΧΗΜΑ 3-3. ΕΙΣΟΔΟΙ – ΕΞΟΔΟΙ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ	33
ΣΧΗΜΑ 3-4. ΔΙΑΧΕΙΡΙΣΗ ΠΟΛΛΑΠΛΩΝ ΘΕΡΜΟΚΗΠΙΩΝ.	33
ΣΧΗΜΑ 3-5. FRONT ΚΑΙ BACK END ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ.....	34
ΣΧΗΜΑ 3-6. ΔΙΑΓΡΑΜΜΑ ΕΝΕΡΓΕΙΩΝ ΕΠΙΚΟΙΝΩΝΙΑΣ ΘΕΡΜΟΚΗΠΙΟΥ ΚΑΙ WEB SERVER.....	38
ΣΧΗΜΑ 3-7. ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΣΥΣΤΗΜΑΤΟΣ MONITORING.	40
ΣΧΗΜΑ 4-1. BACKEND DIAGRAM.....	41
ΣΧΗΜΑ 4-2. TOP LEVEL ΜΟΝΤΕΛΟ ΣΥΣΤΗΜΑΤΟΣ.....	42
ΣΧΗΜΑ 4-3. SIDE SERVER SCRIPT GROUPS.	43
ΣΧΗΜΑ 4-4. ΤΕΛΙΚΗ ΔΟΜΗ ΣΥΣΤΗΜΑΤΟΣ.	43
ΣΧΗΜΑ 5-1. ΥΠΟΣΥΣΤΗΜΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	44
ΣΧΗΜΑ 5-2. ATTRIBUTES ΧΡΗΣΤΗ.	46
ΣΧΗΜΑ 5-3. ATTRIBUTES ΘΕΡΜΟΚΗΠΙΟΥ.....	47
ΣΧΗΜΑ 5-4. ATTRIBUTES ΓΕΓΟΝΟΤΩΝ.	48
ΣΧΗΜΑ 5-5. ΟΝΤΟΤΗΤΑ ΙΣΤΟΡΙΚΟ	48
ΣΧΗΜΑ 5-6. ATTRIBUTES ΣΥΝΔΕΔΕΜΕΝΩΝ ΣΥΣΚΕΥΩΝ.	49
ΣΧΗΜΑ 5-7. ΕΞΑΡΤΗΣΕΙΣ ΜΕΤΑΞΥ ΧΡΗΣΤΗ & ΘΕΡΜΟΚΗΠΙΟΥ	50
ΣΧΗΜΑ 5-8. ΕΞΑΡΤΗΣΕΙΣ ΜΕΤΑΞΥ ΘΕΡΜΟΚΗΠΙΟΥ & ΙΣΤΟΡΙΚΟΥ/ΓΕΓΟΝΟΤΟΣ.....	50
ΣΧΗΜΑ 5-9 . RELATIONAL SCHEMA ΒΑΣΗΣ ΣΥΣΤΗΜΑΤΟΣ.	51
ΣΧΗΜΑ 5-10. ΥΠΟΣΥΣΤΗΜΑ API	52
ΣΧΗΜΑ 5-11. ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΣΥΣΚΕΥΩΝ.....	59
ΣΧΗΜΑ 5-12. ΡΟΗ ΛΕΙΤΟΥΡΓΙΑΣ FIREBASE CLOUD MESSAGING.....	59
ΣΧΗΜΑ 5-13. API AUTHENTICATION FLOW.	61
ΣΧΗΜΑ 5-14. ΧΑΡΤΗΣ ΕΝΕΡΓΕΙΩΝ ΕΦΑΡΜΟΓΗΣ	64
ΣΧΗΜΑ 5-15. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΠΙΚΟΙΝΩΝΙΑΣ ANDROID – SERVER.....	65
ΣΧΗΜΑ 7-1. REQUEST VS 90% RESPONSE TIME RANGE.....	98
ΣΧΗΜΑ 7-2. AVERAGE RESPONSE TIME PER TRANSACTION.	99
ΣΧΗΜΑ 7-3. MONITORING RESPONSE TIMES AFTER EACH CYCLE.....	99
ΣΧΗΜΑ 7-4. ERROR RATES 3 ^{ΟΥ} ΚΥΚΛΟΥ.....	103
ΣΧΗΜΑ 7-5. ΑΡΙΘΜΟΣ ΠΑΚΕΤΩΝ ΑΝΑ ΚΥΚΛΟ.....	107
ΣΧΗΜΑ 7-6. ΑΡΙΘΜΟΣ ΠΑΚΕΤΩΝ ΛΕΙΤΟΥΡΓΙΑΣ MONITORING ΕΠΕΙΤΑ ΑΠΟ ΚΑΘΕ ΚΥΚΛΟ.....	107
ΣΧΗΜΑ 7-7. CPU & MEMORY UTILIZATION.	108

Ευρετήριο πινάκων

ΠΙΝΑΚΑΣ 3-1. ΣΥΓΚΡΙΣΗ ΙΣΤΟΣΕΛΙΔΑΣ-ΕΦΑΡΜΟΓΗΣ ANDROID.....	35
ΠΙΝΑΚΑΣ 3-2. ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΠΡΩΤΟΚΟΛΛΩΝ ΙΟΤ, HTTP, MQTT, CoAP.....	36
ΠΙΝΑΚΑΣ 3-3. ΔΟΜΗ ΔΕΔΟΜΕΝΩΝ ΕΝΗΜΕΡΩΣΗΣ ΘΕΡΜΟΚΗΠΙΟΥ.....	38
ΠΙΝΑΚΑΣ 5-1. RELATIONAL VS NON- RELATIONAL DB.....	45
ΠΙΝΑΚΑΣ 5-2. ΕΠΙΠΛΕΟΝ ΠΕΔΙΑ ΤΗΣ ΟΝΤΟΤΗΤΑΣ ΧΡΗΣΤΗ.....	47
ΠΙΝΑΚΑΣ 5-3. REST vs SOAP.....	52
ΠΙΝΑΚΑΣ 5-4. PHP vs PYTHON vs RUBY.....	53
ΠΙΝΑΚΑΣ 5-5. API DOCUMENTATION- USER ENDPOINTS.....	54
ΠΙΝΑΚΑΣ 5-6. API DOCUMENTATION - LECO ENDPOINTS.....	55
ΠΙΝΑΚΑΣ 5-7. API DOCUMENTATION-SYSTEM ENDPOINTS.....	55
ΠΙΝΑΚΑΣ 5-8. ΣΥΝΙΣΤΩΜΕΝΑ ΧΩΡΙΑ ΠΕΡΙΒΑΛΛΟΝΤΙΚΩΝ ΤΙΜΩΝ ΑΝΑ ΤΥΠΟ ΚΑΛΛΙΕΡΓΕΙΑΣ [50].....	58
ΠΙΝΑΚΑΣ 5-9. ΟΘΟΝΕΣ ΕΦΑΡΜΟΓΗΣ LECOAPP.....	63
ΠΙΝΑΚΑΣ 6-1. ΔΕΙΓΜΑΤΑ ΘΕΡΜΟΚΗΠΙΟΥ.....	80
ΠΙΝΑΚΑΣ 6-2. ΘΕΡΜΟΚΗΠΙΑ ΧΡΗΣΤΗ.....	86
ΠΙΝΑΚΑΣ 7-1. TEST TRANSACTIONS.....	94
ΠΙΝΑΚΑΣ 7-2. ΜΟΝΑΔΕΣ TEST.....	95
ΠΙΝΑΚΑΣ 7-3. ΠΑΡΑΜΕΤΡΟΙ 1 ^{ΟΥ} ΚΥΚΛΟΥ.....	96
ΠΙΝΑΚΑΣ 7-4. ΠΑΡΑΜΕΤΡΟΙ 1 ^{ΟΥ} ΚΥΚΛΟΥ.....	96
ΠΙΝΑΚΑΣ 7-5. ΠΑΡΑΜΕΤΡΟΙ 1 ^{ΟΥ} ΚΥΚΛΟΥ.....	97
ΠΙΝΑΚΑΣ 7-6. ΕΠΙΣΚΟΠΗΣΗ TEST CYCLES.....	97
ΠΙΝΑΚΑΣ 7-7. ΚΑΤΑΝΟΜΗ RESPONSE TIME ΑΝΑ ΚΥΚΛΟ.....	98
ΠΙΝΑΚΑΣ 7-8. RESPONSE TIMES 1 ^{ΟΥ} ΚΥΚΛΟΥ.....	100
ΠΙΝΑΚΑΣ 7-9. RESPONSE TIMES 2 ^{ΟΥ} ΚΥΚΛΟΥ.....	101
ΠΙΝΑΚΑΣ 7-10. RESPONSE TIMES 3 ^{ΟΥ} ΚΥΚΛΟΥ.....	102
ΠΙΝΑΚΑΣ 7-11. THROUGHPUT ANALYSIS 1 ^{ΟΥ} ΚΥΚΛΟΥ.....	104
ΠΙΝΑΚΑΣ 7-12. THROUGHPUT ANALYSIS 2 ^{ΟΥ} ΚΥΚΛΟΥ.....	105
ΠΙΝΑΚΑΣ 7-13. THROUGHPUT ANALYSIS 3 ^{ΟΥ} ΚΥΚΛΟΥ.....	106

1. ΕΙΣΑΓΩΓΗ

Την άνοιξη του 2015, στα πλαίσια του μαθήματος “Ενσωματωμένα Συστήματα”, του τμήματος ΗΜΜΥ του Πολυτεχνείου Κρήτης, με διδάσκοντα τον καθηγητή κο Δόλλα Απόστολο, εγώ, ο συγγραφέας της παρούσας διπλωματικής, και ο τότε συμφοιτητής μου κος Σαραντάρης Ελευθέριος, αναπτύξαμε ένα αυτόνομο θερμοκήπιο εσωτερικού χώρου, βασισμένο σε μικροελεγκτή. Το θερμοκήπιο αυτό το ονομάσαμε Leko, εμπνευσμένο από τα αρχικά των ονομάτων μας. Το project αυτό ήταν καθοριστικό για εμάς, καθώς ήρθαμε σε επαφή με τον μαγικό και ιστορικό τομέα της γεωργίας, και συνειδητοποιήσαμε τις επιπλέον δυνατότητες που μπορεί να προσφέρει η τεχνολογία σε αυτόν. Θέλοντας να εμβαθύνουμε περαιτέρω στο IoT και συγκεκριμένα στο smart farming, αποφασίσαμε να εξελίξουμε το Leko, οδηγούμενοι σε δύο ξεχωριστές διπλωματικές εργασίες. Την διπλωματική εργασία “Αυτόνομο θερμοκήπιο εσωτερικού χώρου βασισμένο σε μικροελεγκτή” του κος Σαραντάρη, και την παρούσα διπλωματική εργασία “Σύστημα απομακρυσμένης διαχείρισης και ελέγχου θερμοκηπίου εσωτερικού χώρου”. Η πρώτη αφορά την αναβάθμιση του hardware του αρχικού θερμοκηπίου, και η παρούσα διπλωματική αφορά το υπόλοιπο σύστημα διαχείρισης, που καθιστά την αλληλοεπίδραση του χρήστη με αυτό αλλά και την εξασφάλιση της ομαλής του λειτουργίας.

Το σύστημα διαχείρισης όμως, δεν περιορίζεται μόνο στις τεχνολογίες των υποσυστημάτων που αναπτύχθηκαν και τις μεταξύ τους διεπαφές. Τα παραπάνω αποτελούν τα δομικά του στοιχεία. Το σύστημα διαχείρισης συνθέτει ένα ευρύτερο οικοσύστημα, θερμοκηπίων και χρηστών. Ένα οικοσύστημα, που θα μπορούσε να προωθήσει την αποκεντροποίηση της γεωργίας, κάνοντας την προσιτή στον καθημερινό χρήστη.

Η δημιουργία του συστήματος διαχείρισης απαιτεί το γεφύρωμα θεμελιωδών επιστημών, όπως της επιστήμης των υπολογιστών και της πληροφορικής, με άλλες επιστήμες, όπως η user-centered σχεδίαση με έμφαση στο usability του συστήματος. Στην παρούσα διπλωματική, δίνεται ιδιαίτερη έμφαση στον συνδυασμό των παραπάνω προσεγγίσεων, με απώτερο σκοπό την δημιουργία ενός εύχρηστου και συνάμα αξιόπιστου συστήματος διαχείρισης, που θα ολοκληρώνει το θερμοκήπιο Leko.

1.1 Θερμοκήπιο εσωτερικού χώρου LEKO

Το πρώτο θερμοκήπιο εσωτερικού χώρου Leko, εκπονήθηκε αρχικά από εμένα, συγγραφέα της παρούσας διπλωματικής εργασίας και τον κο. Σαραντάρη Ελευθέριο. Κατόπιν, το Θερμοκήπιο εσωτερικού χώρου Leko αναβαθμίστηκε στα πλαίσια της διπλωματικής εργασίας “Αυτόνομο θερμοκήπιο εσωτερικού χώρου βασισμένο σε μικροελεγκτή” από τον δεύτερο. Αποτελεί ένα σύστημα περιβαλλοντολογικής διαχείρισης με σκοπό την υγιή ανάπτυξη φυτών αυτοματοποιημένα. Παρακάτω περιγράφεται το θερμοκήπιο Leko, η λειτουργικότητά του και οι αναβαθμίσεις που έγιναν.

1.1.1 Περιγραφή Leko

Το Leko είναι ένα θερμοκήπιο εσωτερικού χώρου. Η λειτουργία του βασίζεται στην λήψη βασικών μετρήσεων κρίσιμων για την υγεία ενός φυτού. Οι τιμές αυτές είναι:

- η θερμοκρασία αέρα

- η υγρασία εδάφους
- το επίπεδο του ημερήσιου φωτισμού

Η δειγματοληψία των παραπάνω τιμών, γίνεται με τους κατάλληλους αισθητήρες. Εάν οι τιμές αυτές παρεκκλίνουν των ιδανικών που ορίζει η οικογένεια φυτών που έχει επιλεγεί, το σύστημα παρεμβαίνει για την τροποποίηση και την επαναφορά τους στα επιθυμητά επίπεδα. Αυτό πραγματοποιείται μέσω των ενεργοποιητών. Συγκεκριμένα μέσω ενός ανεμιστήρα, μίας ηλεκτρικής αντλίας νερού και ειδικών λαμπτήρων led.

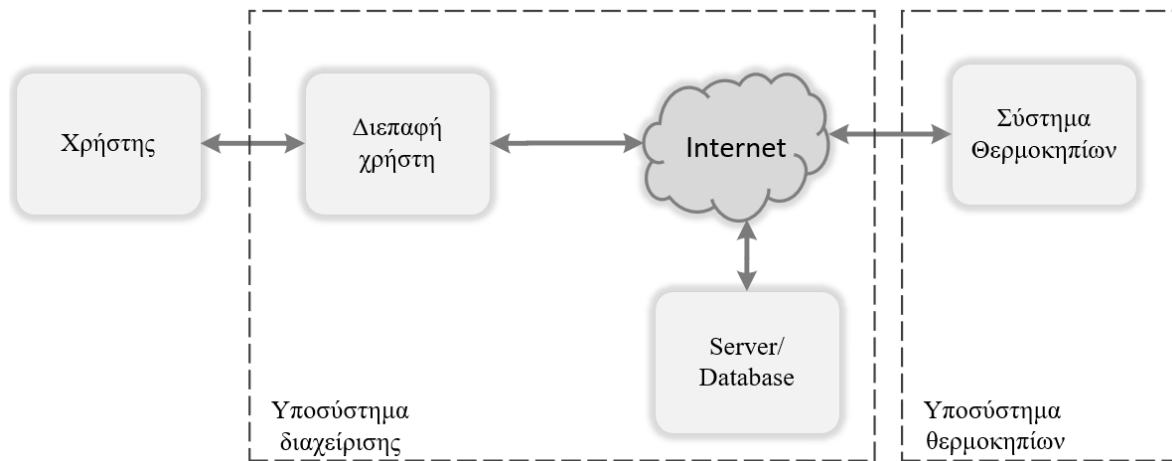
Στην πρώτη έκδοσή του, το θερμοκήπιο δεν παρέχει δυνατότητες συνδεσιμότητας και η παραμετροποίηση του γίνεται μόνο μέσω της οθόνης lcd και των button εισόδου που φέρει το θερμοκήπιο. Στην ανανεωμένη έκδοσή, αποτέλεσμα της δουλειάς του κου. Ε. Σαραντάρη, το Leko φέρει δυνατότητα σύνδεσης στο Internet μέσω της κάρτας διασύνδεσης ESP8266-01, η οποία λειτουργεί μόνο τοπικά μέσω LAN και όχι απομακρυσμένα. Οι τροποποιήσεις που έγιναν στο θερμοκήπιο για την επίτευξη απομακρυσμένης διασύνδεσης, αφορούν την παρούσα διπλωματική και αναφέρονται στην ενότητα 3.

Επιπλέον, στην ανανεωμένη έκδοση του Leko, δίνεται η δυνατότητα επεκτασιμότητας και σύνταξης ενός τοπικού δικτύου θερμοκηπίων, με χαμηλό κόστος, μέσω του πρωτοκόλλου ασύρματης επικοινωνίας RF24[50]. Συγκεκριμένα, ένα πλήρες τοπικό δίκτυο θερμοκηπίων Leko, αποτελείται από μία κεντρική μονάδα, Master, και από έως δύο επιμέρους υπομονάδες Slaves συνδεδεμένες στην κεντρική. Τα θερμοκήπια αυτά έχουν τους ίδιους αισθητήρες και παρέχουν την ίδια λειτουργία. Η διαφορά τους έγκειται στο γεγονός ότι μόνο το Master θερμοκήπιο διαθέτει κάρτα διασύνδεσης στο Internet (ESP8266-01), ρίχνοντας αισθητά το κόστος των θερμοκηπίων Slave [50].

Στην παρούσα διπλωματική δεν διαφοροποιείται ο τύπος του θερμοκηπίου, σε Master ή Slave, καθώς η διευθυνσιοδότηση, μέσω IP, από την κάρτα διασύνδεσης είναι διαφορετική για το καθένα, και η δρομολόγηση των δεδομένων για κάθε θερμοκήπιο του τοπικού δικτύου διεκπεραιώνεται εσωτερικά στο δίκτυο των θερμοκηπίων. Επιπλέον, παραλείποντας τον τύπο του θερμοκηπίου και τον ρόλο του στο τοπικό δίκτυο, δεν προστίθεται επιπλέον πληροφορία στον χρήστη, πέρα των απαραίτητων, που είναι η πορεία της καλλιέργειάς του.

1.1.2 Τα δύο υποσυστήματα

Στην παρούσα διπλωματική εργασία υλοποιείται το υποσύστημα διαχείρισης, όπως φαίνεται στο Σχήμα 1-1. Ο σχεδιασμός και η υλοποίηση του υποσυστήματος θερμοκηπίων, αποτελεί την διπλωματική εργασία του κου. Ε. Σαραντάρη.



Σχήμα 1-1. Top Level μοντέλο front-end συστήματος & συστήματος θερμοκηπίων

1.1.3 Επικοινωνία Υποσυστημάτων

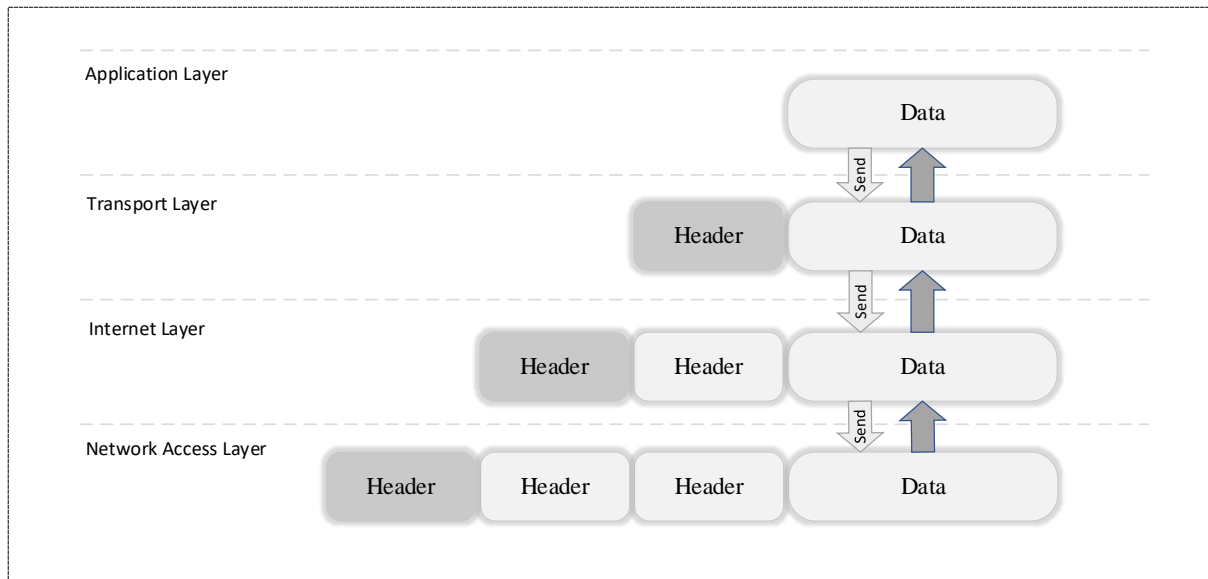
Η κάρτα διασύνδεσης ESP8266-01, λειτουργεί αρχικά ως Access point. Ο χρήστης εισάγει το SSID και το password που απαιτείται για την σύνδεση με το οικιακό δίκτυο. Η παραπάνω διαδικασία εκτελείται μόνο μια φορά. Έπειτα, η συσκευή παραμένει συνδεδεμένη στο συγκεκριμένο δίκτυο WiFi του router, και αποθηκεύει τα στοιχεία σύνδεσης. Η σύνδεση αυτή, πραγματοποιείται μέσω του πρωτοκόλλου IEEE 802.11. Έπειτα, η κάρτα διασύνδεσης, μπαίνει σε station mode, και συνδέεται στο modem/router του σπιτιού. Το θερμοκήπιο πλέον, μπορεί να συνδεθεί σε έναν Web Server με τον οποίο η επικοινωνία γίνεται με HTTP αιτήματα και χρησιμοποιώντας τις μεθόδους Get και Post[50]. Η σχεδίαση της επικοινωνίας παρουσιάζεται στην ενότητα 3.



Σχήμα 1-2. Επικοινωνία Server με το σύστημα των θερμοκηπίων μέσω modem/router

1.1.4 Κάρτα διασύνδεσης ESP8266-01 και TCP/IP

Όπως αναφέρθηκε και στην υποενότητα 1.1.3, το θερμοκήπιο διαθέτει την κάρτα διεπαφής TCP/IP, ESP8266-01. Η στοίβα πρωτοκόλλων TCP/IP, αποτελείται από μία συστοιχία τεσσάρων επιπέδων, σε κάθε ένα από αυτά προστίθεται και το αντίστοιχο πακέτο δεδομένων για να επιτευχθεί η δρομολόγηση [57].



Σχήμα 1-3. TCP/IP data encapsulation[57].

Τα διαφορετικά επίπεδα και τα επιμέρους πρωτόκολλα του καθενός, αναφέρονται παρακάτω.

- Network Access Layer

Είναι το χαμηλότερο επίπεδο της στοίβας και ορίζει την φυσική διασύνδεση με τις συσκευές. Τέτοιες διασυνδέσεις μπορεί να είναι Ethernet ή WiFi. Στο επίπεδο αυτό, υπάρχει και ο μηχανισμός ARP (address resolution protocol), που χρησιμοποιείται στις διευθυνσιοδοτήσεις IP των φυσικών συσκευών[57].

- Internet Layer

Στο επίπεδο αυτό, βρίσκεται το Internet Protocol (IP). Εκεί, πραγματοποιείται η διευθυνσιοδότηση των διαδικτυακών συσκευών. Όλα τα πακέτα που δρομολογούνται στο διαδίκτυο, περνούν από αυτό το επίπεδο[57].

- Transport Layer

Σε αυτό το επίπεδο, τα διαθέσιμα πρωτόκολλα είναι το TCP και το UDP. Το TCP προσφέρει αξιόπιστη μεταφορά δεδομένων από κόμβο σε κόμβο, με ανίχνευση και διόρθωση σφαλμάτων. Το UDP, προσφέρει μεταφορά δεδομένων με ανίχνευση σφαλμάτων αλλά όχι διόρθωση. Το UDP είναι αρκετά πιο γρήγορο, αλλά όχι τόσο αξιόπιστο όσο το TCP. Χρησιμοποιείται σε επικοινωνίες που επιτρέπουν μικρές αστοχίες μεταφοράς, όπως video streaming. Το TCP, προτιμάται σε εφαρμογές που δεν επιτρέπεται σφάλμα κατά την μεταφορά, όπως email και λήψεις ιστοσελίδων[57].

- Application Layer

Στην κορυφή του TCP/IP βρίσκεται το application Layer. Σε αυτό το επίπεδο, εμπεριέχονται όλες οι διαδικασίες που χρησιμοποιούν το Transport Layer για να αποστείλουν δεδομένα. Τα πιο διαδεδομένα πρωτόκολλα αυτού του επιπέδου, είναι το HTTP, το FTP, το SMTP, το OSC, και συστήματα όπως το DNS[57].

1.2 Σκοπός Διπλωματικής Εργασίας

Σκοπός της παρούσας διπλωματικής εργασίας, είναι η ανάπτυξη ενός πλήρους IoT συστήματος διαχείρισης και ελέγχου, για το θερμοκήπιο Leko, που θα ολοκληρώνει την διαδικασία αυτόνομης καλλιέργειας στο σπίτι.

Στόχος του συστήματος διαχείρισης είναι η συνεισφορά επιπρόσθετης αξίας στο υφιστάμενο θερμοκήπιο, μέσω της ευχρηστίας και της αξιοπιστίας. Η ευχρηστία της αλληλοεπίδρασης ορίζεται από την εύκολη παραμετροποίηση και την ολοκληρωμένη επίβλεψη του θερμοκηπίου από τον χρήστη. Η αξιοπιστία προκύπτει από τον μηχανισμό monitoring που παρέχει το σύστημα διαχείρισης.

Επιπροσθέτως, στα πλαίσια της διαδικασίας ανάπτυξης, σκοπός είναι η διερεύνηση τεχνολογιών, σχετικές με τα υποσυστήματα που συνθέτουν το σύστημα διαχείρισης, και κατόπιν η εξοικείωση με αυτές και ο κατάλληλος συνδυασμός τους.

Επίσης, σκοπός της παρούσας διπλωματικής, είναι η ποσοτικοποίηση του συστήματος που υλοποιήθηκε, εφαρμόζοντας κατάλληλες δοκιμές απόδοσης (performance tests), καταγράφοντας και αξιολογώντας τα αποτελέσματα. Τα αποτελέσματα της αξιολόγησης, θα κρίνουν και την ετοιμότητα του συστήματος για εξυπηρέτηση μεγάλου όγκου χρηστών και θερμοκηπίων, σε ενδεχόμενη εμπορική χρήση.

Κλείνοντας, αιώτερος σκοπός της παρούσας διπλωματικής εργασίας, είναι η προσφορά στην τεχνολογία και στην γεωργία, παρουσιάζοντας ένα αποκεντροποιημένο μοντέλο καλλιέργειας, από τον μεγάλο παραγωγό στον καθημερινό χρήστη.

1.3 Οργάνωση διπλωματικής εργασίας

Παρακάτω παρατίθενται τα κεφάλαια που περιλαμβάνονται στην παρούσα διπλωματική εργασία.

- Στο Κεφάλαιο 1 γίνεται μία εισαγωγή στο περιεχόμενο της διπλωματικής εργασίας καθώς και στον σκοπό της.
- Στο Κεφάλαιο 2 παρουσιάζεται η έρευνα που πραγματοποιήθηκε, παρόμοιες εφαρμογές, καθώς και σχετικές τεχνολογίες.
- Στο Κεφάλαιο 3 γίνεται η ανάλυση του συστήματος του θερμοκηπίου, των απαιτήσεων του χρήστη, και παρουσιάζονται οι σχεδιαστικές επιλογές του συστήματος που υλοποιήθηκε.
- Στο Κεφάλαιο 4 παρουσιάζεται η αρχιτεκτονική του συστήματος, με τα λειτουργικά στοιχεία που το απαρτίζουν.
- Στο Κεφάλαιο 5 παρουσιάζεται η σχεδίαση των υποσυστημάτων του συστήματος και οι επιλογές που έγιναν.
- Στο Κεφάλαιο 6 ελέγχεται η λειτουργία και οι δυνατότητες του συστήματος μέσω ενός case study.
- Στο Κεφάλαιο 7 ελέγχεται το σύστημα που υλοποιήθηκε σε διαφορετικά φορτία.
- Στο Κεφάλαιο 8 παρουσιάζονται τα συμπεράσματα που προέκυψαν καθώς και μελλοντικές βελτιώσεις του συστήματος.
- Στο Κεφάλαιο 9 παρουσιάζονται οι πηγές από τις οποίες αντλήθηκαν πληροφορίες για την υλοποίηση της παρούσας διπλωματικής.

2. ΣΧΕΤΙΚΗ ΕΡΕΥΝΑ

Στα συστήματα IoT, εμπλέκονται διαφορετικές τεχνολογίες, ενσωματωμένες σε διαφορετικά υποσυστήματα, με συγκεκριμένους ρόλους το καθένα. Κάθε ένα από αυτά, πρέπει να λειτουργεί αυτόνομα αλλά και συνεργατικά με τα υπόλοιπα. Στο κεφάλαιο αυτό, εξετάζονται μερικές από αυτές τις τεχνολογίες, καθώς και εφαρμογές τους.

2.1 IoT

Ο όρος “Internet of Things” επινοήθηκε πρώτη φορά από τον Kevin Ashton το 1999, στα πλαίσια μίας παρουσίασης για την χρήση RFID στο supply chain[3]. Το internet of Things περιγράφει το δίκτυο των συσκευών που έχουν την δυνατότητα να συνδέονται στο διαδίκτυο, συνήθως σε κάποιον Web Server, και να επικοινωνούν μεταξύ τους. Οι συσκευές αυτές διαθέτουν αισθητήρες και ενεργοποιητές, για να προσφέρεται στον χρήστη η δυνατότητα επίβλεψης και ελέγχου, ελαχιστοποιώντας την δικιά του παρέμβαση[1]. Ένας άλλος πιο απλός ορισμός για το Internet of Things, είναι το αποτέλεσμα συνδεδεμένων πραγμάτων στο διαδίκτυο, που δεν λειτουργούν με την άμεση παρέμβαση του χρήστη [2].

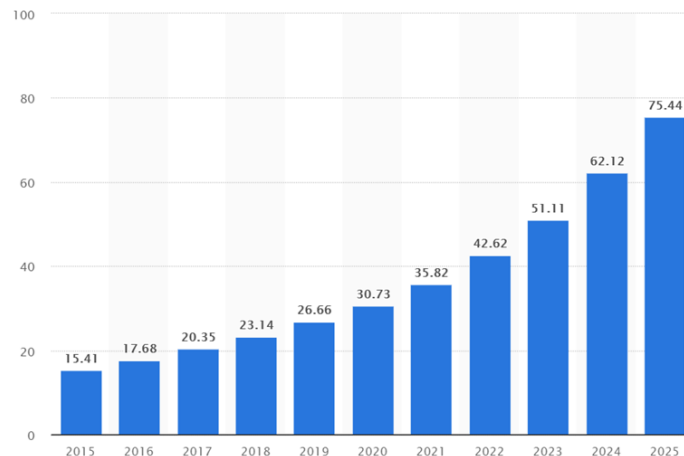
Οι εφαρμογές του IoT αφορούν πολλούς τομείς. Η κατηγοριοποίηση της κάθε εφαρμογής, γίνεται με βάση τον τύπο της διασύνδεσης, την κάλυψη, την κλίμακα, την συμμετοχή των χρηστών και την επίδραση της.

Οι βασικότερες κατηγορίες εφαρμογών IoT είναι:

- οι προσωπικές ή σπιτιού
- οι επαγγελματικές
- οι λειτουργικότητας
- οι κινητές[4].

Στην κατηγορία προσωπικής χρήσης εμπίπτουν εφαρμογές υγείας και διαχείριση οικιακών και άλλων συσκευών. Στις επαγγελματικές εφαρμογές, υπόκειται η χρήση διασυνδεδεμένων αισθητήριων για έλεγχο και συντήρηση συστημάτων παραγωγής, για ενίσχυση ασφαλείας, κλιματική διαχείριση και συστήματα εξοικονόμησης. Στις εφαρμογές λειτουργικότητας υπόκεινται έξυπνοι μετρητές κατανάλωσης ηλεκτρικής ενέργειας, ανάλυση και επεξεργασία video. Τέλος στην κατηγορία των κινητών εφαρμογών, υπόκεινται εφαρμογές που αφορούν κυρίως την βελτιστοποίηση του supply chain, όπως η χρήση WSNs για καταγραφή, ανάλυση ή αυτοματοποίηση μεταφοράς και αποθήκευσης αγαθών[4].

Όπως φαίνεται και στην Εικόνα 2-1, η διεισδυτικότητα του IoT στην αγορά είναι εκθετική, προσφέροντας μία πληθώρα εφαρμογών που χρησιμοποιούνται για την βελτιστοποίηση ευρέων διαδικασιών. Για τον λόγο αυτό, η εποχή του IoT και του Big Data, έχει χαρακτηριστεί και ως η 4^η βιομηχανική επανάσταση.



Εικόνα 2-1. IoT users trend[51].

Ο συνολικός αριθμός των συνδεδεμένων “έξυπνων” συσκευών προβλέπεται να ανέλθει σε 75,44 δισεκατομμύρια παγκοσμίως έως το 2025, δηλαδή πενταπλάσια αύξηση σε δέκα χρόνια. Το IoT, το οποίο ενισχύεται από την ήδη εξελισσόμενη τεχνολογία διαδικτύου, είναι το επόμενο σημαντικό βήμα για το διαδίκτυο και την τεχνολογία τις επόμενες δεκαετίες [51].

2.1.1 Αρχιτεκτονικές δικτύων IoT

Τα δίκτυα των συστημάτων IoT διακρίνονται σε τρεις αρχιτεκτονικές,

- την τοπολογία αστέρα
- την απευθείας σύνδεση (P2P)
- και την τοπολογία cluster tree[44].

Στην τοπολογία αστέρα όλες οι συσκευές συνδέονται σε μία κεντρική, που παίζει τον ρόλο του συντονιστή. Είναι η πιο συνηθισμένη από τις τρεις τοπολογίες. Σε αυτήν την τοπολογία, το θετικό είναι, ότι πέρα του συντονιστή, οι υπόλοιπες συσκευές μπορούν να είναι χαμηλότερης υπολογιστικής ικανότητας, χωρίς την δυνατότητα δρομολόγησης[44].

Στην P2P αρχιτεκτονική η επικοινωνία είναι άμεση μεταξύ των συσκευών και δεν είναι απαραίτητη η παρουσία κεντρικού κόμβου. Αποτελεί επέκταση της τοπολογίας αστέρα.

Η τοπολογία cluster tree είναι παρόμοια με την τοπολογία αστέρα, με την διαφορά ότι οι συσκευές περιορισμένης λειτουργικότητας συνδέονται στις εξωτερικές διακλαδώσεις, και όχι απευθείας στον κεντρικό κόμβο. Εφαρμόζεται σε δίκτυα μεγάλης κλίμακας[44].

2.1.2 Επίπεδα IoT

Το IoT δεν αφορά μόνο την σύνδεση συσκευών μεταξύ τους. Η διασύνδεση είναι απλά ένα κομμάτι από τα τέσσερα επίπεδα που το καθορίζουν[5]. Πολλοί ερευνητές, ορίζουν το IoT σε τρία επίπεδα αντί για τέσσερα [45][46]. Ο διαχωρισμός σε επίπεδα, επιδιώκει να επεξηγήσει τον τρόπο με τον οποίο οι διάφορες τεχνολογίες σχετίζονται μεταξύ τους και να γνωστοποιήσουν την επεκτασιμότητα, τη διαμόρφωση και τη διαμόρφωση των εφαρμογών IOT σε διάφορα σενάρια. Για την περαιτέρω ανάλυση, και λόγω του γεγονότος ότι η επιστήμη του Data Analytics έχει ενσωματωθεί με την έννοια του IoT, αναλύονται και τα τέσσερα επίπεδα. Τα επίπεδα αυτά είναι:

- Επίπεδο έξυπνης συσκευής / αισθητήρα

Το χαμηλότερο επίπεδο αποτελείται από έξυπνα αντικείμενα, που είναι ενσωματωμένα σε αισθητήρες. Οι αισθητήρες επιτρέπουν τη διασύνδεση του φυσικού και του ψηφιακού κόσμου, επιτρέποντας τη συλλογή και επεξεργασία πληροφοριών σε πραγματικό χρόνο [5]. Στο επίπεδο αυτό, συγκεντρώνονται τα δεδομένα, τα οποία έπειτα περνάνε στο επόμενο επίπεδο Διασύνδεσης[45].

- Επίπεδο Διασύνδεσης

Το επίπεδο διασύνδεσης, που βρίσκεται πάνω από το επίπεδο συσκευής, συνδέει τις συσκευές αυτές μεταξύ τους, ασύρματα ή ενσύρματα[5]. Μερικές από τις πιο πρόσφατες τεχνολογίες διασύνδεσης είναι το WiFi, LTE, Bluetooth, 3G, Zigbee κλπ. [46]. Αυτά τα δίκτυα μπορούν να έχουν τη μορφή ιδιωτικών, δημόσιων ή υβριδικών μοντέλων και είναι κατασκευασμένα για να υποστηρίζουν τις απαιτήσεις επικοινωνίας για λανθάνουσα κατάσταση, εύρος ζώνης ή ασφάλεια [6].

Αξίζει να σημειωθεί ότι το επίπεδο αυτό, διασπάται μεμονωμένα και σε περισσότερα επίπεδα διασύνδεσης με διαφορετικές ιεραρχίες. Στην τοπολογία αστέρα για παράδειγμα, όλες οι διασυνδέσεις ανήκουν στο επίπεδο διασύνδεσης, ο κεντρικός κόμβος όμως, ανήκει στο υψηλότερο επίπεδο των επιμέρους διασυνδέσεων.

- Επίπεδο Data Analytics

Στο επίπεδο αυτό, τα δεδομένα που έχουν συλλεχθεί από τα προηγούμενα επίπεδα αναλύονται για την παραγωγή χρήσιμων συμπερασμάτων και πληροφοριών[5].

- Επίπεδο Εφαρμογής και υπηρεσίας

Στο επίπεδο υπηρεσίας, εάν μεσολαβεί και το επίπεδο του Data Analytics, είναι το επίπεδο όπου παίρνονται οι αποφάσεις με βάση τις πληροφορίες που έχουν εξαχθεί[5]. Είναι συνδεδεμένο με το επίπεδο εφαρμογής, που αφορά την τελική υλοποίηση του IoT συστήματος και την δημιουργία ενός έξυπνου περιβάλλοντος[45].

2.1.3 Πρωτόκολλα IoT

Τα IoT δίκτυα απαρτίζονται από πολλά πρωτόκολλα διασύνδεσης, τα οποία χρησιμοποιούνται αναλόγως την εφαρμογή και το επίπεδο επικοινωνίας. Οι τεχνολογίες επικοινωνίας μεταξύ των συσκευών παίζουν σημαντικό ρόλο. Ο σχεδιασμός ενός συστήματος IoT, θα πρέπει να εστιάζει στις συσκευές που το απαρτίζουν και τις δυνατότητες τους, καθώς το δίκτυο χρησιμοποιείται απλά ως δίαυλος επικοινωνίας[46]. Σε αυτό το πλαίσιο σχεδιασμού, έχουν επικρατήσει πρωτόκολλα επικοινωνίας που αρμόζουν σε συσκευές περιορισμένης υπολογιστικής ικανότητας και σε δίκτυα με χαμηλό bandwidth.

Στο κατώτερο επίπεδο των συσκευών, χρησιμοποιούνται ασύρματα πρωτόκολλα ραδιοκυμάτων, Bluetooth Low Energy (BLE), Power Line Communication (PLC), Radio Frequency Identification (RFID), Near Field Communication (NFC), Radio Frequency Identification (RFID), ή ZigBee[47]. Ανεξαρτήτως των πρωτοκόλλων που χρησιμοποιούνται στο κατώτερο επίπεδο, η τερματική συσκευή θα πρέπει να δέχεται και να στέλνει δεδομένα στο διαδίκτυο [7].

Τα βασικότερα πρωτόκολλα του υψηλότερου επιπέδου, που χρησιμοποιούνται σε IoT εφαρμογές, είναι τα **HTTP**, **MQTT**, **CoAP** και **AMQP** [8].

- HTTP

Το HTTP είναι το επικρατέστερο πρωτόκολλο στο διαδίκτυο σήμερα, σε επίπεδο application. Υποστηρίζει την request/ response αρχιτεκτονική, και η διευθυνσιοδότηση που χρησιμοποιεί βασίζεται στα URIs (Universal Resource Identifiers) [10]. Η σύσταση ενός request, αποτελείται από την μέθοδο, τον πόρο, τα HEADER και το περιεχόμενο. Στο πρωτόκολλο αυτό, η συσκευή λειτουργεί ως client και αποστέλλει τις μετρήσεις στον server που παίζει τον ρόλο του διαμεσολαβητή, μεταξύ χρήστη και συσκευής. Με αυτό το μοντέλο επικοινωνίας, η συσκευή δεν δέχεται αιτήματα από τον server, πέραν των responses των requests [6].

- UPnP

Η αρχιτεκτονική UPnP, βασίζεται κατά κύριο λόγο στο πρωτόκολλο HTTP, και μπορεί να εφαρμοστεί στο TCP αλλά και στο UDP. Πρόκειται για μία αρχιτεκτονική συσκευών, με την οποία οι συσκευές μπορούν να συνδεθούν μεταξύ τους, και να κοινοποιήσουν στο δίκτυο τα χαρακτηριστικά και τις δυνατότητές τους, περιμένοντας κάποιο αίτημα. Κάθε συσκευή που συνδέεται στο δίκτυο κοινοποιεί ένα Device Description Document (DDD) για τους υπόλοιπους συμμετέχοντες στο δίκτυο αυτό[6]. Αυτή η αρχιτεκτονική είναι ιδανική για συσκευές που συνδέονται σε δίκτυα με μη καθορισμένη τοπολογία και διαπιστευτήρια όπως το οικιακό, που βασίζεται στην διευθυνσιοδότηση IPv4 και IPv6.

Την ανιχνευσιμότητα των συσκευών στο δίκτυο, υποστηρίζουν και τα πρωτόκολλα MQTT και CoAP που ακολουθούν.

- MQTT

Το MQTT είναι από τα πρώτα M2M πρωτόκολλα επικοινωνίας. Αναπτύχθηκε το 1999 από την IBM[10]. Είναι ένα πρωτόκολλο που βασίζεται στο publish/subscribe μοντέλο επικοινωνίας σχεδιασμένο για συσκευές περιορισμένων πόρων. Ο MQTT client κάνει publish μηνύματα στον MQTT broker στον οποίο είναι subscribed και άλλοι clients. Το κάθε μήνυμα γίνεται published σε μία διεύθυνση, που ονομάζεται topic. Οι clients μπορούν να κάνουν subscribe σε πολλαπλά topic, και να λαμβάνουν τα μηνύματα που γίνονται publish σε αυτά [11].

- CoAp

Το CoAP (Constrained application Protocol) υποστηρίζει και το publish/subscribe μοντέλο που περιεγράφηκε παραπάνω και το resource/observe μοντέλο επικοινωνίας, που είναι μία παραλλαγή του προηγούμενου[12]. Σε αντίθεση με το MQTT, όπως και το HTTP χρησιμοποιεί URIs έναντι των topics. Το CoAP, σχεδιάστηκε κυρίως για να συνυπάρχει με το HTTP πρωτόκολλο που ορίζει παρόμοιας φιλοσοφίας μοντέλο [13].

2.1.4 Θέματα ασφαλείας στις IoT εφαρμογές

Το οικοσύστημα των IoT υπηρεσιών έχει γίνει πόλος έλξης κακόβουλων επιθέσεων τα τελευταία χρόνια [25]. Botnets όπως το Mirai έχουν οδηγήσει μαζικές επιθέσεις σε συσκευές IoT. Τα botnet αυτά, ανιχνεύουν διευθύνσεις IP συσκευών με προκαθορισμένα διαπιστευτήρια. Έπειτα, φορτώνουν σε αυτές τις συσκευές κακόβουλα λογισμικά, τα οποία δέχονται συντονισμένα εντολές επιθέσεων σε συγκεκριμένες IP. Με αυτόν τον τρόπο, δημιουργείται μία ομάδα συσκευών που μπορούν να εκτελέσουν ομαδικές συντονισμένες επιθέσεις.

Ο αριθμός των συσκευών με το malware, έχουν φτάσει σε περιπτώσεις και τις 400.000, και διατίθενται στο dark web και για ενοικίαση σε υποψήφιους attackers. Η μεγαλύτερη επίθεση από

το Mirai, έχει δημιουργήσει traffic του 1.1 Tbps. Ικανό για να ρίξει τα περισσότερα web services. Η ευαισθησία των IoT συστημάτων, επιβάλλει προσαρμογές από τους κατασκευαστές των συσκευών, σε ισχυρότερα και εξατομικευμένα για κάθε συσκευή διαπιστευτήρια, και συχνές ενημερώσεις ασφαλείας[26].

2.2 Γεωργία ακριβείας

Ένας σύντομος ορισμός για την γεωργία ακριβείας ή αλλιώς precision agriculture είναι, σύμφωνα με τον J.V. Stafford, “η ρύθμιση των εισόδων μίας γεωργικής παραγωγής, με βάσει τις απαιτήσεις της τοπικά”[14]. Όπως αναφέρει ο J. Pierce, σκοπός της γεωργίας ακριβείας είναι να προσαρμόσει τις γεωργικές εισόδους και πρακτικές, στις συνθήκες που επικρατούν σε μία καλλιέργεια, ώστε να κάνουν το σωστό πράγμα, στο σωστό μέρος, την σωστή στιγμή και με τον σωστό τρόπο [15].

Η γεωργία ακριβείας, βασίζεται σε πολλές τεχνολογίες, διασύνδεσης, υπολογισμού, Big Data και machine learning. Η συνεργασία αυτών αποσκοπούν στην βελτιστοποίηση της γεωργικής παραγωγής.

2.2.1 Εφαρμογές IoT στην γεωργία

Η ανάπτυξη πολλών τεχνολογιών την τελευταία δεκαετία, έχει εμφανή επίδραση στην γεωργία ακριβείας. Πέρα από υλοποιήσεις που αφορούν παραγωγικές μονάδες, η τάση έχει μεταφερθεί και στο smart farming που αφορά καλλιέργειες μικρότερης κλίμακας. Οι επαγγελματικές εφαρμογές μπορεί να απαιτούν πιο αυστηρούς χρονισμούς και να παρουσιάζουν εντονότερα το στοιχείο των Data Analytics σε σχέση με εφαρμογές οικιακής χρήσης ή μικρότερης κλίμακας. Οι προδιαγραφές όμως παραμένουν περίπου οι ίδιες, όσο αναφορά την διασύνδεση των συσκευών στο δίκτυο.

- Εφαρμογές μικρής κλίμακας

Οι εφαρμογές αυτές αφορούν κυρίως προϊόντα εμπορικού σκοπού με αποδέκτες τους τελικούς καταναλωτές. Τα τελευταία χρόνια έχουν κυκλοφορήσει στην αγορά διάφορα smart farming συστήματα οικιακής χρήσης. Τα δημοφιλέστερα, είναι το Clickandgrow [59], το Plantui [60], το Leaf[61] και το Niwa[62]. Τα πρώτα δύο προϊόντα δεν αποτελούν όλα αυστηρά εφαρμογές της γεωργίας ακριβείας, αλλά υβριδικές εφαρμογές. Λειτουργούν με κάψουλες που προμηθεύει η αντίστοιχη εταιρεία, όπου μέσω των διατομών τους ελέγχεται η χορήγηση νερού. Το Niwa και το Leaf, που είναι και το ακριβότερο με διαφορά μεταξύ τους, προσφέρει πλήρη παρέμβαση και έλεγχο στις περιβαλλοντικές τιμές της καλλιέργειας.

Ανάμεσα στα παραπάνω προϊόντα, το θερμοκήπιο Leko, αποτελεί μία εφαρμογή με περισσότερες επιλογές καλλιέργειας από τα συστήματα καψουλών, και σε αρκετά ανταγωνιστικότερη τιμή από τα υπόλοιπα.

- Εφαρμογές μεγάλης κλίμακας ή ερευνητικού σκοπού

Οι έρευνες που έχουν γίνει σε ερευνητικό επίπεδο και αφορούν την γεωργία ακριβείας είναι πολλές. Σε γενικές γραμμές, όταν αφορούν εφαρμογές μεγάλης κλίμακας, αποτελούνται από WSNs. Τα πρωτόκολλα που χρησιμοποιούνται στις εφαρμογές αυτές είναι πολλά και τα περισσότερα είναι γνωστά και έχουν αναφερθεί παραπάνω, στην υποενότητα 2.1.3. Ένα ακόμα πρωτόκολλο που χρησιμοποιείται σε τέτοιες εφαρμογές είναι το Lora, που είναι παρόμοιο με το GSM, πιο μικρού βεληνεκούς (έως 15 χλμ.), αλλά πολύ πιο αποδοτικό στην κατανάλωση της μπαταρίας (έως 20 χρόνια). Αποτελείται από ένα module πομποδέκτη και ένα gateway [16].

- **Τεχνολογία Backscatter**

Στις εφαρμογές μεγάλης κλίμακας, συναντάται και η τεχνολογία backscatter. Το backscatter βασίζεται σε αρχές ανάκλασης, όπου ένας κόμβος διαμορφώνει την πληροφορία στην κορυφή ενός λαμβανόμενου σήματος, συνδέοντας την κεραία του σε διαφορετικά φορτία. Η εξαιρετικά χαμηλή κατανάλωση, σε συνδυασμό με τις μεγάλες αποστάσεις που μπορεί να στηθεί το δίκτυο, καθιστούν την τεχνολογία αυτή ιδανική για καταναλώσεις επιπέδου μW ή λιγότερο και δίκτυα IoT[60]. Στα δίκτυα που λειτουργούν με backscattering τεχνολογία, εφαρμόζονται “energy harvesting” τεχνικές, για χρήση κόμβων χωρίς μπαταρία. Κάποιες από τις τεχνικές αυτές, υλοποιούνται με χρήση φωτοδιόδου, ή RF energy harvesting με χρήση ευαίσθητου ορθόκενου, και χρησιμοποιώντας μονάδες ενίσχυσης ενέργειας, για την κάλυψη της απαιτούμενης κατανάλωσης[61].

Εκτενής μελέτη έχει γίνει στην αξιοποίηση των δεδομένων που προκύπτουν σε επίπεδο Data Analytics. Σε πολλές περιπτώσεις τα δεδομένα αυτά, χρησιμοποιούνται για δημιουργία μοντέλων με machine learning [19] [20] [22], ή reinforcement learning [17]. Η μεταφορά των δεδομένων από τον κεντρικό κόμβο συνήθως γίνεται είτε με GSM [18], είτε σε κάποιο data center πλησίον της εφαρμογής, είτε με σύνδεση στο διαδίκτυο χρησιμοποιώντας LTE/3G/4G ή WiFi.

Η πρόβλεψη συγκομιδής συνήθως γίνεται με τεχνικές image processing, αλλά η επιλογή των input μπορεί να είναι και άλλη, όπως το βάρος της καλλιέργειας [21]. Η επιλογή του image processing είναι πιο κατάλληλη, καθώς μέσω machine learning εξυπηρετεί και εφαρμογές εντοπισμού πιθανών ασθενειών, με καλά αποτελέσματα [23][24].

2.3 Έρευνα Τεχνολογιών

Τα επιμέρους δομικά στοιχεία που απαρτίζουν το σύστημα διαχείρισης που αφορά την παρούσα διπλωματική, είναι:

- η **βάση δεδομένων**, στην οποία αποθηκεύονται οι τιμές των φυτών και τα στοιχεία των θερμοκηπίων και των χρηστών στα οποία ανήκουν
- το **API**, το οποίο αλληλοεπιδρά με τα υπόλοιπα στοιχεία
- και το **Android**, μέσω του οποίου ο χρήστης αλληλοεπιδρά με το σύστημα

Στις επόμενες υποενότητες ακολουθεί μία τεχνολογική ανάλυση των επικρατέστερων μεθόδων και εργαλείων που εφαρμόζονται για την υλοποίηση καθενός από τα παραπάνω δομικά στοιχεία του υλοποιηθέντος συστήματος.

2.3.1 Σχεσιακή & Μη-σχεσιακή λογική βάσεως δεδομένων

Οι εφαρμογές IoT παράγουν μεγάλο πλήθος δεδομένων. Ουσιαστικά, το IoT αφορά λιγότερο τις ίδιες τις συσκευές και περισσότερο τα δεδομένα που δημιουργούν και συλλέγουν. Τα δεδομένα πρέπει να είναι προσπελάσιμα από πολλούς χρήστες, εσωτερικούς και εξωτερικούς. Εσωτερικούς χρήστες για αναλύσεις δεδομένων, εξωτερικούς χρήστες για αποθήκευση και ανάκτηση δεδομένων σε πραγματικό χρόνο.

Για την αποθήκευση και διαχείριση των δεδομένων αυτών, απαιτείται κάποια βάση δεδομένων. Το βασικότερο χαρακτηριστικό που διακρίνει την βάση δεδομένων είναι η λογική της δομής. Οι δύο βασικές δομές βάσης δεδομένων είναι:

- της σχεσιακής λογικής, και
- μη-σχεσιακής λογικής

Το Database Management System, μπορεί να είναι είτε σχεσιακό, είτε μη σχεσιακό. Τα πιο διαδεδομένα συστήματα σχεσιακής και μη σχεσιακής λογικής είναι η MySQL και η MongoDB αντίστοιχα, και αναλύονται παρακάτω.

Οι σχεσιακές βάσεις χρησιμοποιούνται ευρέως για την διαχείριση περιορισμένων σε όγκο δεδομένων, που αφορά δομημένη πληροφορία[33]. Τα σχεσιακά συστήματα όπως η SQL, επικεντρώνονται στην συνέπεια και χαρακτηρίζονται από τις ιδιότητες ACID (atomicity, consistency, isolation, durability)[31]. Με τις σχεσιακές βάσεις δεδομένων είναι απαραίτητη η χρήση σχεσιακών σχημάτων για τη μοντελοποίηση των δεδομένων, όπου ο σωστός σχεδιασμός, είναι απαραίτητη προϋπόθεση για μία λειτουργική δομή χωρίς τεχνικές δυσκολίες. Οι αποφάσεις σχεδιασμού σχετικά με τον τρόπο αποθήκευσης των δεδομένων, είναι δύσκολο να αλλάξουν στο μέλλον. Παρόλο την περιορισμένη ευελιξία, η δημιουργία σχήματος εκ των προτέρων με την λογική της SQL επιτρέπει την αποδοτικότερη εκτέλεση πολύπλοκων ερωτημάτων. Στην διαδικασία του σχεδιασμού είναι σημαντικό να υποδειχθούν σωστά τα δεδομένα για να βελτιωθεί η απόδοση του συστήματος. Η δημιουργία των κατάλληλων ευρετηρίων και σχημάτων πίνακα για ένα δεδομένο φόρτο εργασίας μπορεί να έχει ως αποτέλεσμα σημαντικές βελτιώσεις απόδοσης. Αντίστροφα, ο σχεδιασμός ενός λανθασμένου σχήματος μπορεί να οδηγήσει σε σημαντική υποβάθμιση της απόδοσης.

Οι μη σχεσιακές βάσεις από την άλλη, υπερτερούν στην διαχείριση μη δομημένων δεδομένων όπως documents ή multimedia[33]. Ένα από τα πλεονεκτήματα της μη σχεσιακής βάσης είναι η επεκτασιμότητα της. Η επεκτασιμότητα της είναι οριζόντια, που σημαίνει ότι το φορτίο του συστήματος, μπορεί να διαμοιραστεί σε πολλούς διαφορετικούς server ταυτόχρονα. Το χαρακτηριστικό αυτό καθιστά ένα σημαντικό πλεονέκτημα σε εκθετικά αυξανόμενα συστήματα, αφού με αυτόν τον τρόπο μειώνεται το κόστος της αρχικής επένδυσης[31].

Η μετάβαση της διαχείρισης δεδομένων σε δεύτερο χρόνο, από σχεσιακή σε μη σχεσιακή λογική, είναι εφικτή. Η κλίμακα των δεδομένων, η δομή τους, και οι δυνατότητες επεκτασιμότητας είναι οι σημαντικότεροι παράγοντες που λαμβάνονται υπόψιν. Μία τέτοια μετάβαση παρουσιάζει σημαντικές προκλήσεις, που όμως μπορούν να αντιμετωπιστούν με την κατάλληλη μοντελοποίηση των σχέσεων μεταξύ των οντοτήτων[32].

Η σύγκριση της σχεσιακής και μη σχεσιακής διαχείρισης, δείχνει και σε διάφορα πειράματα σε εφαρμογές IoT ότι είναι κοντινή [31], με την MySQL να είναι πιο σταθερή στους χρόνους εκτέλεσης από την MongoDB.

2.3.2 Αρχιτεκτονικές API & Server Scripting Languages

Το API (application programming interface), καθορίζει την λειτουργικότητα μίας διαδικτυακής εφαρμογής ή υπηρεσίας. Στο API, ορίζονται οι διαδικασίες και οι μέθοδοι του συστήματος, τα αιτήματα καθώς και ο τρόπος που εξυπηρετούνται, ο τύπος των δεδομένων που δέχεται το σύστημα ως εισόδους και παράγει ως εξόδους.

Ένα backend, μπορεί να αποτελείται από ένα, ή περισσότερα APIs. Τα APIs αυτά, μπορεί να είναι αποκλειστικά εσωτερικά, χωρίς να έχει πρόσβαση κάποιος εξωτερικός client, ή δημόσια διαθέσιμο για συγκεκριμένους ή όλους τους εξωτερικούς clients.

Κάθε API είναι διαφορετικό, προσφέρει και να εξυπηρετεί συγκεκριμένες λειτουργίες. Η δομή ενός API, χαρακτηρίζεται από την αρχιτεκτονική του. Οι δύο επικρατέστερες αρχιτεκτονικές API, είναι:

- η Rest
- και η SOAP

Το REST, ή αλλιώς representational State Transfer, είναι μία αρχιτεκτονική επικοινωνίας που χρησιμοποιείται σε διαδικτυακές εφαρμογές και web services. Το REST API βασίζεται στην διευθυνσιοδότηση URI (Uniform Resource Identifier). Αυτό προσφέρει ευελιξία, γιατί διασπά την επικοινωνία μεταξύ client και Server σε υπομονάδες, καθεμία από τις οποίες, είναι υπεύθυνη για ξεχωριστό κομμάτι της επικοινωνίας και μοναδικά προσπελάσιμη. Είναι υποχρέωση του client να γνωρίζει την URL διεύθυνση της κάθε μεθόδου που θα καλέσει για να εξυπηρετήσει το αίτημά του[58]. Βασίζεται στο πρωτόκολλο HTTP, για την μετάδοση των δεδομένων στο application layer. Οι απαντήσεις της REST υπηρεσίας είναι σε μορφή JSON ή XML [34]. Τα μηνύματα του REST είναι μικρότερα σε μέγεθος, γεγονός που ελευθερώνει bandwidth και το κάνει “οικονομικότερο”. Επιπλέον, προσφέρει μεγαλύτερη ευελιξία στην σχεδίαση [34].

Η αρχιτεκτονική SOAP, ή αλλιώς Simple Object Access Protocol, βασίζεται και αυτή συνήθως στο πρωτόκολλο HTTP, αλλά μπορεί να εφαρμοστεί και σε άλλα πρωτόκολλα, όπως SMTP ή JMS. Απαντάει μόνο σε XML μορφή [35]. Το SOAP παρουσιάζει ευελιξία στις γλώσσες, και το framework που μπορεί να στηθεί. Επιπλέον, λειτουργεί αποδοτικά σε διαμοιρασμένα περιβάλλοντα, σε αντίθεση με το REST που βασίζεται σε point-to-point επικοινωνίες.

Όσον αφορά τις IoT εφαρμογές, προτιμάται συνήθως η REST αρχιτεκτονική. Αποτελέσματα πειραμάτων, υποδεικνύουν ότι το REST χρησιμοποιεί λιγότερο bandwidth έναντι του SOAP, προσφέρει ευκολότερη πρόσβαση στους πόρους και τις υπηρεσίες της εφαρμογής, έχει μικρότερο request latency, και χαμηλότερη κατανάλωση CPU[34].

- **Server Scripting Language**

Η server scripting language αποτελεί την γλώσσα στην οποία υλοποιείται ένα API. Οι πιο δημοφιλής είναι η PHP, Python, Ruby, Asp.net, Javascript, C και Perl.

Η PHP είναι η δημοφιλέστερη, και μαζί με την Ruby, είναι από τις πιο αποδοτικές, με κριτήρια τον αριθμό γραμμών κώδικα και τους χρόνους εκτέλεσης[59]. Η PHP πέρα από την συχνή χρήση της σε RESTful περιβάλλοντα, παρουσιάζει καλύτερες αποδόσεις και σε SOAP, συγκριτικά με scripting γλώσσες όπως Java και C [37].

2.3.3 Τεχνικές API authentication

Το API είναι ένα θεμελιώδες σύστημα, με ένα εξωτερικό δίκτυο, ικανό να δέχεται και να εξυπηρετεί αιτήματα από διάφορους clients, που αφορούν προστατευμένες βάσεις δεδομένων. Πρέπει λοιπόν, να διαμορφωθεί έτσι ώστε να διασφαλίζει ότι οι εφαρμογές και οι clients που προσπαθούν να αποκτήσουν πρόσβαση σε δεδομένα είναι εξουσιοδοτημένοι, έτσι ώστε να μπορούν να προχωρήσουν σε πλήρη επιτρεπόμενη πρόσβαση μόλις επαληθευτεί η ταυτότητα τους. Οι διαδικασίες εξουσιοδότησης της ταυτότητας ενός ατόμου που προσπαθεί να αποκτήσει πρόσβαση σε υπηρεσίες στον διακομιστή ορίζεται ως έλεγχος ταυτότητας API ή API authentication[40].

Για το authentication του API υπάρχουν πολλοί μέθοδοι, διαφορετικής πολυπλοκότητας και προστασίας. Οι δημοφιλέστεροι είναι οι μέθοδοι JWT, OAuth και HTTP header.

- Το JWT είναι ένα σύστημα authentication βασισμένο σε κρυπτογραφημένα διακριτικά τύπου JSON, στην base64 κρυπτογράφηση. Το JWT έχει τρία τμήματα, όπως payload, header και signature και κάθε ένα χωρίζεται από τον dot operator. Ο header ορίζει το είδος της διαδικασίας διακριτικού και κατακερματισμού που θα χρησιμοποιηθεί για την signature JWT. Το payload αφορά τις πληροφορίες που μεταφέρονται. Οποιαδήποτε τέτοια τιμή που περιλαμβάνει αυτές τις τρεις ενότητες μπορεί να χρησιμοποιηθεί ως διακριτικό και αναφέρεται ως JWT[41].

- Το OAuth είναι ίσως το δημοφιλέστερο πρωτόκολλο, που αντιμετωπίζει το πρόβλημα της αποθήκευσης πόρων από τον κάτοχο πόρων χωρίς να απαιτείται τρίτος. Οι πόροι ανήκουν σε έναν κάτοχο πόρων που καλείται RO (Resource Owner). Επιτρέπει στον RO να ανταλλάσσει τους ιδιωτικούς πόρους του, όπως εικόνες που είναι αποθηκευμένες σε έναν απομακρυσμένο server σε άλλον webserver, παρά την παροχή των διαπιστευτηρίων[42].
- Η τεχνική του API key authentication με HTTP header, εξαλείφει το πρόβλημα της χρήσης κοινών διαπιστευτηρίων, απαιτώντας την ανάκτηση μοναδικού κλειδιού από το API. Εδώ το κλειδί είναι μια μεγάλη σειρά αριθμών και γραμμάτων που είναι ανεξάρτητα από τα διαπιστευτήρια του κατόχου. Ο client πιστοποιείται από το κλειδί API που του έχει δοθεί, οπότε ο server του επιτρέπει να έχει πρόσβαση στα δεδομένα. Η ευελιξία σε αυτό το σύστημα, έγκειται στο γεγονός, ότι μέσω του authentication επιτυγχάνεται η εποπτεία των χρηστών, προστατεύοντας παράλληλα τους κωδικούς πρόσβασης του χρήστη[43].

2.3.4 Αποστολή Notifications σε android

Τα notifications, χρησιμοποιούνται στο λειτουργικό android, για την ειδοποίηση κάποιας πληροφορίας στον χρήστη. Οι περισσότερες εφαρμογές πλέον, πλαισιώνονται με κάποιον μηχανισμό notification, για διάφορους σκοπούς, όπως διαφημιστικούς, ενημερωτικούς, υπενθυμίσεις και άλλα.

Υπάρχουν πολλές υπηρεσίες αποστολής notification στο διαδίκτυο, είτε open source είτε με πληρωμή. Ο πιο δημοφιλής μηχανισμός, είναι η σουίτα Firebase της Google. Με την σουίτα αυτή, η λειτουργία των ειδοποιήσεων μπορεί να υλοποιηθεί στον server του συστήματος, χρησιμοποιώντας τα APIs του FCM (firebase cloud messaging) για την αποστολή των μηνυμάτων. Αυτή η αρχιτεκτονική είναι ευέλικτη καθώς κάνει ευκολότερη την διαχείριση των χρηστών, την κατηγοριοποίηση και σύνδεσή τους σε ομάδες συσκευών και την αποστολή μηνυμάτων βασισμένη σε εξειδικευμένα γεγονότα. Η στοχοποίηση των συσκευών που θα λάβουν κάποιο notification, πραγματοποιείται με το διακριτικό εγγραφής που παράγει το FCM API, το οποίο είναι μοναδικό για κάθε συσκευή[27]. Ένα από τα πλεονεκτήματα του Firebase, είναι ότι χρησιμοποιεί ένα TCP socket για την επικοινωνία με τον Server της Firebase, ανεξαρτήτως των εφαρμογών που χρησιμοποιούν την υπηρεσία. Επομένως εάν ο χρήστης έχει ήδη κάποια άλλη εφαρμογή που χρησιμοποιεί firebase, δεν θα απαιτείται επιπλέον socket και επομένως εξοικονομείται σημαντική απώλεια ενέργειας για την μπαταρία της κινητής συσκευής

2.4 Λειτουργικό σύστημα Android

Η λέξη Android αναφέρεται σε ένα λειτουργικό σύστημα για κινητά τηλέφωνα και όχι μόνο. Είναι βασισμένο στο λειτουργικό Linux και αναπτύσσεται από την Google. Πρόκειται για λογισμικό ανοιχτού κώδικα (Open Source Software - OSS), το οποίο σημαίνει ότι ο πηγαίος κώδικάς του διατίθεται ελεύθερα σε όσους ζητούν να τον εξετάσουν ή να τον τροποποιήσουν για να τον αξιοποιήσουν σε άλλες εφαρμογές. Το android προσφέρει στους κατασκευαστές λογισμικού ένα πλήρες πακέτο ανάπτυξης, που διαφοροποιεί το λογισμικό από το hardware. Αυτό οδηγεί σε εφαρμογές που υποστηρίζονται από μεγάλο αριθμό συσκευών διαφορετικών ειδών, προσφέροντας ένα πλούσιο οικοσύστημα για τους developers και τους καταναλωτές[28].

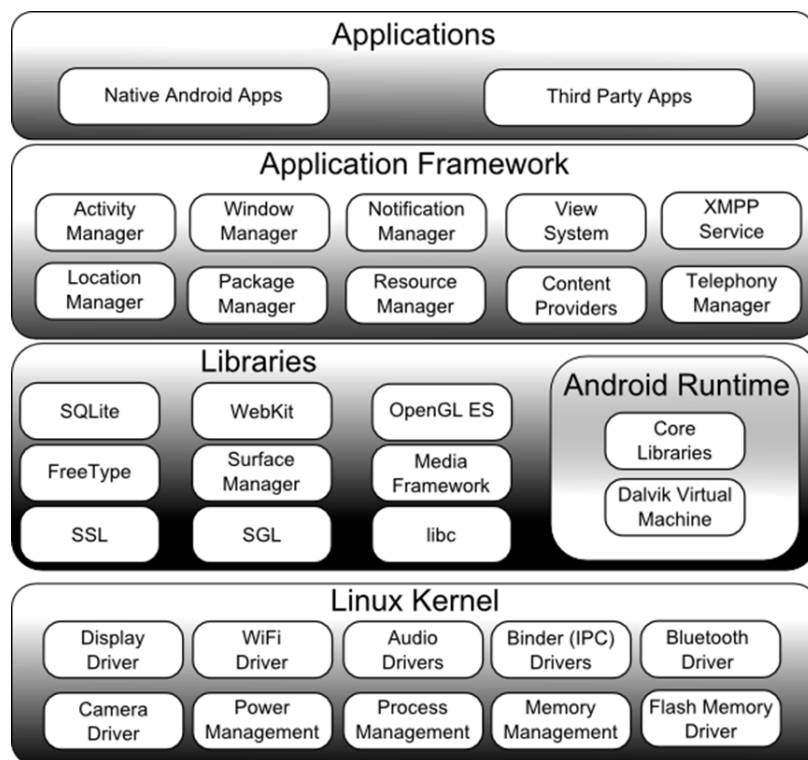
2.4.1 Αρχιτεκτονική Android

Το λειτουργικό σύστημα android αποτελείται από διαφορετικές συνιστώσες. Η αρχιτεκτονική του συστήματος απαρτίζεται από πέντε τομείς και τέσσερα επίπεδα. Τα επίπεδα αυτά είναι:

- Τον πυρήνα Linux (Linux kernel) : Το Android είναι βασισμένο στον πυρήνα του Linux Kernel[29]. Η χρήση ενός πυρήνα Linux επιτρέπει στο Android να εκμεταλλευτεί τα

βασικά χαρακτηριστικά ασφαλείας και επιτρέπει στους κατασκευαστές συσκευών να αναπτύξουν προγράμματα οδήγησης υλικού για έναν ήδη γνωστό πυρήνα. Λειτουργεί σαν επίπεδο διασύνδεσης μεταξύ του hardware και της στοίβας λογισμικού.

- Τις εγγενείς και τις προηγμένες βιβλιοθήκες (Libraries) και τον χρόνο εκτέλεσης (Android Runtime) : Οι βιβλιοθήκες του Android, περιλαμβάνουν ένα σύνολο από C/C++ βιβλιοθήκες, που χρησιμοποιούνται από διάφορα δομικά στοιχεία του συστήματος. Οι προγραμματιστές αποκτούν πρόσβαση σε αυτές μέσω του android application framework. Στην περιοχή του Runtime (ART) περιλαμβάνονται επίσης ένα σύνολο βασικών βιβλιοθηκών χρόνου εκτέλεσης που παρέχουν το μεγαλύτερο μέρος της λειτουργικότητας της γλώσσας προγραμματισμού Java[52].
- Το πλαίσιο εφαρμογής (Application Framework) : Το σύνολο των χαρακτηριστικών του Android OS είναι διαθέσιμο μέσω των java API. Μέσω αυτών, είναι δυνατό εφαρμογές να χρησιμοποιήσουν στοιχεία διαφορετικών εφαρμογών του συστήματος ή τρίτων. Την χρήση τέτοιων API κάνει και το σύστημα προβολής για την δημιουργία ενός περιβάλλοντος χρήστη, συμπεριλαμβανομένων λιστών, πλεγμάτων, πλαισίων κειμένου και κουμπιών[52].



Σχήμα 2-1. Αρχιτεκτονική android.

2.4.2 Δραστηριότητα Android

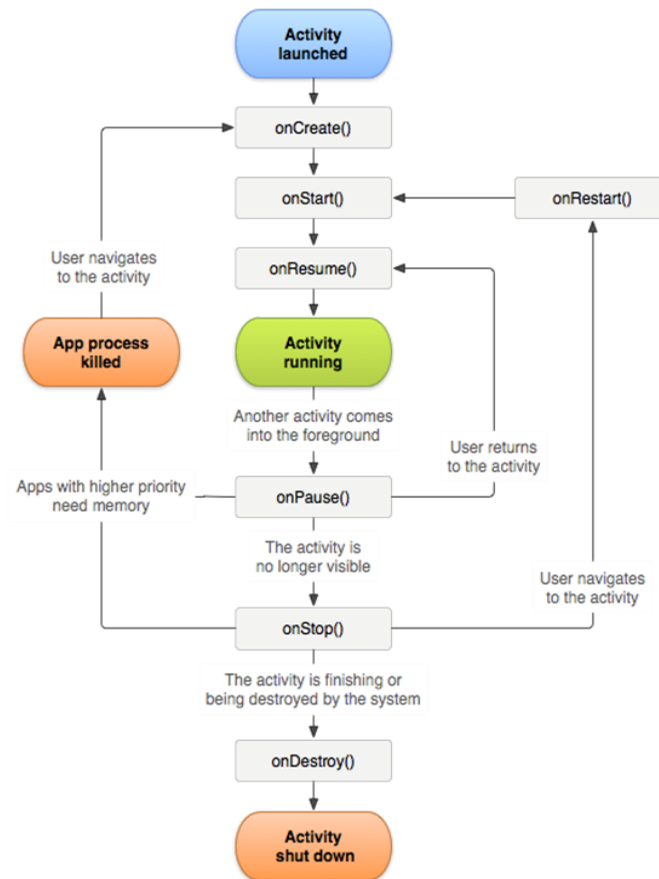
Η δραστηριότητα είναι το βασικό συστατικό μίας εφαρμογής, που παρέχει μια οθόνη αποτελούμενη από γραφικά και πιθανές ενέργειες, με την οποία οι χρήστες μπορούν να αλληλοεπιδρούν. Κάθε δραστηριότητα διαθέτει ένα προκαθορισμένο παράθυρο, που συνήθως γεμίζει την οθόνη, προς σχεδιασμό[30]. Μια εφαρμογή αποτελείται συνήθως από πολλαπλές δραστηριότητες που είναι συνδεδεμένες μεταξύ τους. Συνήθως, μια δραστηριότητα σε μια εφαρμογή έχει οριστεί ως η «κύρια» δραστηριότητα, η οποία παρουσιάζεται στο χρήστη κατά την εκκίνηση της εφαρμογής. Κάθε δραστηριότητα μπορεί στη συνέχεια να αρχίσει μια άλλη

δραστηριότητα, προκειμένου να εκτελέσει διάφορες ενέργειες. Κάθε φορά που μια νέα δραστηριότητα ξεκινά, η προηγούμενη δραστηριότητα έχει σταματήσει, αλλά το σύστημα διατηρεί τη δραστηριότητα σε μια στοίβα.

Όπως φαίνεται και στο Σχήμα 2-2 οι μέθοδοι που καλούνται κατά την διάρκεια ζωής μία δραστηριότητας είναι οι παρακάτω:

- Η μέθοδος `onCreate ()` καλείται όταν η δραστηριότητα δημιουργείται για πρώτη φορά. Αφού ολοκληρωθεί η εκτέλεση της μεθόδου `onCreate ()`, η δραστηριότητα εισέρχεται στην κατάσταση έναρξης και το σύστημα καλεί την μέθοδο `onStart ()`
- Όταν η δραστηριότητα δεν είναι υπάρχει στην μνήμη, τότε είναι σε κατάσταση εκκίνησης. Αμέσως επόμενη κατάσταση είναι η τρέχων. Η μετάβαση από την αρχική στην τρέχων κατάσταση είναι από τις πιο δαπανηρές χρονικά, γεγονός που επηρεάζει άμεσα την μπαταρία[28].
- Η δραστηριότητα που βρίσκεται στην τρέχων κατάσταση, είναι αυτή που την δεδομένη στιγμή είναι στο προσκήνιο και αλληλοεπιδρά με τον χρήστη. Όλες οι ενέργειες του χρήστη διεκπεραιώνονται από αυτή την δραστηριότητα, η οποία έχει προτεραιότητα στην μνήμη για πιο γρήγορες εκτελέσεις[28].
- Μία δραστηριότητα είναι σε κατάσταση παύσης, όταν είναι ορατή στον χρήστη, αλλά δεν αλληλοεπιδρά με αυτή. Αυτό μπορεί να συμβεί, όταν για παράδειγμα αναδύονται παράθυρα διαλόγων πάνω από μία δραστηριότητα. Ομοίως με την τρέχουσα, μία δραστηριότητα σε κατάσταση παύσης, έχει προτεραιότητα στην μνήμη και άλλους πόρους[28].
- Μία δραστηριότητα είναι σε κατάσταση αναστολής, όταν δεν είναι ορατή, αλλά βρίσκεται στην μνήμη. Εάν επιστρέψει στο προσκήνιο, έρχεται σε τρέχων κατάσταση, ή μπορεί να διαγραφεί από την μνήμη. Είναι πιο “οικονομικό” για το σύστημα από πλευράς πόρων να επαναφέρει μία δραστηριότητα από αναστολή σε τρέχων κατάσταση, απ’ ότι να δημιουργήσει μία δραστηριότητα από την αρχή[28].
- Στην κατάσταση καταστροφής η δραστηριότητα έχει πλέον διαγραφεί από την μνήμη[28].

Στο Σχήμα 2-2. Κύκλος ζωής δραστηριότητας[56]. Σχήμα 2-2 παρουσιάζονται οι καταστάσεις μίας δραστηριότητας, οι μεταβάσεις μεταξύ αυτών, καθώς και οι μέθοδοι του συστήματος που τις προκαλούν.



Σχήμα 2-2. Κύκλος ζωής δραστηριότητας[56].

2.4.3 Χρήσιμες βιβλιοθήκες android

Το λογισμικό android, και η ευρεία κοινότητα του, προσφέρει μία πληθώρα βιβλιοθηκών, για την υλοποίηση διαφόρων λειτουργιών. Κάποιες από αυτές, αφορούν την συνδεσιμότητα, την αποθήκευση δεδομένων και την γραφική αναπαράσταση και αναφέρονται παρακάτω.

- Networking Libraries- Volley vs Retrofit

Η Volley είναι μία HTTP βιβλιοθήκη τύπου REST, που αναπτύχθηκε από την Google και κατασκευάζει εργασίες δικτύωσης σε εφαρμογές Android εύκολα και γρήγορα. Η Volley δουλεύει ασύγχρονα[38]. Περιλαμβάνει αυτόματο προγραμματισμό αιτημάτων, πολλαπλές παράλληλες συνδέσεις, δυνατότητα ταξινόμησης αιτημάτων κατά προτεραιότητα, και ασύγχρονη μεταφορά δεδομένων από το δίκτυο στον client. Από τα σημαντικότερα πλεονεκτήματα της Volley έναντι άλλων βιβλιοθηκών, είναι τα εργαλεία εντοπισμού σφαλμάτων που διαθέτει.

Η Retrofit είναι επίσης μία HTTP βιβλιοθήκη τύπου REST, η οποία είναι ελαφριά και ευκολότερη στην υλοποίηση της. Από τα σημαντικότερα πλεονεκτήματα της Retrofit, είναι η ευκολία στην διαχείριση των JSON αντικειμένων[38].

Οι παραπάνω βιβλιοθήκες διασύνδεσης, είναι και οι δύο ευρέως διαδεδομένες και εφάμιλλες όσον αφορά την απόδοσή τους. Υπάρχουν πειράματα και αναφορές που ευνοούν από πλευράς απόδοσης την Volley [38], και άλλα που ευνοούν την Retrofit [39].

- **SQLite**

Η SQLite, είναι μία σχεσιακή βάση δεδομένων, η οποία μπορεί να στηθεί αυτόνομα και να ενσωματωθεί σε μία εφαρμογή. Είναι ένα RDBMS ελαφρύ στην εγκατάσταση και την χρήση πόρων[40]. Η SQLite, η οποία είναι ενσωματωμένη στο λειτουργικό Android, θα χρησιμοποιηθεί για να περιοριστούν τα αιτήματα προς τον webserver. Ο περιορισμός αυτός, επιτυγχάνεται αποθηκεύοντας τα δεδομένα του χρήστη που δεν αλλάζουν, όπως το username και το email του. Όταν ένας χρήστης συνδεθεί στην εφαρμογή, τα στοιχεία του αποθηκεύονται στην SQLite βάση δεδομένων του κινητού. Σε ένα session, ο χρήστης μπορεί να ζητήσει από τον Server διάφορα δεδομένα ή ενέργειες, όπως προσθήκη θερμοκηπίου, επισκόπηση τιμών συγκεκριμένου θερμοκηπίου, κλπ. Όλα αυτά τα αιτήματα αφορούν τον συνδεδεμένο χρήστη. Αν η υλοποίηση δεν γινόταν με χρήση της SQLite τότε για κάθε αίτημα του χρήστη θα έπρεπε να γίνει ένα επιπλέον αίτημα στον webserver για να ανακτηθούν τα δεδομένα του συνδεδεμένου χρήστη, γεγονός που θα φόρτωνε τον webserver με περιττά επαναλαμβανόμενες διεργασίες και θα χρησιμοποιούσε bandwidth από το δίκτυο.

- **Βιβλιοθήκη γραφικών παραστάσεων Anychart**

Για την ενημέρωση του χρήστη, για τις περιβαλλοντικές τιμές, επιλέχθηκε η βιβλιοθήκη ANyChart. Η βιβλιοθήκη αυτή, είναι ελαφριά στην χρήση πόρων, και προσφέρει πληθώρα επιλογών για κάθε μορφή plot. Επιπλέον, η βιβλιοθήκη αυτή είναι open-source, όταν πρόκειται για μη εμπορικά projects.

2.5 Εργαλεία ανάπτυξης

Παρακάτω παρουσιάζονται τα βασικά εργαλεία ανάπτυξης που χρησιμοποιήθηκαν για την υλοποίηση των υποσυστημάτων.

2.5.1 Android studio

Το Android Studio αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού (IDE) Android εφαρμογών μέσα στο οποίο μπορεί να αναπτυχθεί και να εκτελεστεί ο κώδικας με την χρήση του emulator. Θεωρείται το επίσημο περιβάλλον ανάπτυξης Android εφαρμογών καθώς αναπτύχθηκε και υποστηρίζεται από την Google [52].

2.5.2 WampServer

Το WampServer είναι ένα πρόγραμμα για λειτουργικό Windows το οποίο συνθέτει ένα περιβάλλον ανάπτυξης διαδικτυακών εφαρμογών. Το όνομα Wamp οφείλεται στα προγράμματα που περιλαμβάνει, ‘A’ για το **Apache**, ‘P’ για το **PHP** και ‘M’ για **MySQL**. Τέλος το ‘W’ αναφέρεται ότι το συγκεκριμένο πρόγραμμα τρέχει σε λογισμικό Windows. Αποτελεί ένα ολοκληρωμένο αναπτυξιακό πακέτο για prototypes διαδικτυακών εφαρμογών, πριν αυτές κλιμακωθούν ή βγουν στο εμπόριο.

2.5.3 Jmeter

Το Jmeter, είναι ένα load testing εργαλείο, για τον έλεγχο απόδοσης διαδικτυακών υπηρεσιών και APIs. Διαθέτει ευρεία γκάμα επιλογών, για αποστολή HTTP requests και προσομοίωση χρηστών μέσω threads, για την δημιουργία διαφορετικών load στο API.

2.5.4 Postman

Το εργαλείο postman, χρησιμοποιείται για την αποστολή HTTP request σε web services και APIs. Είναι πιο εύχρηστο από το Jmeter, αλλά δεν διαθέτει τον ίδιο αριθμό επιλογών. Το εργαλείο postman στέλνει ένα request την φορά, και είναι πιο εύχρηστο στον αρχικό έλεγχο επαλήθευσης των μεθόδων του API.

2.5.5 Wireshark

Το wireshark, είναι ένα εργαλείο ανάλυσης και παρακολούθησης δικτύου σε επίπεδο πακέτων. Εντοπίζει τα πακέτα που μεταφέρονται σε ένα δίκτυο, τον αποστολέα και τα στοιχεία των επικεφαλίδων τους, τις εσφαλμένες αποστολές, και το traffic του δικτύου.

3. ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ ΚΑΙ ΣΧΕΔΙΑΣΤΙΚΕΣ ΕΠΙΛΟΓΕΣ

Σε αυτήν την ενότητα, αναλύεται το σύστημα του θερμοκηπίου, οι αρχές λειτουργίας του, οι δυνατότητες συνδεσιμότητάς του, και οι ανάγκες του χρήστη. Κατόπιν, παρουσιάζονται οι τροποποιήσεις που έγιναν για να επιτευχθεί η σύνδεσή του και η ομαλή συνεργασία του με το σύστημα διαχείρισης, και παρουσιάζονται οι θεμελιώδεις σχεδιαστικές επιλογές που ακολουθήθηκαν με βάσει τα παραπάνω.

3.1 Επισκόπηση συστήματος

Το σύστημα διαχείρισης, που αφορά την παρούσα διπλωματική εργασία, συνάπτει την επικοινωνία του χρήστη με το θερμοκήπιο Leko.



Σχήμα 3-1. Σύστημα διαχείρισης θερμοκηπίου Top Level.

Το σύστημα διαχείρισης, λειτουργεί ως ενδιάμεσος σταθμός μεταξύ χρήστη και θερμοκηπίου, και συμπληρώνει την αλυσίδα της διαδικασίας αυτόματης καλλιέργειας.

Για την υλοποίηση του συστήματος διαχείρισης του θερμοκηπίου που αφορά την παρούσα διπλωματική, αναλύθηκε πρώτα μεμονωμένα το σύστημα του θερμοκηπίου.

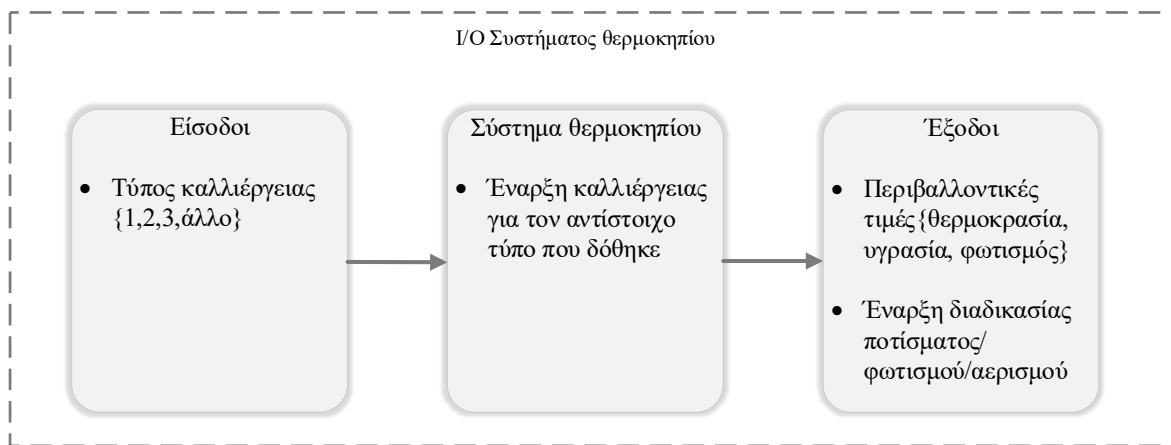
3.2 Ανάλυση συστήματος θερμοκηπίου

Το θερμοκήπιο εσωτερικού χώρου Leko, αποτελείται από έναν μικροελεγκτή, συνδεδεμένο με αισθητήρες θερμοκρασίας, υγρασίας και φωτισμού. Ο μικροελεγκτής, με βάση τις μετρήσεις που λαμβάνει από τους αισθητήρες, οδηγεί και τις αντίστοιχες εξόδους, π.χ. πότισμα εάν η υγρασία είναι χαμηλή. Το σύστημα αυτό έχει την δυνατότητα να λαμβάνει μετρήσεις από το περιβάλλον της καλλιέργειας, να εκτιμάει τις τιμές αυτές και να ενεργεί κατάλληλα.

3.2.1 Μοντελοποίηση συστήματος θερμοκηπίου

Για να τεθεί σε λειτουργία το θερμοκήπιο, αρχικά καταχωρείται ο τύπος της καλλιέργειας που θα αναπτυχθεί σε αυτό. Έπειτα ο χρήστης να παρακολουθεί τις περιβαλλοντικές συνθήκες του, μέσω του δικτύου LAN. Για την υλοποίηση της λειτουργίας monitoring, που θα αναλυθεί παρακάτω στην παρούσα ενότητα, το θερμοκήπιο πρέπει να ενημερώνει και για την

δραστηριότητα των ενεργοποιητών του. Το μοντέλο του θερμοκηπίου, οι είσοδοι και η έξοδοί του, όπως προέκυψαν μετά την παραπάνω τροποποίηση, φαίνονται στο Σχήμα 3-2.



Σχήμα 3-2. Είσοδοι – έξοδοι συστήματος θερμοκηπίου.

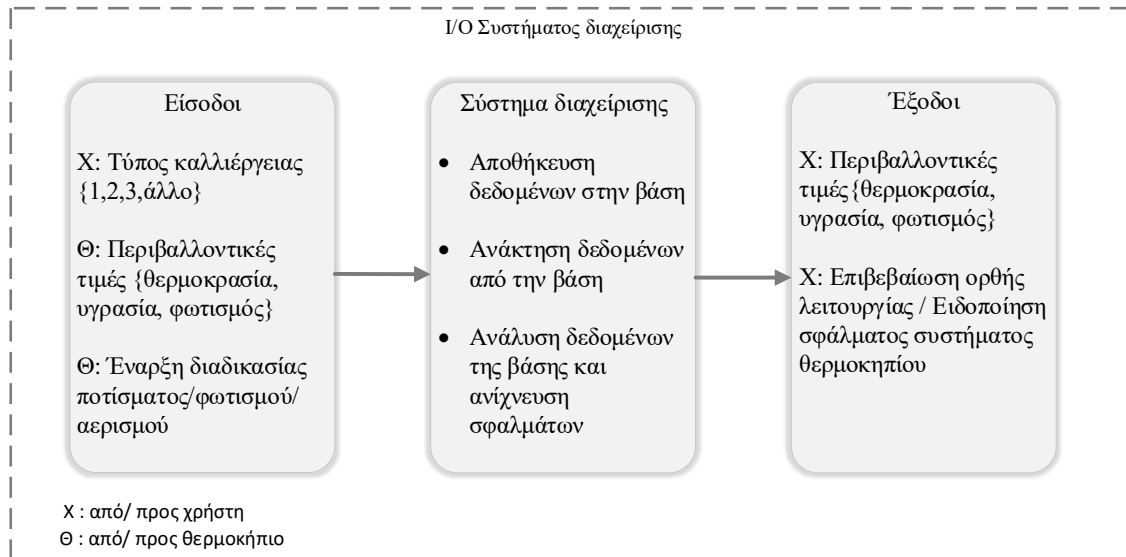
3.3 Ανάλυση συστήματος διαχείρισης

Η σχεδίαση του συστήματος διαχείρισης, εξαρτάται από το μοντέλο των υπόλοιπων συναλλασσόμενων συστημάτων. Έπειτα από την ανάλυση και μοντελοποίηση του θερμοκηπίου, ακολουθεί η μοντελοποίηση του συστήματος διαχείρισης και οι σχεδιαστικές επιλογές που έγιναν.

3.3.1 Μοντελοποίηση συστήματος διαχείρισης

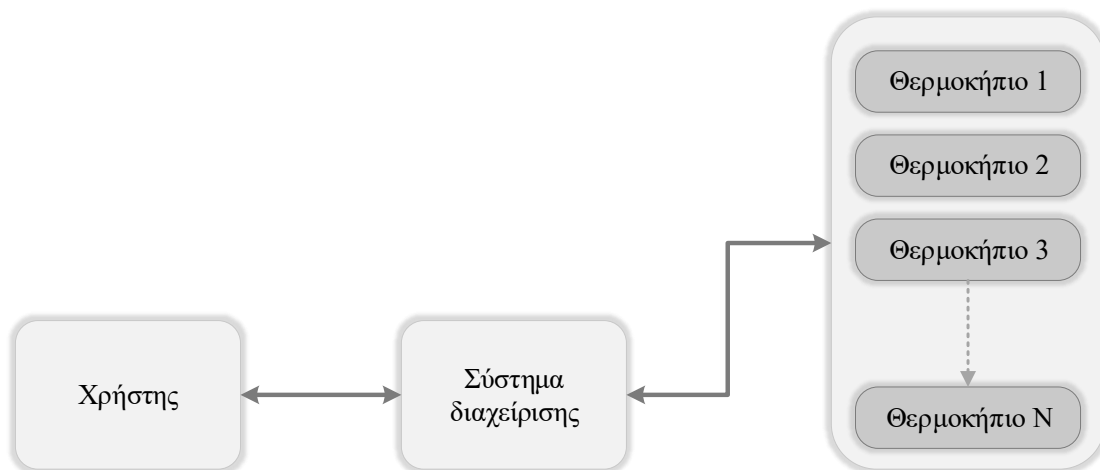
Με βάση τις εισόδους του συστήματος του θερμοκηπίου, προκύπτουν και οι δυνατότητες αλληλοεπίδρασης του χρήστη με αυτό. Ο χρήστης, ορίζει τον τύπο της καλλιέργειας κατά την εισαγωγή. Κατόπιν, παρακολουθεί τις περιβαλλοντικές τιμές της, επιβλέποντας με αυτόν τον τρόπο την διαδικασία ανάπτυξής της. Το σύστημα διαχείρισης του θερμοκηπίου, αλληλοεπιδρά ταυτόχρονα και με τον χρήστη και με το θερμοκήπιο, γεφυρώνοντας τα δύο αυτά υποσυστήματα.

Για την εξασφάλιση την ομαλής λειτουργίας του θερμοκηπίου, το υλοποιηθέν σύστημα ανιχνεύει πιθανά σφάλματα λειτουργίας και ενημερώνει τον χρήστη για αυτά. Με βάση τις παραπάνω προδιαγραφές, τις ιδιότητες του χρήστη και του θερμοκηπίου, οι είσοδοι και οι έξοδοι του συστήματος διαχείρισης φαίνονται στο Σχήμα 3-3.



Σχήμα 3-3. Είσοδοι – έξοδοι συστήματος διαχείρισης

Επιπλέον, το σύστημα διαχείρισης επιτρέπει στον χρήστη να εγκαταστήσει όσα θερμοκήπια επιθυμεί, χωρίς να επηρεάζεται η απόδοση του.



Σχήμα 3-4. Διαχείριση πολλαπλών θερμοκηπίων.

3.3.2 Απαιτήσεις χρήστη

Οι απαιτήσεις του χρήστη που αφορούν την διαχείριση της καλλιέργειας του, συνοψίζονται στα παρακάτω:

- Απομακρυσμένη παρακολούθηση
- Υποστήριξη παραπάνω του ενός θερμοκηπίου
- Παρακολούθηση από πολλές συσκευές
- Αξιοπιστία
- Ευκολία στην χρήση
- Αυτόματη ανίχνευση σφαλμάτων και ειδοποίηση

3.3.3 Υποσυστήματα συστήματος διαχείρισης

Το σύστημα διαχείρισης αποτελείται από δύο υποσυστήματα. Το ένα είναι η διεπαφή του χρήστη που αποτελεί το frontend, και το άλλο ο web server που αποτελεί το backend του συστήματος.



Σχήμα 3-5. Front και Back end Συστήματος διαχείρισης.

3.4 Σχεδιαστικές επιλογές

Σε αυτήν την υποενότητα, αναλύονται οι σχεδιαστικές επιλογές που ακολουθήθηκαν για την υλοποίηση του συστήματος διαχείρισης. Αυτές περιλαμβάνουν την διεπαφή χρήστη, την διεπαφή θερμοκηπίου, καθώς και διαδικασίες και τροποποιήσεις που έγιναν στο σύστημα θερμοκηπίου, για να εναρμονιστεί με το σύστημα διαχείρισης.

3.4.1 Διεπαφή χρήστη

Το front end, ορίζει την αλληλεπίδραση του χρήστη με το υπόλοιπο σύστημα. Χαρακτηρίζεται από την λειτουργία αποστολής και λήψης αιτημάτων προς και από το back end που το χαρακτηρίζει ως client, και από το γραφικό του περιβάλλον (GUI).

Συνοπτικά οι λειτουργίες του, όπως παρουσιάζονται και στην μοντελοποίηση του συστήματος, στην υποενότητα 3.3.1, είναι οι παρακάτω:

- Ορισμός τύπου καλλιέργειας (ζαρζαβαρικά, παχύφυτα, καλλωπιστικά)
- Προβολή περιβαλλοντικών τιμών και ιστορικού καλλιέργειας (θερμοκρασία, υγρασία εδάφους, φωτισμός)
- Λήψη ειδοποιήσεων εσφαλμένης λειτουργίας

Δύο πιθανές λύσεις, για την διεπαφή χρήστη, είναι η δημιουργία ιστοσελίδας, ή εφαρμογής έξυπνου κινητού τηλεφώνου. Οι δύο αυτές επιλογές αναλύονται παρακάτω.

- Ιστοσελίδα

Η ιστοσελίδα που αποτελείται κυρίως από την γλώσσα HTML, μπορεί κάλλιστα να λειτουργήσει ως GUI για το σύστημά, καθώς πληροί πολλές από τις απαιτήσεις του. Για την πρόσβαση σε αυτήν απαιτείται απλά ένας περιηγητής ιστού που υπάρχει σε όλους τους ηλεκτρονικούς υπολογιστές, έξυπνα κινητά και tablet.

- Εφαρμογή έξυπνης συσκευής

Η δημιουργία εφαρμογής για front end, μπορεί να λειτουργήσει σε έξυπνες συσκευές, δηλαδή σε έξυπνα κινητά ή tablet. Η εφαρμογή προσφέρει μεγάλη ευκολία στον χρήστη καθώς είναι άμεσα προσπελάσιμη από την έξυπνη συσκευή του οποιαδήποτε στιγμή, και επίσης προσφέρει ένα πιο πλούσιο γραφικό περιβάλλον, που ενισχύεται από τις επιπλέον λειτουργίες που προσφέρει η συσκευή και βελτιώνουν αισθητά την εμπειρία του χρήστη.

- Επιλογή λειτουργικού εφαρμογής

Τα δύο δημοφιλέστερα λειτουργικά συστήματα είναι το Android (της Google) και το iOS (της Apple). Επιλέχθηκε το android, λόγω του ότι είναι λογισμικό ανοικτού

κώδικα και έχει μεγαλύτερη διαθεσιμότητα. Οι συσκευές android αποτελούν περίπου το 72% της αγοράς έναντι περίπου 27% το ποσοστό των iOS[54].

Στον Πίνακα 3-1. Σύγκριση Ιστοσελίδας-Εφαρμογής Android., παρουσιάζονται τα βασικότερα χαρακτηριστικά των παραπάνω σχεδιαστικών επιλογών.

Πίνακας 3-1. Σύγκριση Ιστοσελίδας-Εφαρμογής Android.		
Χαρακτηριστικά	Ιστοσελίδα	Εφαρμογή android
Ευκολία χρήσης	Εύκολο	Εύκολο
Προσβασιμότητα	Από οποιαδήποτε συσκευή διαθέτει web browser όπως υπολογιστή, smartphone και tablet	Μόνο από smartphone και tablet που διαθέτουν λειτουργικό android
Εμπειρία χρήστη	Περιορισμένη	Πλούσια. Δυνατότητα χρήσης λειτουργιών της έξυπνης συσκευής, όπως GPS, επιταχυνσιόμετρο και άλλα.
Αμεσότητα	Ο χρήστης πρέπει να βρίσκεται σε υπολογιστή	Ο χρήστης μπορεί να κάνει χρήση του GUI όποτε θέλει αν έχει την εφαρμογή στο smartphone του.
Δυνατότητα ειδοποιήσεων	Δεν υπάρχει	Η εφαρμογή μπορεί να στέλνει ειδοποιήσεις στον χρήστη για κάποιο γεγονός που συμβαίνει σε πραγματικό χρόνο, ακόμα και όταν ο χρήστης δεν χρησιμοποιεί την εφαρμογή

Η επιλογή της εφαρμογής android, για την διεπαφή χρήστη, έγινε γιατί η εφαρμογή είναι πιο άμεσα προσβάσιμη, καθώς ο χρήστης μπορεί να αλληλοεπιδράσει με το σύστημα σχεδόν από παντού μέσω WiFi ή 3G/4G. Επίσης, το σημαντικότερο πλεονέκτημα της εφαρμογής, από πλευρά λειτουργικότητας, είναι η δυνατότητα λήψης ειδοποιήσεων. Ο χρήστης μπορεί να ενημερώνεται για την καλλιέργειά του σε πραγματικό χρόνο, όπου και αν είναι, χωρίς να απαιτείται να χρησιμοποιεί την εφαρμογή την συγκεκριμένη στιγμή.

3.4.2 Διεπαφή θερμοκηπίου

Η τοπολογία του IoT δικτύου που αφορά το παρόν σύστημα, ορίζεται ως τοπολογία αστέρα (βλ. ενότητα 2.1.1). Ο κεντρικός κόμβος της τοπολογίας, είναι ο κεντρικός server του συστήματος διαχείρισης, και τα θερμοκήπια των χρηστών συνδέονται σε αυτόν.

Η σύνδεση των θερμοκηπίων Leko, στον κεντρικό server, μπορεί να πραγματοποιηθεί με διαφορετικούς τρόπους, αξιοποιώντας το δεδομένο hardware που φέρει. Το θερμοκήπιο Leko, μέσω της κάρτας ESP8266-01, διαθέτει την στοίβα πρωτοκόλλων TCP/IP. Όπως αναφέρθηκε και στην υποενότητα 1.1.4, η στοίβα TCP/IP, αποτελείται από τέσσερα επίπεδα. Το επίπεδο που απασχολεί την διεπαφή του θερμοκηπίου με το σύστημα διαχείρισης, είναι το επίπεδο εφαρμογής (application layer). Τα βασικότερα IoT πρωτόκολλα σε αυτό το επίπεδο, είναι το

HTTP, το MQTT και το CoAP. Στον **Πίνακα 3-2**, παρουσιάζονται τα ποιοτικά χαρακτηριστικά τους.

Πίνακας 3-2. Χαρακτηριστικά των πρωτοκόλλων IoT, HTTP, MQTT, CoAP.			
Χαρακτηριστικά	HTTP	MQTT	CoAP
Transfer Protocol που χρησιμοποιεί	TCP	TCP	UDP
Τύπος πρωτοκόλλου	Web	M2M	M2M
Συμβατότητα με API	Restful API	Broker API	Restful/Broker API
Ανιχνευσιμότητα συσκευής	Μη εφικτή -εφικτή με χρήση uPnP	Εφικτή	Εφικτή
Latency	Πιο υψηλό	Χαμηλό	Χαμηλό
Αξιοπιστία	Πολύ αξιόπιστο	Πολύ αξιόπιστο	Λιγότερο αξιόπιστο

Η επικοινωνία σχεδιάστηκε με το πρωτόκολλο HTTP. Η επιλογή του πρωτοκόλλου HTTP, έγινε γιατί είναι αξιόπιστο, και μπορεί να υποστηρίξει τις λειτουργίες του συστήματος που απαιτούν web διεργασίες στο backend (λειτουργία monitoring).

Το MQTT είναι M2M πρωτόκολλο, και δεν υποστηρίζει εύκολα ένα backend με σύνθετες λειτουργίες. Το CoAP, που όπως αναφέρθηκε και στην υποενότητα 2.1.3, βασίζεται στο HTTP πρωτόκολλο, υποστηρίζει την επικοινωνία με Restful APIs όπως αυτό του συστήματος. Είναι όμως, ένα πρωτόκολλο για συσκευές με περιορισμένους πόρους και ασθενή δίκτυα, γεγονός που το κάνει να υστερεί στην αξιοπιστία σε σχέση με το HTTP. Η χαμηλότερη αξιοπιστία του, οφείλεται στην χρήση UDP στο transfer layer, που διακρίνεται για την απόκρισή του και όχι για την άψογη επικοινωνία.

Το HTTP, βέβαια, έχει κάποια μειονεκτήματα σε σχέση με τα παραπάνω πρωτόκολλα. Το πρώτο, είναι τα υψηλότερα latencies και το μεγαλύτερο bandwidth που καταλαμβάνει. Στην εφαρμογή του οικοσυστήματος Leko, τα παραπάνω μειονεκτήματα δεν αποτελούν πρόβλημα, καθώς τα δίκτυα που βρίσκονται τα θερμοκήπια είναι οικιακά. Επιπλέον η επικοινωνία και η μετάδοση της πληροφορίας δεν είναι real-time, για να επηρεάζει το υψηλότερο latency.

Το δεύτερο μειονέκτημα, είναι ότι το HTTP, εξυπηρετεί επικοινωνίες που ακολουθούν το web μοντέλο, στο οποίο ο κάθε κόμβος έχει τον ρόλο του client ή του server, με τους ρόλους αυτούς να είναι προκαθορισμένοι. Αυτό σημαίνει, ότι το θερμοκήπιο, θα πρέπει να έχει τον ρόλο του client, που στέλνει δεδομένα στον server, αλλά δεν δέχεται αιτήματα από αυτόν. Αυτός είναι και ο κυριότερος λόγος που αναπτύχθηκαν πρωτόκολλα όπως τα παραπάνω, και αφορούν αποκλειστικά IoT δίκτυα, στα οποία οι συσκευές είναι ανιχνεύσιμες, δέχονται και στέλνουν αιτήματα σε άλλες συσκευές.

Η έλλειψη ανιχνευσιμότητας από το HTTP, μπορεί να λυθεί με την χρήση του πρωτοκόλλου uPnP, που προσφέρει αυτόματο port forwarding στο router και κοινοποίηση της IP της συσκευής σε άλλες στο δίκτυο. Αυτή η πρακτική είναι σύνηθες σε IoT δίκτυα που βασίζονται σε HTTP

πρωτόκολλο. Η υλοποίηση του uPnP μηχανισμού στο παρόν σύστημα, δεν είναι απαραίτητη, γιατί η αρχιτεκτονική του συστήματος διαχείρισης, δεν απαιτεί την αποστολή αιτημάτων προς το θερμοκήπιο. Η υλοποίηση του θα ήταν απαραίτητη, εάν ο χρήστης μπορούσε να στέλνει αυθαίρετα αιτήματα στο θερμοκήπιο, όπως για παράδειγμα αίτημα ποτίσματος εκτός του προγραμματισμένου.

Συνεπώς, η χρήση του uPnP, δημιουργεί απλά ένα επιπλέον κενό ασφαλείας. Πολλές συντονισμένες επιθέσεις με malwares σε IoT δίκτυα, όπως αυτές που αναφέρθηκαν στην υποενότητα 2.1.4, έγιναν σε συσκευές βασισμένες σε uPnP. Η σχεδίαση των διαδικασιών του συστήματος, δεν απαιτούν την ανιχνευσιμότητα του θερμοκηπίου στο δίκτυο.

- Διευθυνσιοδότηση θερμοκηπίου

Η IP του θερμοκηπίου, ορίζεται μέσω του DHCP πρωτοκόλλου, που είναι εγκατεστημένο σε όλα τα οικιακά router. Η δυναμική ανάθεση IP, σημαίνει ότι σε περίπτωση επανασύνδεσης του θερμοκηπίου, η IP του μπορεί να αλλάξει. Αυτό δεν αποτελεί πρόβλημα, εφόσον στο θερμοκήπιο έχει εισαχθεί το μοναδικό του ID και η διεύθυνση url του server. Επιπλέον, η διαδικασία της αυτόματης ανάθεσης, εξασφαλίζει την ευκολία σύνδεσης του από τον χρήστη.

3.4.3 Διαδικασία σύνδεσης του θερμοκηπίου με το σύστημα

Για την σύνδεση του συστήματος διαχείρισης με το σύστημα του θερμοκηπίου, δημιουργείται στον Web Server, ένας αλφαριθμητικός κωδικός id, που αποτελεί την υπογραφή του κάθε θερμοκηπίου. Ο κάθε κωδικός είναι μοναδικός για κάθε θερμοκήπιο και εμφανίζεται στον χρήστη στην οθόνη της android εφαρμογής. Περισσότερα για τα μοναδικά ID του συστήματος στην ενότητα 5.1.1.

Το διακριτικό id αυτό, εισάγεται κατόπιν από τον χρήστη στο τερματικό του θερμοκηπίου. Με την IP του server, ή το url του και την αξιοποίηση της DNS υπηρεσίας, και τον κωδικό ID, το θερμοκήπιο συνδέεται και μπορεί να στέλνει τα δεδομένα του. Η διαδικασία σύνδεσης φαίνεται στην Εικόνα 3-1.

Διαδικασία Σύνδεσης Θερμοκηπίου Leko	
Βήμα 1	Προσθήκη νέου θερμοκηπίου μέσω της εφαρμογής
Βήμα 2	Δημιουργία μοναδικού Id
Βήμα 3	Εισαγωγή μοναδικού Id στο τερματικό του θερμοκηπίου
Βήμα 4	Σύνδεση θερμοκηπίου με τον Server

Εικόνα 3-1. Διαδικασία σύνδεσης θερμοκηπίου Leko με τον webserver.

Αφότου συνδεθεί στον server, το θερμοκήπιο επικοινωνεί σύμφωνα με τις διαδικασίες που έχουν σχεδιαστεί, και αναφέρονται στην υποενότητα που ακολουθεί.

3.5 Σχεδίαση διαδικασιών συστήματος

Για την ολοκληρωμένη ενημέρωση του χρήστη, και τον έλεγχο ορθής λειτουργίας του θερμοκηπίου, τα δεδομένα που λαμβάνονται, είναι οι περιβαλλοντικές τιμές και η δραστηριότητα των ενεργοποιητών

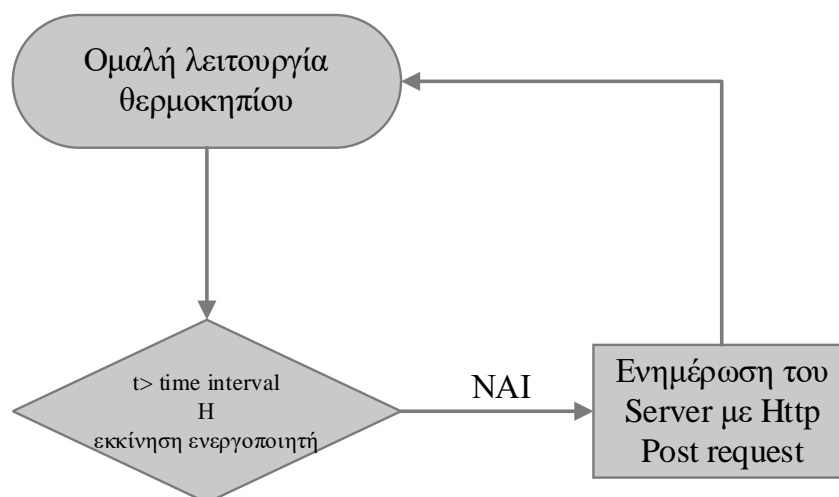
Για την απλοποίηση της επικοινωνίας, τον αξιόπιστο έλεγχο, και την ευκολία στην διαχείριση των δεδομένων στο backend, το θερμοκήπιο στέλνει τα δεδομένα αυτά σε ένα session επικοινωνίας και όχι ξεχωριστά. Κάθε φορά που γίνεται χρήση κάποιου ενεργοποιητή, το θερμοκήπιο αποστέλλει όλο το πακέτο πληροφορίας, που περιλαμβάνει την δραστηριότητα του ενεργοποιητή και τις περιβαλλοντικές τιμές την συγκεκριμένη χρονική στιγμή. Ένα πακέτο πληροφορίας από το θερμοκήπιο, έχει την μορφή:

Πίνακας 3-3. Δομή δεδομένων ενημέρωσης θερμοκηπίου					
Θερμοκρασία	Υγρασία	Φωτισμός	Ανεμιστήρας	Αντλία	Led
Τιμή	Τιμή	Χρόνος	Boolean	Boolean	Boolean

Η διαδικασία αυτή περιγράφεται στο Σχήμα 3-6, μέσω του διαγράμματος ενεργειών.

- Διαδικασία αποστολής δεδομένων θερμοκηπίου

Το θερμοκήπιο, ανά κάποιο συγκεκριμένο χρονικό διάστημα (time interval), αποστέλλει την κατάσταση του στον webserver. Η αποστολή των δεδομένων του θερμοκηπίου, γίνεται και στην περίπτωση που ενεργοποιηθεί κάποια από τις διαδικασίες (πότισμα, αερισμός, φωτισμός). Ο χρόνος ενημέρωσης, που ενεργοποιεί την αποστολή των δεδομένων, μετρίεται και ανανεώνεται εσωτερικά σε κάθε θερμοκήπιο.



Σχήμα 3-6. Διάγραμμα ενεργειών επικοινωνίας θερμοκηπίου και web server.

Για την υλοποίηση της παραπάνω διαδικασίας, απαιτείται τροποποίηση στο software του θερμοκηπίου. Η τροποποίηση αυτή, περιλαμβάνει την ενημέρωση σε σταθερό time interval, και την αποστολή με την ενεργοποίηση κάποιου actuator.

Με τα δεδομένα που συλλέγει το σύστημα από το κάθε θερμοκήπιο, ελέγχεται η ορθή λειτουργία του. Η ανάλυση της ορθής λειτουργίας του θερμοκηπίου και του μηχανισμού monitoring, παρουσιάζεται παρακάτω.

- Monitoring Συστήματος

Η μονάδα αυτή είναι υπεύθυνη για την ανίχνευση σφαλμάτων στο σύστημα του θερμοκηπίου, και την ειδοποίηση του χρήστη. Για να γίνει δυνατή η ανίχνευση σφαλμάτων πρέπει πρώτα να αναλυθούν τα πιθανά σφάλματα που μπορούν να προκύψουν στο σύστημα του θερμοκηπίου. Τα πιθανά σφάλματα διακρίνονται στις παρακάτω κατηγορίες:

- Σφάλμα δικτύωσης
- Αστοχία μικροελεγκτή
- Αστοχία αισθητήρων
- Αστοχία ενεργοποιητών
- Κενή στάθμη νερού

Οι παραπάνω κατηγορίες, εντοπίζονται με δύο βασικούς μηχανισμούς monitoring.

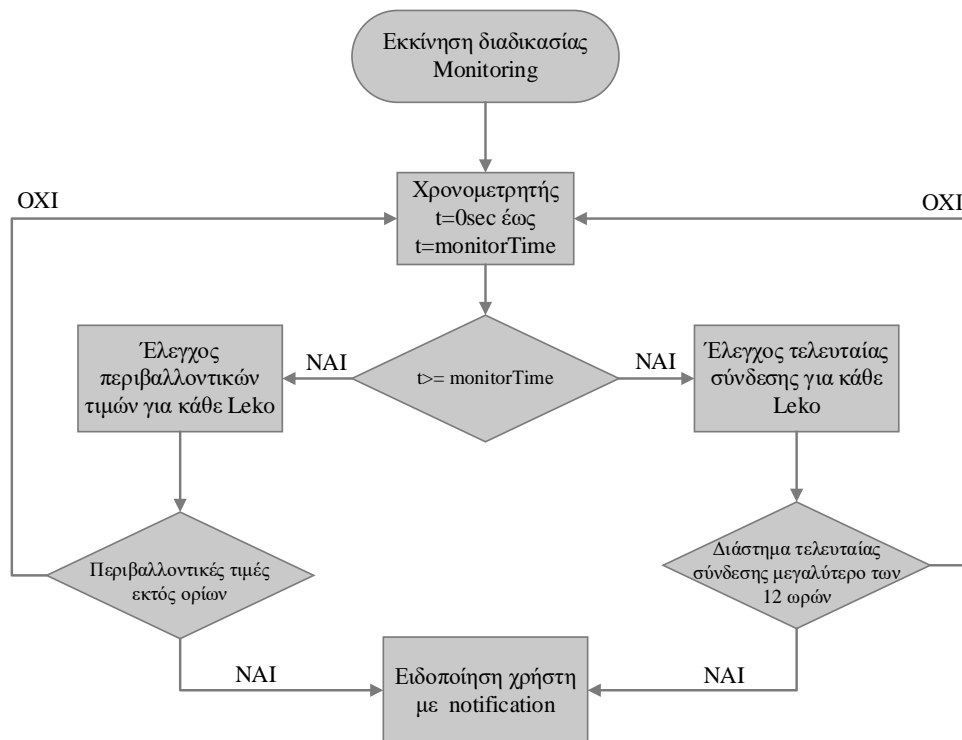
- Έλεγχος ορίων περιβαλλοντικών τιμών

Η ομαλή λειτουργία του θερμοκηπίου, εξασφαλίζει την διατήρηση των περιβαλλοντικών τιμών, στα επιτρεπτά εύρη, για κάθε τύπο καλλιέργειας. Συνεπώς, οποιαδήποτε δυσλειτουργία ή αστοχία προκύψει στο σύστημα του θερμοκηπίου, μπορεί να ανιχνευθεί παρατηρώντας τις τιμές αυτές. Το γεγονός αυτό, επιτρέπει την ικανοποίηση της monitoring λειτουργίας, με έναν μόνο μηχανισμό, αντί για τέσσερις. Ο μηχανισμός αυτός θα ελέγχει τις περιβαλλοντικές τιμές, και ειδοποιεί τον χρήστη σε περίπτωση απόκλισης.

- Σφάλμα δικτύωσης

Το σφάλμα δικτύωσης, παρουσιάζεται όταν υπάρχει πρόβλημα με την σύνδεση του θερμοκηπίου στο διαδίκτυο. Το θερμοκήπιο, σε αυτήν την περίπτωση, δεν θα μπορεί να ενημερώνει τον webserver για την κατάστασή του. Ο μηχανισμός ανίχνευσης του σφάλματος δικτύωσης, ελέγχει την ημερομηνία και ώρα των ανανεώσεων των θερμοκηπίων, και ενημερώνει τον χρήστη, εάν το διάστημα της τελευταίας ενημέρωσης είναι μεγαλύτερο του time interval αποστολής κατάστασης.

Το σύστημα monitoring περιγράφεται στο Σχήμα 3-7.



Σχήμα 3-7. Διάγραμμα ροής συστήματος monitoring.

3.6 Λειτουργικές απαιτήσεις και θέματα αξιοπιστίας συστήματος

Για να μπορεί ο χρήστης να παρακολουθεί το θερμοκήπιο του μέσω του συστήματος διαχείρισης, απαιτείται συσκευή Android με σύνδεση στο Internet, καθώς και θερμοκήπιο Leko συνδεδεμένο στο internet. Σε διαφορετική περίπτωση ο χρήστης δεν μπορεί να παρακολουθεί την καλλιέργειά του ούτε να ειδοποιείται για κάποια δυσλειτουργία στο θερμοκήπιο.

Ένα επιπλέον θέμα αξιοπιστίας είναι η ενδεχόμενη υπερφόρτωση του webserver. Η χωρητικότητα και η ισχύς επεξεργασίας του, θέτουν περιορισμούς στον αριθμό χρηστών και θερμοκηπίων που μπορούν να εξυπηρετηθούν. Για την εξασφάλιση της ομαλής λειτουργίας του Server, το σύστημα είναι επεκτάσιμο, καθώς μπορούν να προστεθούν επιπλέον servers εξυπηρέτησης, μέσω τρίτων παρόχων. Η αξιολόγηση των δυνατοτήτων του συστήματος, γίνεται στην ενότητα 7.

Ο χρήστης πρέπει να διαθέτει έξυπνη συσκευή λογισμικού android, με έκδοσης τουλάχιστον 4.0.x και άνω. Η συσκευή αυτή μπορεί να είναι, έξυπνο κινητό τηλέφωνο, tablet, ή οτιδήποτε άλλο «τρέχει» λειτουργικό android 4.0.x και άνω, ακόμα και smart TV. Για να μπορεί ο χρήστης να αλληλεπιδράει με το θερμοκήπιο του, θα πρέπει να έχει πρόσβαση στο Internet, μέσω WiFi ή μέσω δεδομένων (3G/4G).

Οι λειτουργικές απαιτήσεις του συστήματος συνοπτικά είναι οι παρακάτω:

- Έξυπνη συσκευή με λειτουργικό android 4.0.x και άνω
- Πρόσβαση στο διαδίκτυο (μέσω WiFi ή δεδομένων)
- Θερμοκήπιο εσωτερικού χώρου LEKO συνδεδεμένο στον Server

4. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

Στο κεφάλαιο αυτό, αναλύεται σε βάθος, πως τα λειτουργικά απαιτούμενα του backend του συστήματος, μεταφράστηκαν σε μία αρχιτεκτονική με τα βασικά υποσυστήματά της, και τις διεπαφές της με τα υπόλοιπα στοιχεία του συστήματος.

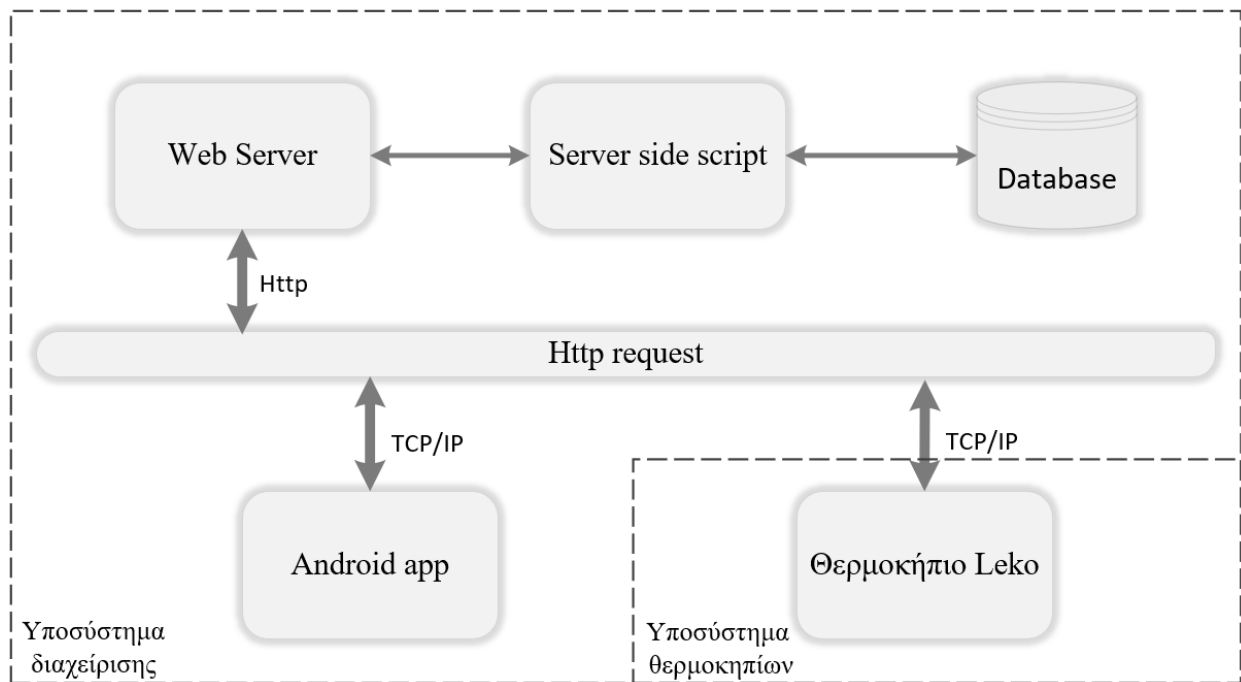
4.1 Αρχιτεκτονική backend συστήματος

Το backend του συστήματος, έχει την δομή, που φαίνεται στο Σχήμα 4-1. Ο server λειτουργεί ως δρομολογητής, για την εκτέλεση του κώδικα. Ο κώδικας, κατά την εκτέλεσή του αλληλοεπιδρά με την βάση δεδομένων. Έπειτα, επιστρέφει στον δρομολογητή, για να δρομολογηθεί η απάντηση του αιτήματος.



Σχήμα 4-1. Backend Diagram.

Η δομή του συνολικού συστήματος, που περιγράφει την διασύνδεση όλων των συστημάτων φαίνεται στο Σχήμα 4-2. Το backend, επικοινωνεί μέσω του δρομολογητή που ορίζει και το gateway, με το σύστημα του θερμοκηπίου και τον χρήστη. Ο χρήστης αλληλοεπιδρά με το backend μέσω της εφαρμογής Android, που αποτελεί το frontend, όπως αναφέρθηκε και στην προηγούμενη ενότητα.



Σχήμα 4-2. Top Level μοντέλο συστήματος.

Το παραπάνω μοντέλο χαρακτηρίζεται από την φορητότητα που προσφέρει με το χειρισμό από πολλαπλές συσκευές android, αλλά και δυνατότητα επέκτασης μελλοντικά από υπολογιστή μέσω ιστοσελίδας. Η αρχιτεκτονική του συστήματος, επιτρέπει την εξυπηρέτηση πολλαπλών χρηστών, παρέχοντας παράλληλα την δυνατότητα σε κάθε χρήστη να χειριστεί παραπάνω από ένα θερμοκήπιο.

Ένα ακόμα προτέρημα αυτής της προσέγγισης είναι ο όγκος των δεδομένων που συλλέγονται στον κεντρικό server. Οι πληροφορίες που εξάγονται από τα δεδομένα αυτά, είναι χρήσιμες στον χρήστη, που θέλει να παρακολουθεί το ιστορικό της καλλιέργειάς του, αλλά και στον μηχανικό που με την ανάλυσή τους μπορεί να βελτιώσει το σύστημα μελλοντικά.

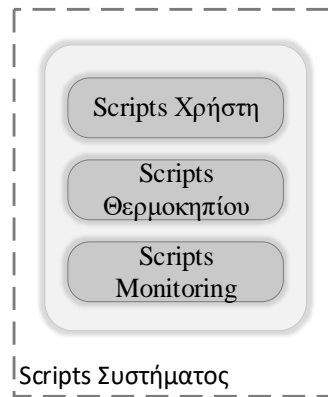
Συνοπτικά τα πλεονεκτήματα του συστήματος με Web Server είναι τα παρακάτω:

- Διαχείριση από πολλές συσκευές
- Εισαγωγή πολλαπλών θερμοκηπίων
- Απομακρυσμένη σύνδεση
- Ανίχνευση σφαλμάτων λειτουργίας θερμοκηπίου

4.1.1 Αρχιτεκτονική API

Το API του backend, αποτελείται από τον εκτελέσιμο κώδικα ή αλλιώς server side script. Το API, διεκπεραιώνει τα αιτήματα που λαμβάνει. Τα αιτήματα αυτά, προέρχονται είτε από την Android εφαρμογή, είτε από το θερμοκήπιο. Επιπλέον, το σύστημα διαχείρισης, μέσω των scripts, διεκπεραιώνει και την επίβλεψη ορθής λειτουργίας monitoring του κάθε συνδεδεμένου θερμοκηπίου.

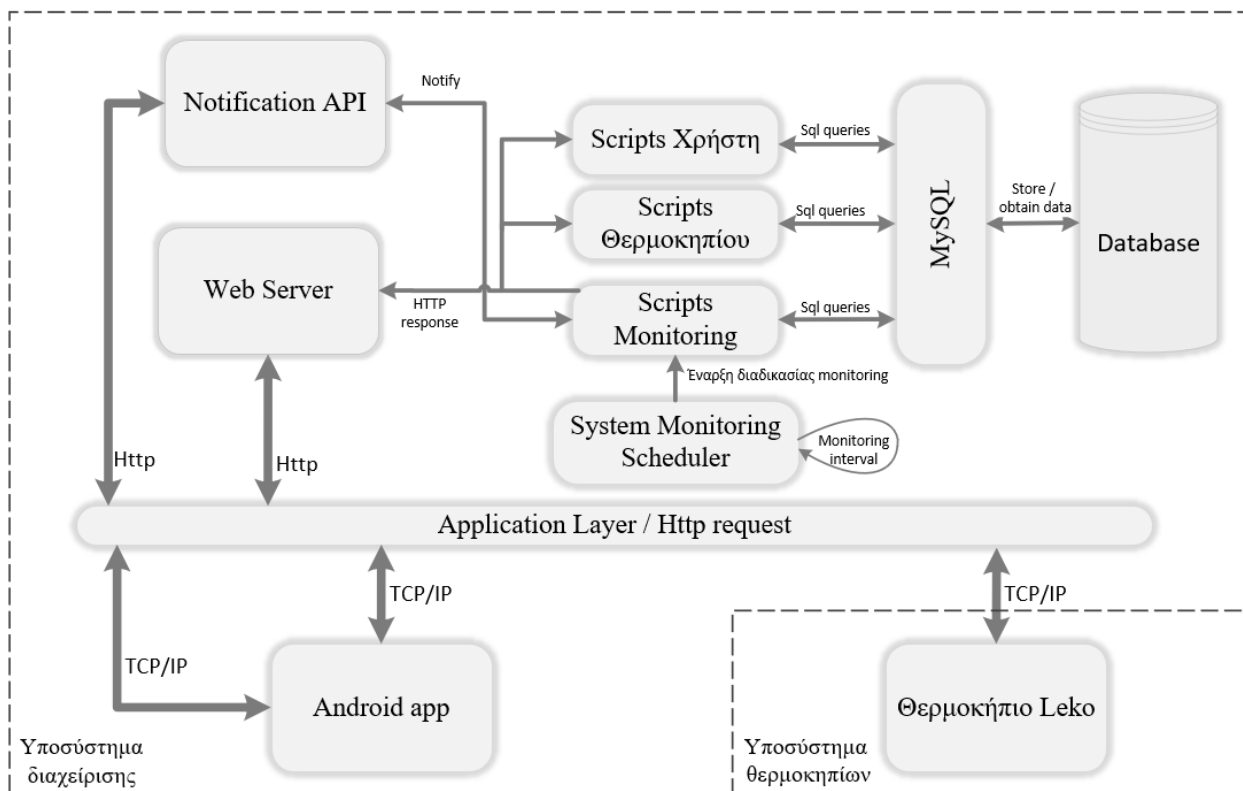
Τα scripts του API, αποτελούνται από τα scripts που αφορούν τον χρήστη, τα scripts που αφορούν το θερμοκήπιο και τα scripts που αφορούν το monitoring του συστήματος.



Σχήμα 4-3. Side server script groups.

4.2 Τελική αρχιτεκτονική συστήματος

Η δομή του τελικού συστήματος, φαίνεται στο Σχήμα 4-4. Τα scripts του API, αλληλοεπιδρούν με την βάση δεδομένων, και με τα αντίστοιχα στοιχεία του συστήματος που αφορούν. Η διαδικασία του ελέγχου που αφορά το monitoring, γίνεται αυτοματοποιημένα μέσω του monitoring scheduler και την εκπνοή του time interval. Σε περίπτωση αστοχίας, το API του συστήματος επικοινωνεί με το Notification API του firebase για την διεκπεραίωση της αποστολής της κάθε ειδοποίησης.



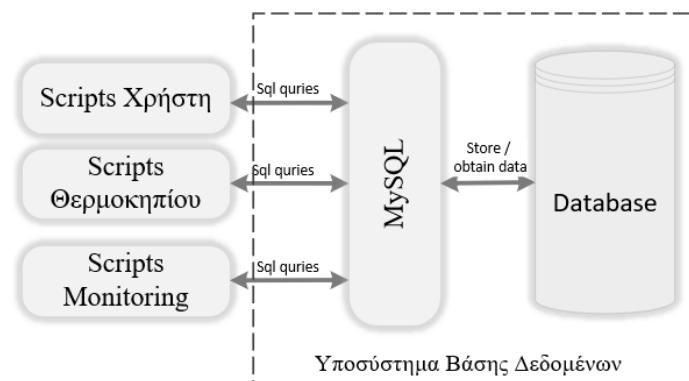
Σχήμα 4-4. Τελική δομή συστήματος.

5. ΣΧΕΔΙΑΣΗ ΣΥΣΤΗΜΑΤΟΣ

Το παρών κεφάλαιο αφορά την σχεδίαση του συστήματος. Τα επιμέρους συστήματα που προέκυψαν, είναι το Backend και το Frontend. Στο πρώτο μέρος σχεδιάζεται το Backend, που απαρτίζεται από τον server, δηλαδή την βάση δεδομένων και το API. Στο δεύτερο μέρος, ακολουθεί η σχεδίαση της εφαρμογής android που αφορά το Frontend του συστήματος.

5.1 Σχεδίαση Βάσης Δεδομένων

Για την εφαρμογή που αφορά την παρούσα διπλωματική, με βάση τα επιμέρους χαρακτηριστικά σχεσιακής και μη σχεσιακής βάσης που παρουσιάστηκαν στην υποενότητα 2.3.1, επιλέχθηκε η MySQL.



Σχήμα 5-1. Υποσύστημα βάσης δεδομένων

Η σχεσιακή λογική υπερτερεί, καθώς τα δεδομένα αφορούν δομημένη και προκαθορισμένη πληροφορία. Η δομημένη ροή της πληροφορίας έγκειται στο γεγονός, ότι οι μετρήσεις των αισθητήρων και τα γεγονότα του θερμοκηπίου έρχονται σε παρτίδες όλα μαζί και όχι τμηματικά.

Επιπροσθέτως, η ακαθόριστη σειρά λήψης των δεδομένων από τα διάφορα θερμοκήπια του οικοσυστήματος, είναι πιο εύκολα διαχειρίσιμα με μία πιο δομημένη αποθήκευση όπως αυτή της MySQL. Επιπλέον, η επεκτασιμότητα της μη σχεσιακής βάσης δεν προσθέτει αξία σε αυτό το στάδιο, καθώς η κλίμακα του συστήματος είναι περιορισμένη.

Στον **Πίνακα 5-1**, παρουσιάζονται τα χαρακτηριστικά που λήφθηκαν υπόψη κατά την διάρκεια της επιλογής. Αξίζει να σημειωθεί, ότι η μετάβαση από σχεσιακή σε μη σχεσιακή λογική σε δεύτερο χρόνο, είναι εφικτή. Μία τέτοια μετάβαση, στο μέλλον, ίσως κριθεί απαραίτητη, τμηματικά, κυρίως στην καταγραφή των περιβαλλοντικών τιμών του θερμοκηπίου, που σε βάθος χρόνου θα αυξάνονται σημαντικά, λόγω της αύξησης των θερμοκηπίων και των αιτημάτων τους.

Πίνακας 5-1. Relational vs Non- Relational DB.		
Τύπος Βάσης	SQL	NoSQL
Τύπος Δεδομένων	Δομημένος(Table based)	Μη Δομημένος (Document Based)
Ευελιξία	<ul style="list-style-type: none"> • Προκαθορισμένο Σχήμα που παραμένει ίδιο • Σχεδίαση όλων των στηλών πριν από την εισαγωγή των δεδομένων • Μετατροπή όλης της βάσης σε περίπτωση οποιασδήποτε αλλαγής 	<ul style="list-style-type: none"> • Μη προκαθορισμένο Σχήμα που αλλάζει συχνά • Δυναμική στον πυρήνα της • Τα δεδομένα μπορούν να εισαχθούν οποιαδήποτε στιγμή, δίχως να απαιτείται σχεδίαση πριν την εισαγωγή
Επεκτασιμότητα	<ul style="list-style-type: none"> • Κάθετη επεκτασιμότητα • Περισσότερα δεδομένα απαιτούν servers μεγαλύτερης χωρητικότητας • Απαιτεί χρόνο και κόστος 	<ul style="list-style-type: none"> • Οριζόντια επεκτασιμότητα • Κατανέμεται σε διαφορετικούς servers • Γρήγορη και αποδοτική κοστολογικά

Στην συνέχεια ακολουθεί ο σχεδιασμός της βάσης που θα εξυπηρετεί το σύστημα διαχείρισης θερμοκηπίων και των οντοτήτων της, λαμβάνοντας υπόψη τις απαιτήσεις που έχουν τεθεί.

5.1.1 Σχεδίαση οντοτήτων

Στην βάση δεδομένων αποθηκεύονται τα απαραίτητα δεδομένα για την λειτουργία του θερμοκηπίου και την επικοινωνία του χρήστη με αυτό. Τα δεδομένα που αποθηκεύονται στην βάση πρέπει να είναι τόσα ώστε να ανταποκρίνεται το σύστημα στις απαιτήσεις που έχουν τεθεί και να εξασφαλίζεται η ορθή λειτουργία του, δίχως πλεονασμούς και σπατάλη πόρων. Ο σχεδιασμός της βάσης του συστήματος είναι πολύ σημαντικός γιατί πέρα από την επικοινωνία χρήστη-θερμοκηπίου, χρησιμοποιείται και για την ανίχνευση σφαλμάτων από την μονάδα διαχείρισης.

Για την λειτουργικότητα του συστήματος, δημιουργήθηκε η οντότητα χρήστη και η οντότητα θερμοκήπιο. Ο σχεδιασμός του συστήματος, απαιτεί από τον χρήστη να δημιουργήσει λογαριασμό στο σύστημα για να μπορεί να διαχειρίζεται τα θερμοκήπιά του.

Για την επίτευξη της monitoring λειτουργίας, δημιουργήθηκαν τρεις επιπλέον οντότητες. Οι οντότητες αυτές είναι, το ιστορικό, τα γεγονότα και οι συσκευές.

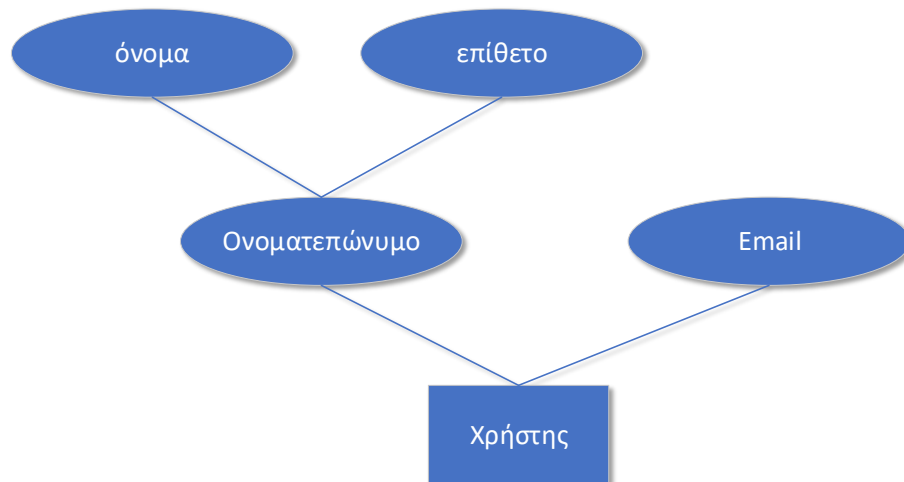
Στην συνέχεια διατυπώνονται τα χαρακτηριστικά της κάθε οντότητας και ο τύπος τους.

- Οντότητα Χρήστης

Τα πεδία της οντότητας χρήστη χωρίζονται σε δύο κατηγορίες. Η μία αναφέρεται στα φυσικά στοιχεία του χρήστη και η άλλη σε στοιχεία που εξυπηρετούν την λειτουργικότητα του συστήματος.

Τα φυσικά στοιχεία είναι το ονοματεπώνυμο και το email του χρήστη. Το ονοματεπώνυμο είναι σύνθετο (composite) χαρακτηριστικό και αποτελείται από το όνομα και το επίθετο του χρήστη. Το email του κάθε χρήστη είναι μοναδικό και θα μπορούσε να αποτελεί το primary key, για λόγους όμως καλύτερης απόδοσης χρησιμοποιείται για αυτό το σκοπό ένα ξεχωριστό id. Έτσι το email αποτελεί ένα από τα secondary keys της παρούσας οντότητας.

Το id που χρησιμοποιείται παράγεται από την βάση με την μέθοδο auto increment. Η προσέγγιση αυτή καθιστά την διαχείριση της βάσεως αποδοτικότερη, ειδικότερα σε μεγάλο πλήθος εγγραφών. Για μεγάλο όγκο χρηστών, οι εγγραφές μπορούν να διασπαστούν σε υποομάδες με την μέθοδο indexing για καλύτερη απόδοση.



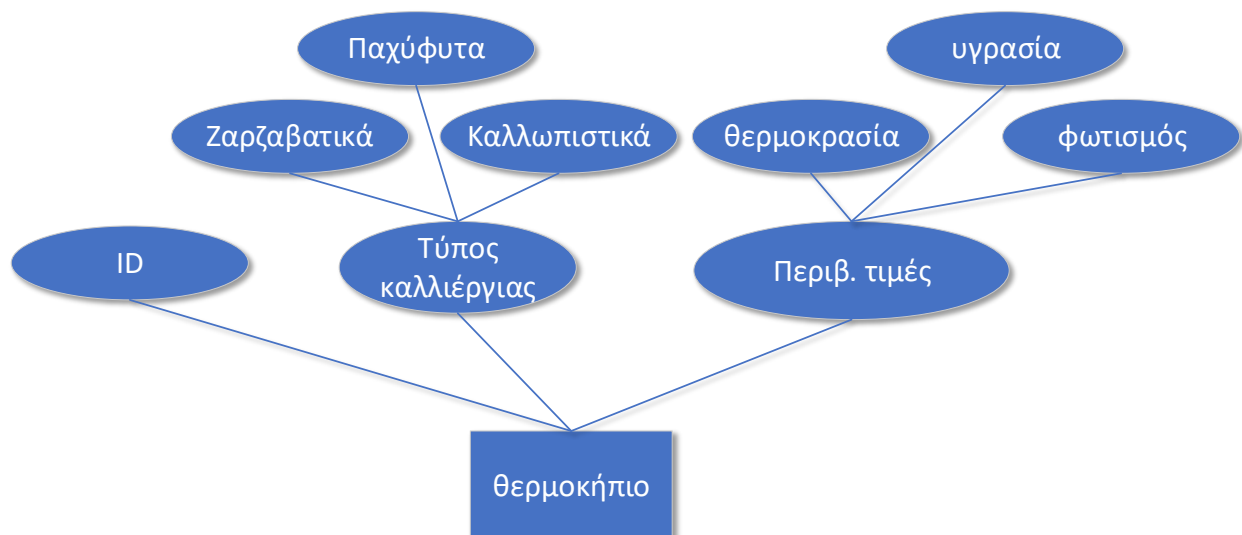
Σχήμα 5-2. Attributes χρήστη.

Τα επιπλέον πεδία που εξυπηρετούν την λειτουργικότητα του συστήματος φαίνονται παρακάτω.

Πίνακας 5-2. Επιπλέον πεδία της οντότητας χρήστη.	
Id	Auto increment user id (Primary key)
Unique_id	Παράγεται μέσω της php μεθόδου uniqid("", true) και είναι ένα αλφαριθμητικό 23 χαρακτήρων. Χρησιμοποιείται για την αύξηση της ασφάλειας κατά την ανταλλαγή πληροφοριών μεταξύ των υποσυστημάτων.
Encrypted_password	Είναι ο κωδικός του χρήστη σε κρυπτογραφημένη μορφή. Η κρυπτογράφηση πραγματοποιείται με την χρήση της base64_encode μεθόδου, του κωδικού του χρήστη και της παραμέτρου salt.
Salt	Η παράμετρος αυτή χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση του κωδικού του χρήστη. Πρόκειται για ένα τυχαίο αλφαριθμητικό 10 χαρακτήρων.
Created_at	Είναι το datetime παραγωγής της καταχώρησης.
Updated_at	Είναι το datetime παραγωγής της ενημέρωσης της καταχώρησης.

- Οντότητα θερμοκήπιο

Η οντότητα του συστήματος του θερμοκηπίου καθορίζεται από τα φυσικά χαρακτηριστικά που φαίνονται στο Σχήμα 5-3. Κύριο κλειδί της οντότητας του θερμοκηπίου είναι το μοναδικό LekoID, αλφαριθμητικό 23 χαρακτήρων, που προκύπτει αυτόματα από το σύστημα μέσω της μεθόδου uniqid(). Επιπλέον χαρακτηριστικά αποτελούν η ημερομηνία εισαγωγής του θερμοκηπίου και οι περιβαλλοντικές τιμές (θερμοκρασία, υγρασία, εδάφους και φωτισμός).

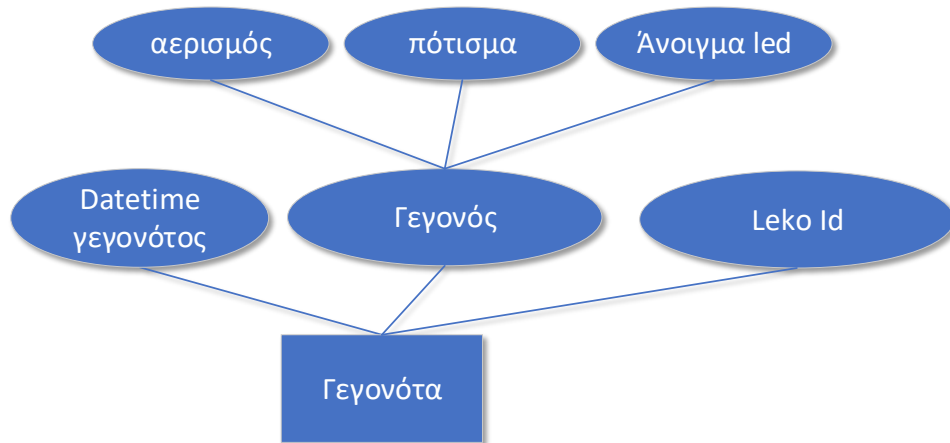


Σχήμα 5-3. Attributes θερμοκηπίου.

- Οντότητα Γεγονότα

Η οντότητα των γεγονότων χρησιμοποιείται για την καταγραφή του ιστορικού ενεργειών της θερμοκηπιακής μονάδας. Το ιστορικό αυτό, απαιτείται για την εξαγωγή

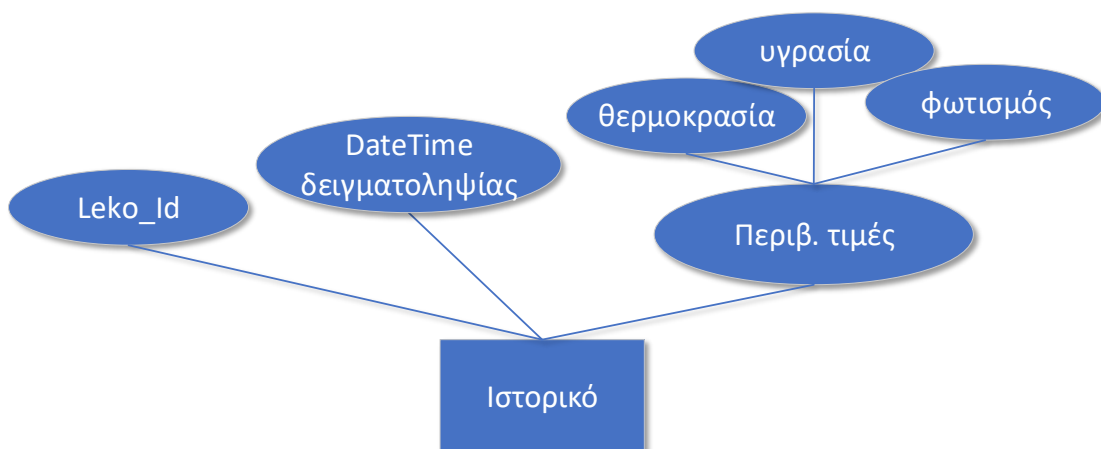
συμπερασμάτων όσο αναφορά την λειτουργία του θερμοκηπίου, αλλά και την ανίχνευση σφαλμάτων. Για παράδειγμα αν καταγραφεί μία ενέργεια ποτίσματος και η υγρασία εδάφους δεν αυξηθεί κατά το αναμενόμενο. Τα δεδομένα που αποθηκεύονται είναι η στιγμή (ημερομηνία και ώρα) του κάθε γεγονότος και το γεγονός (αερισμός, πότισμα και άνοιγμα led).



Σχήμα 5-4. Attributes γεγονότων.

- Οντότητα ιστορικού περιβαλλοντικών τιμών

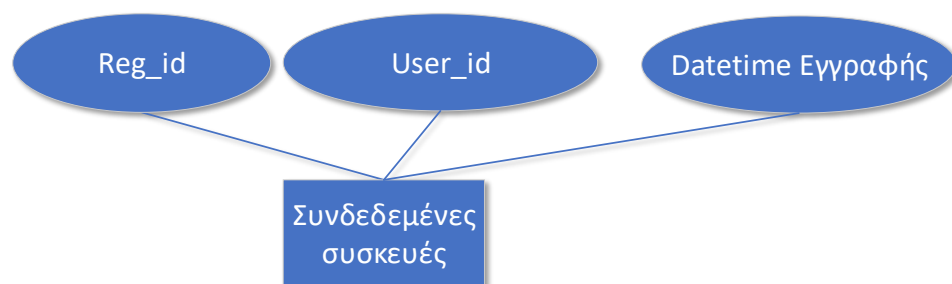
Η οντότητα ιστορικό χρησιμοποιείται για την καταγραφή των περιβαλλοντικών τιμών του εκάστοτε θερμοκηπίου. Καταγράφοντας τις τιμές αυτές μπορούν να ερευνηθούν οι μεταβολές τους σε βάθος χρόνου. Τα συμπεράσματα που προκύπτουν από την ανάλυση των δεδομένων ιστορικού, είναι πολύτιμα για το σύστημα και για τους χρήστες, και μπορούν να οδηγήσουν σε περαιτέρω βελτίωση του συστήματος του θερμοκηπίου.



Σχήμα 5-5. Οντότητα Ιστορικό

- Οντότητα συσκευών χρήστη

Ο κάθε χρήστης μπορεί να χρησιμοποιεί παραπάνω από μία συσκευές λειτουργικού android για την διαχείριση του θερμοκηπίου του. Οι συσκευές αυτές αποθηκεύονται στην βάση δεδομένων του συστήματος. Αυτό συμβαίνει για να μπορεί το σύστημα διαχείρισης να στέλνει ειδοποιήσεις σε όλες τις συνδεδεμένες συσκευές όταν αυτό κρίνεται απαραίτητο.

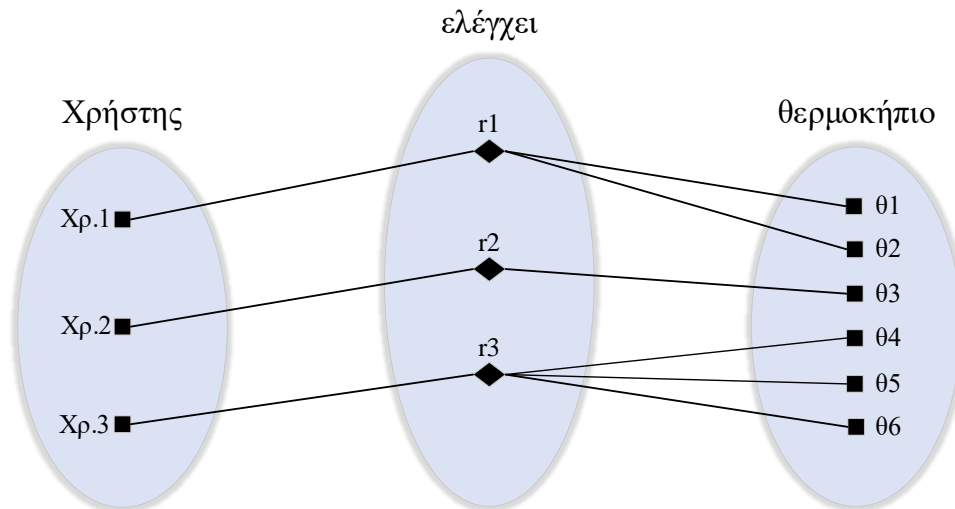


Σχήμα 5-6. Attributes συνδεδεμένων συσκευών.

Το Reg_id είναι ένα μοναδικό κλειδί που χαρακτηρίζει την συσκευή. Παράγεται από το σύστημα Firebase ως device token ID και χρησιμοποιείται για την αποστολή ειδοποιήσεων.

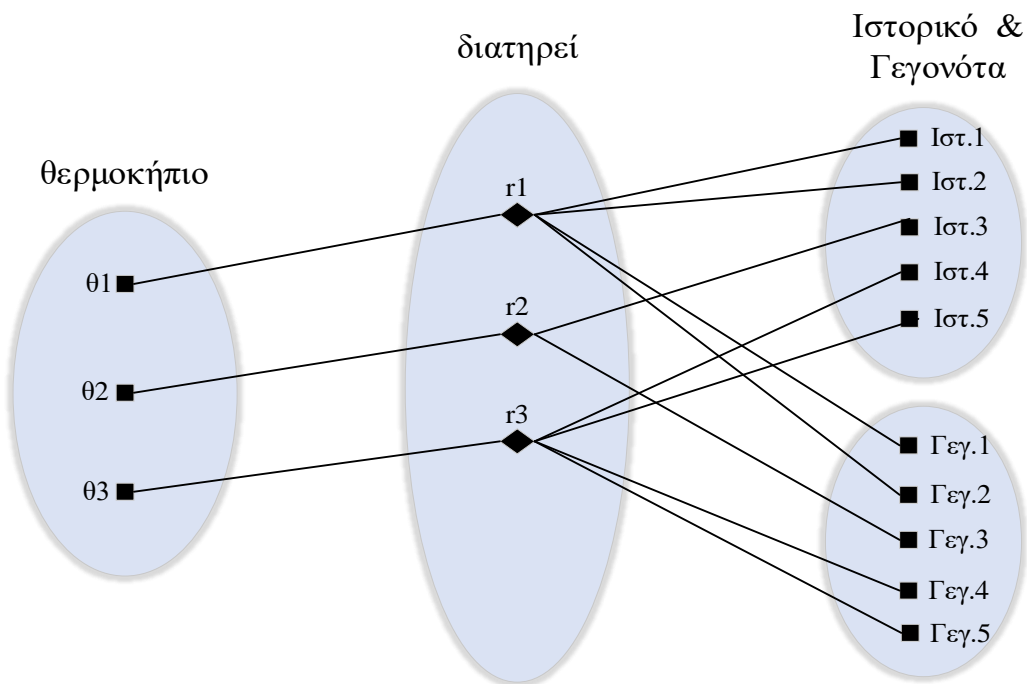
5.1.2 Τελικό σχήμα βάσης

Όπως και σε κάθε σχεσιακή βάση δεδομένων πρέπει να καθοριστούν οι συσχετίσεις μεταξύ των παραπάνω οντοτήτων και οι περιορισμοί αυτών. Στο παρόν σύστημα κάθε χρήστης ελέγχει τουλάχιστον ένα θερμοκήπιο (cardinality 1:N). Επίσης, δεν υπάρχει χρήστης χωρίς θερμοκήπιο, ούτε θερμοκήπιο χωρίς χρήστη στο σύστημα (total participation). Οι παραπάνω περιορισμοί είναι πολύ σημαντικοί και καθορίζουν την λειτουργία ολόκληρου του συστήματος. Στο παρόν σύστημα έχει επιλεγθεί να μην μπορεί να καταχωρηθεί κάποιος χρήστης μόνος του χωρίς να έχει έστω ένα θερμοκήπιο να καταχωρήσει. Η πλήρης συμμετοχή των δύο οντοτήτων καθώς και ο λόγος πληθικότητας τους φαίνεται στο Σχήμα 5-7.



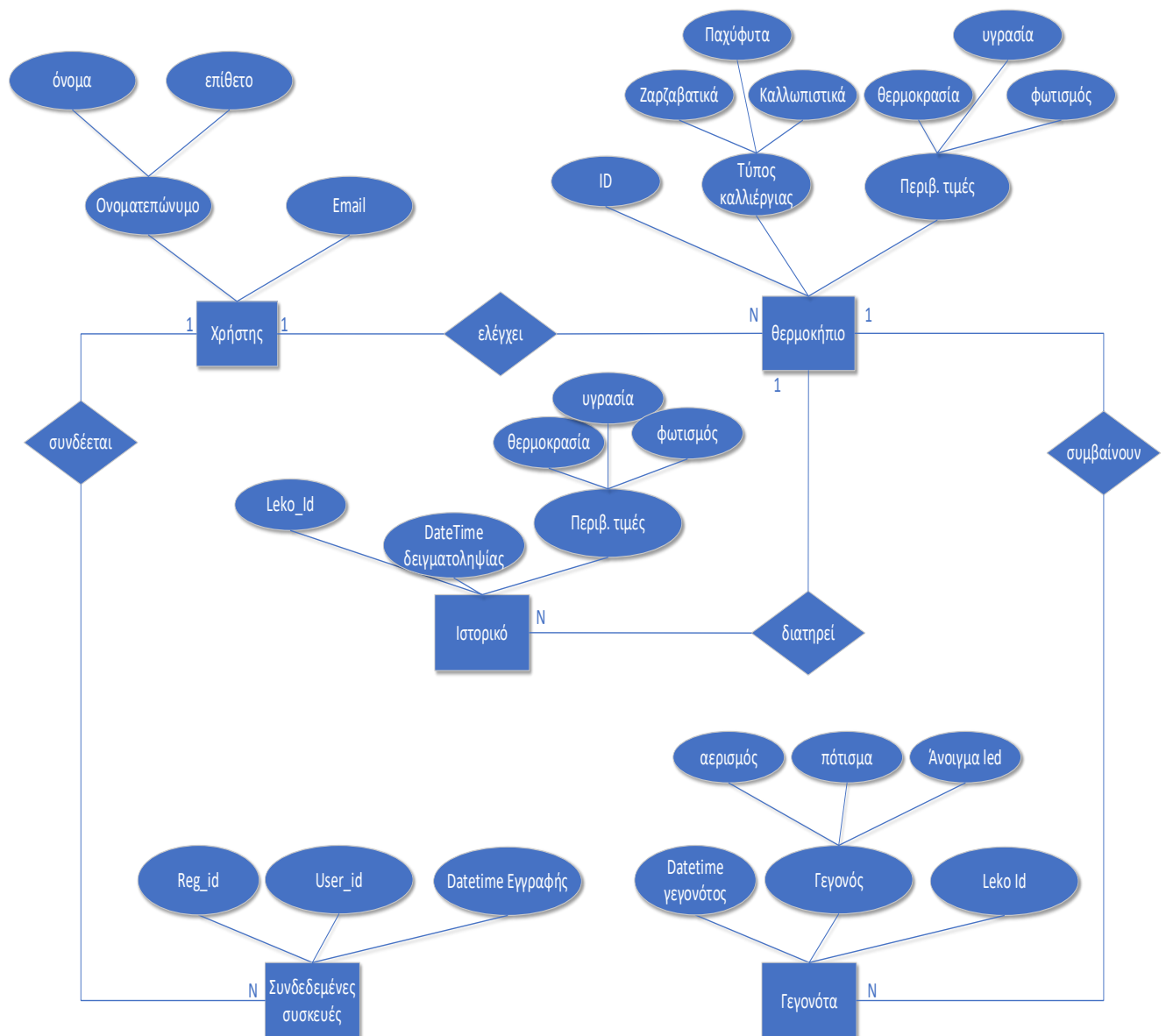
Σχήμα 5-7. Εξαρτήσεις μεταξύ χρήστη & θερμοκηπίου

Όσο αναφορά το θερμοκήπιο και τα δεδομένα που συλλέγει ισχύουν τα ακόλουθα : κάθε θερμοκήπιο έχει τουλάχιστον ένα ιστορικό και ένα γεγονός. Το σύστημα λαμβάνει το ιστορικό και το γεγονός στο ίδιο update. Επομένως για κάθε ιστορικό που διατηρεί το θερμοκήπιο, διατηρεί και τα αντίστοιχα γεγονότα. Οι εξαρτήσεις μεταξύ θερμοκηπίου, οντότητας ιστορικού και γεγονότος, φαίνονται στο Σχήμα 5-8.



Σχήμα 5-8. Εξαρτήσεις μεταξύ θερμοκηπίου & ιστορικού/γεγονότος

Ολόκληρο το μοντέλο της βάσης φαίνεται στο Σχήμα 5-9.



Σχήμα 5-9 . Relational Schema βάσης συστήματος.

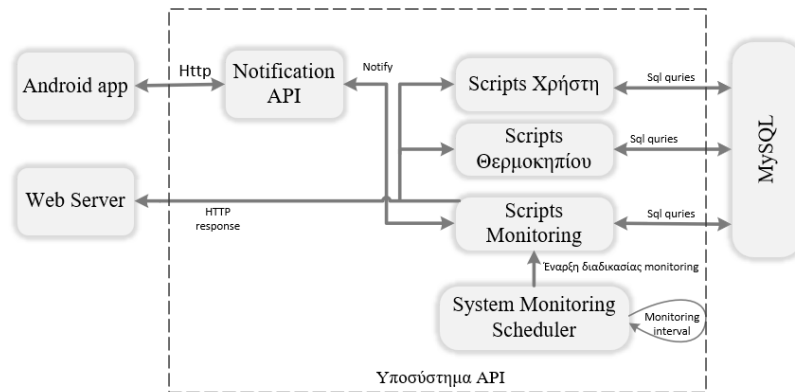
5.2 Σχεδίαση API

Το API του συστήματος εξυπηρετεί δύο ρόλους.

Ο πρώτος ρόλος, είναι να εξυπηρετεί τα αιτήματα του χρήστη και του θερμοκηπίου, αλληλοεπιδρώντας με την βάση δεδομένων και απαντώντας σε κάθε ένα από τα αιτήματα αυτά. Ουσιαστικά ο διαχειριστής μετατρέπει τα αιτήματα αυτά στα ανάλογα queries της βάσης δεδομένων.

Ο δεύτερος ρόλος του είναι να εξυπηρετεί την monitoring λειτουργία του συστήματος, δρώντας επισκοπικά σε όλο το σύστημα και να ενημερώνει τον χρήστη για πιθανές αστοχίες του θερμοκηπίου.

Η σχεδίαση του API επιτρέπει την εξυπηρέτηση αιτημάτων από οποιοδήποτε client. Είναι επίσης εφικτό να λειτουργήσει και σε διαφορετική βάση που διέπει τους ίδιους κανόνες και διατηρεί την ίδια δομή δεδομένων.



Σχήμα 5-10. Υποσύστημα API

Οι δύο βασικές αρχιτεκτονικές API αναφέρθηκαν στην υποενότητα 2.3.2. Τα βασικά τους χαρακτηριστικά φαίνονται και στον **Πίνακα 5-3**. Το REST επιλέχθηκε έναντι του SOAP, γιατί τα χαρακτηριστικά του ανταποκρίνονται περισσότερο στο σύστημα. Τα πλεονεκτήματα του SOAP, όπως η αυξημένη ασφάλεια που παρέχει, δεν αφορούν το πρώτο στάδιο της σχεδίασης. Για την θωράκιση του συστήματος, σχεδιάστηκε μηχανισμός API authentication, που αναφέρεται σε επόμενη υποενότητα.

Πίνακας 5-3. REST vs SOAP.		
Τύπος API	REST	SOAP
Business Logic	URIs	Service interfaces
Response Format	Διαφορετικά data formats όπως JSON & XML	XML
Ταχύτητα	Γρήγορο – Χρησιμοποιεί λιγότερο bandwidth	Μεγαλύτερες καθυστερήσεις – Χρησιμοποιεί περισσότερο bandwidth
Ασφάλεια	Έλλειψη προκαθορισμένου μηχανισμού ασφαλείας	Προκαθορισμένος μηχανισμός ασφαλείας

5.2.1 API Documentation

Η scripting language του API που επιλέχθηκε, είναι η PHP. Τα χαρακτηριστικά των γλωσσών ανάπτυξης, φαίνονται στον **Πίνακα 5-4**. Η PHP θα χρησιμοποιηθεί για την αλληλεπίδραση με τη βάση δεδομένων MySQL. Τα αιτήματα που θα εξυπηρετεί θα προέρχονται από τους δύο clients του συστήματος, τον χρήστη μέσω της εφαρμογής Android, και του θερμοκηπίου, όπως ορίστηκε και στην μοντελοποίηση του συστήματος. Για να εκπληρωθεί ένα αίτημα πρέπει να είναι γνωστή διεύθυνση URL της PHP μεθόδου που το υλοποιεί, καθώς και να προσκομιστούν

από την μεριά του χρήστη οι απαραίτητοι παράμετροι της μεθόδου. Οι παράμετροι αυτοί αφορούν το MySQL query που θα εκτελεστεί, καθώς και επιπλέον data που ενδεχομένως απαιτεί η μέθοδος. Για κάθε ένα από τα αιτήματα του client εκτελείται και διαφορετική PHP μέθοδος, ακολουθώντας την RESTful προσέγγιση.

Η PHP, πέρα από τον εναρμονισμό της με το πρωτόκολλο HTTP, μπορεί να διαχειριστεί πληθώρα διαφορετικών πρωτοκόλλων, όπως FTP, SSL και άλλα[36]. Το γεγονός αυτό προσφέρει την ευελιξία της αναβάθμισης στο μέλλον, όπου για παράδειγμα κάθε θερμοκήπιο θα στέλνει και φωτογραφίες στον server, μέσω FTP.

Πίνακας 5-4. PHP vs PYTHON vs RUBY.

Scripting Language	PHP	PYTHON	RUBY
Community support	HIGH	HIGH	HIGH
Συνδεσιμότητα με Databases	Πολύ Καλή. Συνδεσιμότητα με παραπάνω από 20 είδη βάσεων δεδομένων	Καλή. Όχι τόσο ευρεία όσο η PHP.	Πολύ καλή, αλλά μόνο με την χρήση framework, όπως ruby on rails
Ταχύτητα	Πολύ γρήγορες εκτελέσεις	Γρήγορες εκτελέσεις	Πολύ γρήγορες εκτελέσεις
Γνωστές εφαρμογές που την χρησιμοποιούν	Facebook Wikipedia	Google Youtube	Twitter

Η λειτουργικότητα του API, όπως αναφέρθηκε και παραπάνω, βασίζεται στις μεθόδους που διαθέτει με RESTful προσέγγιση. Η κάθε μία από αυτές τις PHP μεθόδους, εκτελείται από τους clients, στην διεύθυνση του αντίστοιχου URL, χρησιμοποιώντας το HTTP πρωτόκολλο. Τα URL αυτά διαμορφώνουν την αλληλοεπίδραση του API με τους clients, και αποτελούν και τα endpoints του API.

Τα endpoints του API διακρίνονται σε τρεις κατηγορίες. Τα endpoints του:

- χρήστη
- του θερμοκηπίου
- και του συστήματος

αναλόγως με την οντότητα που εξυπηρετούν. Στον **Πίνακα 5-5** παρουσιάζονται τα endpoints του χρήστη, στον **Πίνακα 5-6** τα endpoints του θερμοκηπίου και στον **Πίνακα 5-7** τα endpoints του συστήματος, καθώς και η περιγραφή τους.

Πίνακας 5-5. API documentation- User Endpoints

Τίτλος: Περιγραφή	Http Μέθοδος	URL παράμετροι	URL
Register: Εγγραφή χρήστη	Post	{name, email, password}	/lekoapp/register.php
Login: Σύνδεση χρήστη	Post	{email, password}	/lekoapp/login.php
Add Leko: Προσθήκη Leko	Post	{userId, type}	/lekoapp/lekoadd.php
Leko count of a user: Επιστροφή αριθμού εγγεγραμμένων Leko του χρήστη στο σύστημα	Post	{userId}	/lekoapp/lekonum.php
Get all Lekos of a user: Επιστροφή όλων των αντικειμένων Leko ενός χρήστη	Post	{userId}	/lekoapp/lekoreturn.php
Weekly environmental values: Επιστροφή εβδομαδιαίων μέσων περιβαλλοντικών τιμών	Post	{id, type}	/lekoapp/history.php
Weekly actuator activations: Επιστροφή εβδομαδιαίων μέσων ενεργοποιήσεων/γεγονότων	Post	{id,type}	/lekoapp/eventhistory.php
Change User credentials: Αλλαγή στοιχείων χρήστη	Post	{userId}	/lekoapp/ changeUsernamePassword.php
Delete Leko: Διαγραφή συγκεκριμένου Leko	Post	{lekoId}	/lekoapp/delete.php
Add device's firebase token to dB: Προσθήκη διακριτικού firebase token στο σύστημα	Post	{userId, regId}	/lekoapp/addDevice.php
Delete device's firebase token to dB: Διαγραφή διακριτικού firebase token από το σύστημα	Post	{userId, regId}	/lekoapp/ deleteDevice.php

Πίνακας 5-6. API documentation - Leko Endpoints

Τίτλος: Περιγραφή	Http Μέθοδος	URL παράμετροι	URL
Post Values: Αποστολή κατάστασης Leko, περιβαλλοντικών τιμών και γεγονότων (ενεργοποιητών)	Post	lekoId, humidity, temperature, lighting, pump, fan, led	/lekoapp/addevent.php

Πίνακας 5-7. API documentation-System Endpoints

Τίτλος: Περιγραφή	Http Μέθοδος	URL παράμετροι	URL
Check Values: Αποστολή ειδοποιήσεων στους χρήστες των Leko, των οποίων οι τιμές αποκλίνουν	Get	-	/lekoapp/systemMonitoring/checkLekos.php
Detect Inactivity: Αποστολή ειδοποιήσεων στους χρήστες των Leko, που έχουν να ανανεώσουν το σύστημα κάποιο συγκεκριμένο χρονικό διάστημα	Get	-	/lekoapp/systemMonitoring/checkInactive.php

5.2.2 Αιτήματα Χρήστη

Τα αιτήματα του χρήστη αντιστοιχούν στα endpoints του API στον **Πίνακας 5-5**. Παρακάτω ακολουθεί η περιγραφή της λειτουργικότητας τους.

- Register χρήστη

Για να εγγραφεί ο χρήστης στο σύστημα απαιτείται να συμπληρώσει το ονοματεπώνυμό του, το email του, και τον κωδικό που επιθυμεί. Ο κωδικός κρυπτογραφείται από τον διαχειριστή με την μέθοδο base64 και την χρήση ενός τυχαίου αλφαριθμητικού salt, και έπειτα καταχωρούνται σε μία εγγραφή στον πίνακα users μαζί με τα παραπάνω στοιχεία. Με την χρήση του αλφαριθμητικού αυτού επιτυγχάνεται η αντίστροφη διαδικασία της αποκρυπτογράφησης κατά το Login του χρήστη, ούτως ώστε να επαληθευτεί ο κωδικός. Η κρυπτογράφηση είναι απαραίτητη για να διασφαλιστεί ο κωδικός του χρήστη κατά την ανταλλαγή δεδομένων μεταξύ του χρήστη και του webserver αλλά και την καλύτερη πρόληψη έναντι κακόβουλων επιθέσεων.

- **Login Χρήστη**
Για να συνδεθεί ο χρήστης στο σύστημα απαιτείται να συμπληρώσει το username και τον κωδικό του. Ο διαχειριστής κάνει μία αναζήτηση στον πίνακα των user για να ταυτοποιηθεί η παρουσία του χρήστη στο σύστημα. Εάν η αναζήτηση αυτή είναι επιτυχής το API απαντά επιτυχώς στον χρήστη και επιτρέπεται η σύνδεση στο σύστημα.
- **Αλλαγή Username/Password**
Ο χρήστης μπορεί να αλλάξει τα στοιχεία του μέσω της εφαρμογής. Για να συμβεί αυτό ο χρήστης θα πρέπει να είναι πρώτα συνδεδεμένος στο σύστημα.
- **Προσθήκη θερμοκηπίου Leko**
Ο χρήστης μπορεί να προσθέσει κάποιο θερμοκήπιο Leko μέσω της εφαρμογής. Κατά την προσθήκη ο χρήστης καλείται να επιλέξει τον τύπο της καλλιέργειας του νέου θερμοκηπίου. Η προσθήκη πραγματοποιείται με την δημιουργία μία εγγραφής στον πίνακα Leko, με foreign key το userId του χρήστη. Μετά από την εισαγωγή, το LekoId που αντιστοιχεί στο θερμοκήπιο που προστέθηκε εμφανίζεται στον χρήστη. Αυτό το διακριτικό, προστίθεται μέσω του τερματικού στο θερμοκήπιο για να μπορεί να επικοινωνεί με τον Server.
- **Διαγραφή θερμοκηπίου Leko**
Ο χρήστης μπορεί να αφαιρέσει κάποιο από τα θερμοκήπια του. Στην περίπτωση αυτή η εγγραφή του θερμοκηπίου από τον πίνακα Leko αφαιρείται.
- **Επιστροφή αριθμού θερμοκηπίων χρήστη**
Το API μετρά τα στοιχεία του πίνακα Leko για δεδομένο userId, και επιστρέφει τον αριθμό στον χρήστη.
- **Επιστροφή Lekos χρήστη**
Το API επιστρέφει ολόκληρες τις εγγραφές του πίνακα Lekos για δεδομένο userId. Στην περίπτωση αυτή τα θερμοκήπια είναι προσπελάσιμα από τον χρήστη με όλες τις τιμές τους.
- **Εβδομαδιαίο Report θερμοκηπίου**
Το API εξάγει τις εβδομαδιαίες περιβαλλοντικές τιμές και την συνολική δραστηριότητα των ενεργοποιητών του εκάστοτε θερμοκηπίου από τον πίνακα history και events. Για την εξοικονόμηση του bandwidth του δικτύου, το report αφορά τις μέσες τιμές της κάθε ημέρας για τις τελευταίες επτά ημέρες. Οι τιμές αυτές στέλνονται στον client για να προβληθούν στον χρήστη σε μορφή γραφήματος. Για να εξαχθούν οι τιμές αυτές χρησιμοποιείται το μοναδικό αναγνωριστικό lekoId του θερμοκηπίου, και επιλέγεται το χρονικό περιθώριο των 7 ημερών.

5.2.3 Αιτήματα θερμοκηπίου

Το API δέχεται αιτήματα από τα θερμοκήπια, για να ενημερώσει την βάση δεδομένων για την κατάσταση του συστήματος. Τα αιτήματα που δέχεται αφορούν την ανανέωση των περιβαλλοντικών τιμών και την ενημέρωση της δραστηριότητάς των ενεργοποιητών.

- **Ανανέωση περιβαλλοντικών τιμών**
Οι τιμές που λαμβάνει το API από κάθε θερμοκήπιο εισάγονται στον πίνακα history. Για κάθε ενημέρωση, δημιουργείται και μία εγγραφή στον πίνακα. Ταυτόχρονα, ανανεώνεται και ο πίνακας Leko, με τις τελευταίες τιμές.
- **Ανανέωση γεγονότων θερμοκηπίου**
Αιτήματα από το θερμοκήπιο λαμβάνονται και για την λειτουργία των ενεργοποιητών. Οι ενεργοποιητές αυτοί είναι η αντλία νερού, ο ανεμιστήρας και το σύστημα φωτισμού. Κατά την ενεργοποίηση ενός εξ' αυτών στέλνεται αίτημα από το θερμοκήπιο για ενημέρωση. Η καταγραφή της ενημέρωσης αυτής γίνεται με την ανάλογη εγγραφή στον πίνακα events της βάσης δεδομένων.

5.2.4 Monitoring συστήματος

Το monitoring του συστήματος, θα μπορούσε να υλοποιηθεί και με κάποιο τοπικό API, το οποίο δεν θα είναι διαθέσιμο στο διαδίκτυο για απομακρυσμένη ενεργοποίηση της διαδικασίας. Ο λόγος που προτιμήθηκε να ενσωματωθεί και αυτή η λειτουργικότητα στο υπόλοιπο API, είναι για να μπορεί να εκτελεστεί και απομακρυσμένα από τους admins του συστήματος. Το αρνητικό με την προσέγγιση αυτή, είναι ότι εκθέτει το σύστημα δημιουργώντας ένα κενό ασφαλείας. Για τον λόγο αυτό, δημιουργείται η ανάγκη για κάποιο σύστημα επικαιροποίησης των clients, που περιγράφεται σε επόμενη υποενότητα.

Το API, υλοποιεί το monitoring των θερμοκηπίων με δύο μεθόδους. Η μία ελέγχει την ορθότητα των περιβαλλοντικών τιμών (**checkLekos()**) και η άλλη την τελευταία ενημέρωση από κάθε θερμοκήπιο(**checkInactive()**). Οι δύο αυτές μέθοδοι, εκτελούνται αυτόματα μέσω ενός job scheduler, ή απομακρυσμένα από τους admin του συστήματος, με την εκτέλεση των αντίστοιχων μεθόδων.

- **Έλεγχος περιβαλλοντικών τιμών**
Μέσω της μεθόδου checkLekos(), ελέγχεται, εάν οι περιβαλλοντικές τιμές είναι στα επιτρεπτά όρια. Ανά έξι ώρες το API μέσω ενός job scheduler, ανατρέπει στον πίνακα Lekos. Ανάλογα με τον τύπο της καλλιέργειας προκύπτουν και τα έγκυρα όρια περιβαλλοντικών τιμών. Για κάθε μη έγκυρο όριο ο διαχειριστής ειδοποιεί τον χρήστη μέσω ειδοποίησης στο κινητό.

Οι περιβαλλοντικές τιμές που συνίστανται για την κάθε οικογένεια φυτών φαίνονται στον **Πίνακας 5-8**.

Πίνακας 5-8. Συνιστώμενα χωρία περιβαλλοντικών τιμών ανά τύπο καλλιέργειας [50]				
Οικογένεια Φυτών	Κωδικοποίηση οικ. Φυτών	Θερμοκρασία (Celsius)	Υγρασία (%)	Φωτισμός (ώρες/ημέρα)
Ζαρζαβατικά	type1	30 – 40	55 – 65	8-24
Παχύφυτα	type2	45 – 55	5 – 15	4-14
Καλλωπιστικά	type3	35 - 45	25 - 35	6-18

- Έλεγχος συνδεσιμότητας θερμοκηπίου

Για να πιστοποιηθεί η άρτια επικοινωνία με το κάθε θερμοκήπιο, μέσω της μεθόδου `checkInactive()`, ελέγχεται η χρονική στιγμή της τελευταίας επικοινωνίας. Εάν το διάστημα αυτό είναι μεγαλύτερο των 12 ωρών, εκτιμάται ότι υπάρχει πρόβλημα στην συνδεσιμότητα του θερμοκηπίου με τον server. Το πρόβλημα αυτό μπορεί να οφείλεται στο τοπικό δίκτυο του θερμοκηπίου. Όταν εντοπιστεί, ο χρήστης ενημερώνεται με `notification` σε όλες τις συνδεδεμένες του συσκευές.

- Σύστημα Notifications

Για την επίβλεψη του συστήματος του θερμοκηπίου ο χρήστης ειδοποιείται όταν ανιχνευθεί σφάλμα στην λειτουργία ενός θερμοκηπίου.. Το `notification` αποστέλλεται στον χρήστη ακόμα και όταν η εφαρμογή είναι κλειστή. Για να συμβεί αυτό, αποθηκεύεται στην βάση ένα μοναδικό ID για κάθε συσκευή, που χρησιμοποιεί ο χρήστης για να συνδεθεί στο σύστημα. Ο χρήστης μπορεί να είναι συνδεδεμένος και με παραπάνω από μία συσκευές, για παράδειγμα στο κινητό και στο tablet του. Στην βάση, καταχωρούνται όλες οι συσκευές με τις οποίες είναι συνδεδεμένος ο χρήστης.

Οι ειδοποιήσεις χωρίζονται σε δύο κατηγορίες. Η πρώτη κατηγορία είναι ειδοποιήσεις των κρίσιμων συμβάντων. Σε αυτή την περίπτωση ο χρήστης ειδοποιείται όταν κάποια από τις περιβαλλοντικές τιμές του φυτού που έχει καλλιεργηθεί αποκλίνει των συνιστώμενων ή όταν το θερμοκήπιο δείχνει να είναι εκτός σύνδεσης.

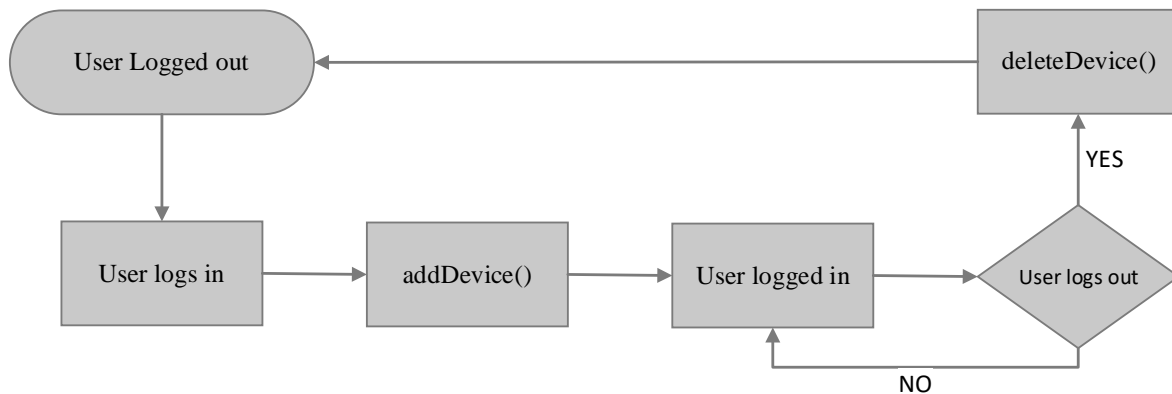
Η δεύτερη κατηγορία ειδοποιήσεων είναι ειδοποιήσεις ενημερωτικού τύπου. Χρησιμοποιώντας αυτές τις ειδοποιήσεις, οι χρήστες του συστήματος είναι σε θέση να ενημερωθούν για βελτιώσεις, ενημερώσεις ή αστοχίες του συστήματος. Για την αποστολή αυτών των ειδοποιήσεων έχει δημιουργηθεί ένα γραφικό περιβάλλον σε HTML, στο οποίο μπορεί να συμπληρωθεί ο κωδικός θερμοκηπίου και το μήνυμα που απαιτείται να σταλεί.

Για την υλοποίηση των `notification` χρησιμοποιήθηκε το `firebase API` της Google. Οι ειδοποιήσεις λαμβάνονται από την συσκευή όταν αυτή έχει πρόσβαση στο διαδίκτυο. Σε περίπτωση που η συσκευή δεν είναι συνδεδεμένη, η ειδοποίηση μένει σε αναμονή για είκοσι οχτώ ημέρες. Μετά το πέρας των είκοσι οχτώ ημερών, εάν η κινητή συσκευή παραμένει εκτός σύνδεσης το μήνυμα διαγράφεται. [55]

- Διαχείριση συσκευών

Για την λειτουργία των ειδοποιήσεων είναι απαραίτητη η σωστή καταγραφή και διαχείριση των ενεργών συσκευών κάθε χρήστη. Οι συσκευές που αποθηκεύονται στο σύστημα είναι αυτές με τις οποίες ο χρήστης παραμένει συνδεδεμένος. Την στιγμή που ο χρήστης αποσυνδεθεί από μία συσκευή, αυτή διαγράφεται από το σύστημα. Αυτό είναι

απαραίτητο για να στέλνονται οι ειδοποιήσεις στις κατάλληλες συσκευές. Κατά την σύνδεση ή την πρώτη εγγραφή του χρήστη λοιπόν ο διαχειριστής μέσω της μεθόδου `addDevice()` προσθέτει το αναγνωριστικό `regId` της συσκευής αυτής στον πίνακα συνδεδεμένες συσκευές της βάσης. Όταν ο χρήστης αποσυνδεθεί από την συσκευή, αφαιρείται η αντίστοιχη εγγραφή από τον πίνακα μέσω της μεθόδου `deleteDevice()`.



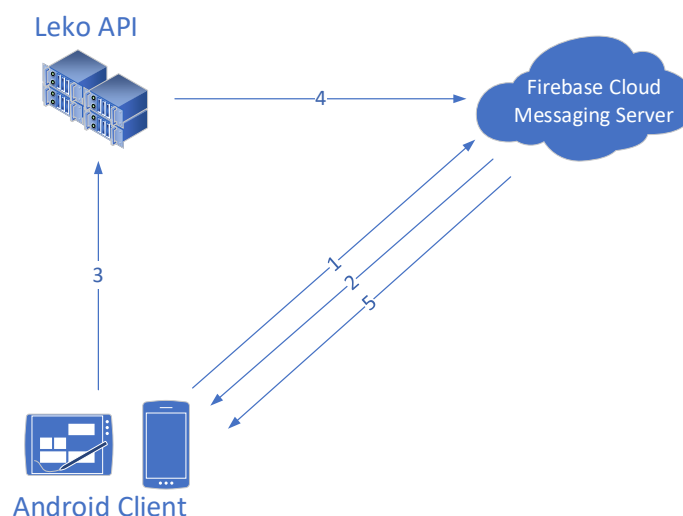
Σχήμα 5-11. Διαδικασία διαχείρισης συσκευών.

5.2.5 Ροή λειτουργίας συστήματος Notification

Κατά την αρχική εκκίνηση της εφαρμογής, δημιουργείται ένα διακριτικό εγγραφής για την περίπτωση της συγκεκριμένης εφαρμογής-πελάτη. Χρησιμοποιώντας αυτό το διακριτικό μπορούν να στοχευθούν μεμονωμένες συσκευές ή να δημιουργηθούν ομάδες συσκευών.

Η ενσωμάτωση του firebase API στο backend του συστήματος γίνεται με την χρήση του Firebase Server API key. Το κλειδί αυτό χρησιμοποιείται από τον διαχειριστή για να επιτευχθεί το σύστημα ειδοποιήσεων.

Η ροή λειτουργίας του συστήματος ειδοποιήσεων, φαίνεται στο Σχήμα 5-12.



Σχήμα 5-12. Ροή λειτουργίας Firebase Cloud Messaging

1. Πρώτα η εφαρμογή συνδέεται με τον FCM server και καταχωρείται.
2. Μετά την επιτυχή εγγραφή, ο FCM εκδίδει διακριτικό εγγραφής fcm token στην συσκευή και δημιουργείται το socket επικοινωνίας. Αυτό το σύμβολο εγγραφής αναγνωρίζει με μοναδικό τρόπο κάθε συσκευή.
3. Η συσκευή στέλνει το διακριτικό εγγραφής στο διαχειριστή για να το αποθηκεύσει στην βάση δεδομένων στον πίνακα διαχείριση συσκευών. Κατά την αποσύνδεση του χρήστη η συσκευή στέλνει το διακριτικό εγγραφής στον διαχειριστή για να το αφαιρέσει από την βάση δεδομένων.
4. Ο διαχειριστής θέλει να στείλει μία ειδοποίηση, αποστέλλει αίτημα στον FCM server να την στείλει μαζί με το διακριτικό εγγραφής της συσκευής.
5. Ο FCM server αναγνωρίζει την συσκευή χρησιμοποιώντας το διακριτικό εγγραφής και εκκινεί το μήνυμα ώθησης. Η συσκευή λαμβάνει τα μηνύματα ώθησης.

5.2.6 Υλοποίηση API authentication

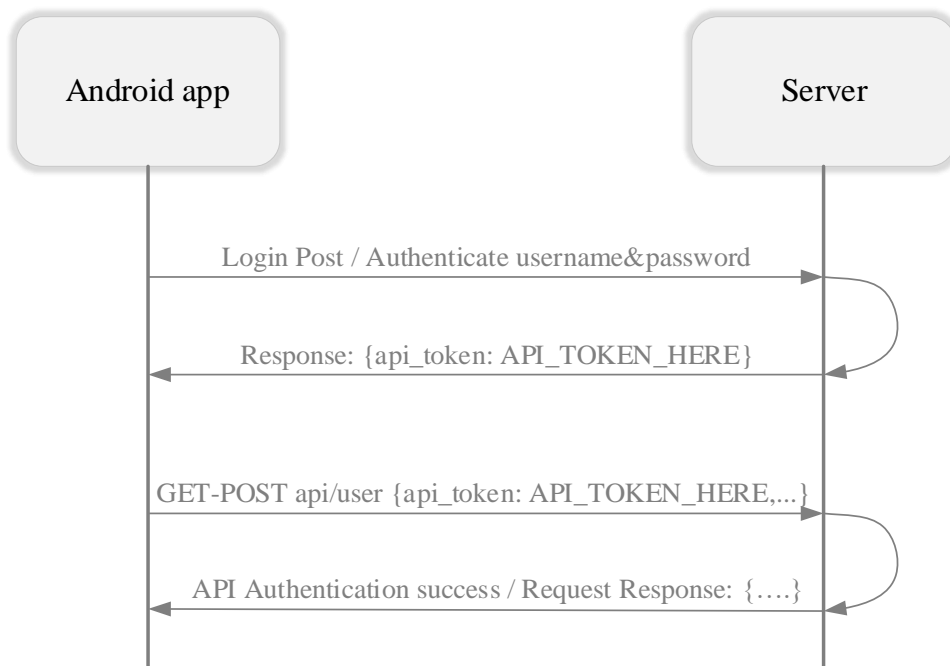
Για να διασφαλιστεί η ασφάλεια του συστήματος, θα πρέπει να εξασφαλιστεί ότι το API του συστήματος εξυπηρετεί τους εξουσιοδοτημένους clients και μόνο. Για να επιτευχθεί αυτό, θα πρέπει να εφαρμοστεί σύστημα ταυτοποίησης στο API, γνωστό και ως API authentication, όπως αναφέρθηκε και στην ενότητα 2.3.3 **Error! Reference source not found.**

Η μέθοδος που επιλέχθηκε είναι η API key μέθοδος μέσω του HTTP header. Ο λόγος που επιλέχθηκε είναι ότι δεν απαιτεί κάποιο συγκεκριμένο framework ανάπτυξης (π.χ. Lavarel) όπως οι υπόλοιπες μέθοδοι, αλλά μπορεί να ενσωματωθεί απευθείας στα PHP files του API.

Η γενική ιδέα πίσω από το σύστημα αυτό είναι απλή. Επιτρέπεται στους χρήστες να εισάγουν το όνομα χρήστη και τον κωδικό πρόσβασής τους, για να αποκτήσουν ένα μοναδικό διακριτικό token που τους επιτρέπει πρόσβαση στους πόρους του συστήματος.

Όταν ο χρήστης εγγράφεται για πρώτη φορά στην εφαρμογή μέσω της Register μεθόδου, το διακριτικό API δημιουργείται χρησιμοποιώντας τον αλγόριθμο κατακερματισμού sha-256 με το uniqueId ως πηγή του. Το διακριτικό API αποθηκεύεται στη στήλη api_token του πίνακα user που προστέθηκε.

Κάθε φορά που ο χρήστης αιτείται για έναν πόρο του συστήματος, η μέθοδος apiAuth.php ελέγχει αν το αίτημα περιλαμβάνει το διακριτικό api_token στο Header και το userId στην παράμετρο της post μεθόδου. Εάν το api_token του χρήστη δεν ταυτίζεται με το api_token στη βάση δεδομένων, το αίτημα ακυρώνεται. Στο Σχήμα 5-13 φαίνεται η ροή του Authentication.



Σχήμα 5-13. API Authentication flow.

Για την αύξηση του επιπέδου ασφαλείας ολόκληρου του συστήματος, πέραν της εφαρμογής, θα πρέπει να σχεδιαστεί και ένα σύστημα authorization και για το θερμοκήπιο. Η πρακτική αυτή, είναι βασική για την θωράκιση των IoT δικτύων [45].

5.3 Εφαρμογή Android

Σε αυτήν την ενότητα, παρουσιάζονται μερικά από τα βασικά στοιχεία της εφαρμογής που υλοποιήθηκε. Οι οθόνες και η λειτουργικότητα της εφαρμογής, παρουσιάζονται εκτενέστερα στην ενότητα 6.

Το όνομα της εφαρμογής είναι lekoApp, εμπνευσμένο από το όνομα του θερμοκηπίου που είναι Leko. Το Λογότυπο (logo) της εφαρμογής φαίνεται στην παρακάτω εικόνα.



Εικόνα 5-1. Λογότυπο (Logo) εφαρμογής.

Η εφαρμογή android, αποτελεί την διεπαφή του χρήστη με το σύστημα. Μέσω της εφαρμογής, ο χρήστης μπορεί, αφού πρώτα δημιουργήσει λογαριασμό με μοναδικό username και password, να συνδέεται σε αυτόν, να προσθέτει και να παρακολουθεί τα θερμοκήπια Leko που του ανήκουν.

Η σχεδίαση της εφαρμογής, έγινε με βάση τις απαιτήσεις συστήματος που έχουν τεθεί, σε συνεργασία με τον σχεδιασμό και τις προδιαγραφές του API, και με γνώμονα την εύκολη και ευχάριστη εμπειρία χρήσης. Η εφαρμογή, αποτελεί έναν από τους δύο clients του API, συνεπώς τα endpoints που ορίστηκαν στον **Πίνακας 5-5**, θα είναι και τα αιτήματα της εφαρμογής προς τον webserver.

5.3.1 Άδειες χρήσης εφαρμογής

Στο αρχείο android manifest.xml, ορίστηκαν όλες οι άδειες χρήσης υπηρεσιών της εφαρμογής. Αυτές είναι, η άδεια χρήσης του διαδικτύου (INTERNET), άδεια για την αποθήκευση (WRITE_EXTERNAL_STORAGE) και ανάκτηση (READ_EXTERNAL_STORAGE) δεδομένων σε directories που αφορούν την εφαρμογή, άδεια για την ανάκτηση της κατάστασης σύνδεσης του κινητού (READ_PHONE_STATE).

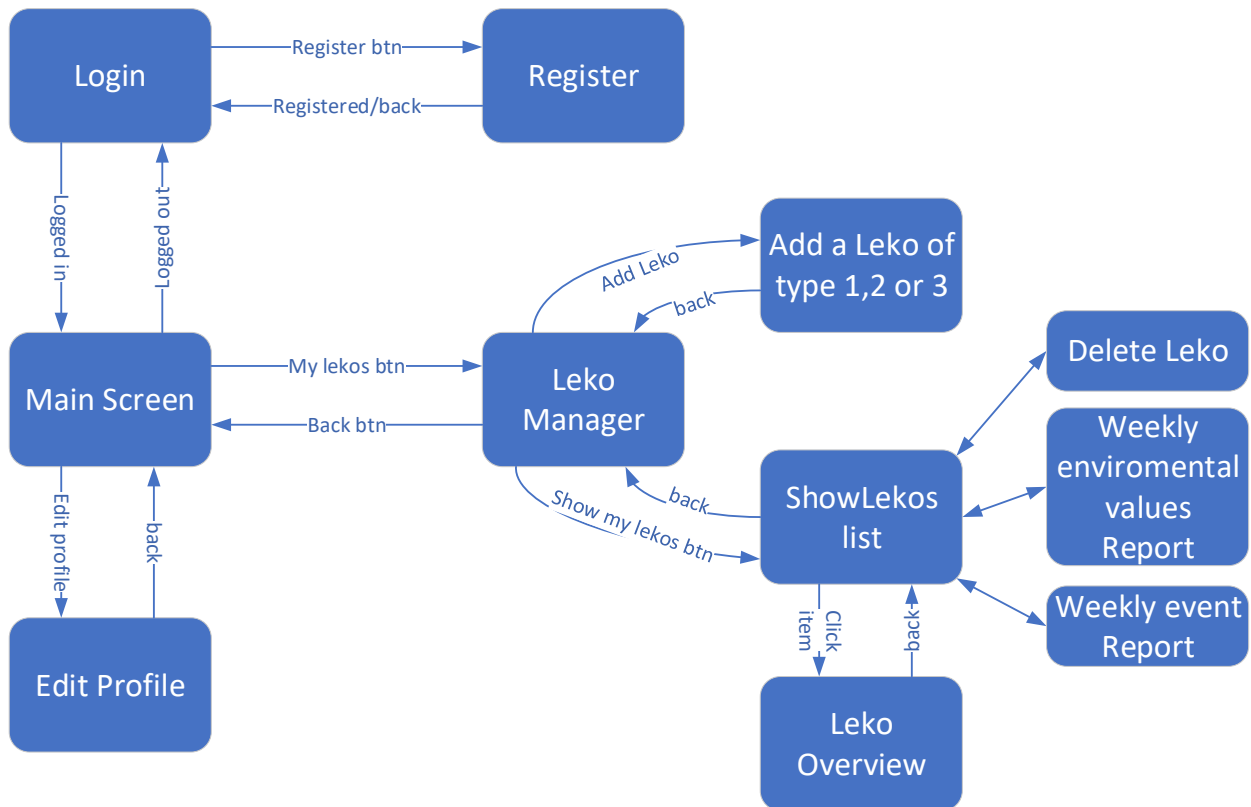
Στις επιπλέον υπηρεσίες, προστέθηκε αυτή του Firebase cloud messaging.

5.3.2 Εφαρμογή LekoApp

Η εφαρμογή LekoApp, αποτελείται από 8 δραστηριότητες, που ουσιαστικά αποτελούν τις οθόνες της. Οι οθόνες της εφαρμογής, η περιγραφή τους, η μεταβατικότητα τους και οι μέθοδοι του API που αφορούν, φαίνονται στον **Πίνακας 5-9**.

Πίνακας 5-9. Οθόνες εφαρμογής lekoApp			
Οθόνη	Περιγραφή οθόνης	Μεταβατικότητα σε άλλες οθόνες	API call δραστηριότητας
Login	Σύνδεση χρήστη στο σύστημα	Register-Main	login.php & addDevice.php
Register	Εγγραφή χρήστη στο σύστημα	Login	register.php
Main	Κεντρική οθόνη πλοήγησης χρήστη	Login- Leko manager -Edit profile	Κανένα. Ανάκτηση στοιχείων χρήστη από SQLite
Edit Profile	Τροποποίηση στοιχείων χρήστη	Main	changeUsernamePassword.php
Leko Manager	Οθόνη πλοήγησης Leko	ShowLekos- Add Leko	lekonum.php
Add Leko	Προσθήκη θερμοκηπίου Leko τύπου 1,2 ή 3	Leko Manager	lekoadd.php
ShowLekos	Προβολή συνδεδεμένων θερμοκηπίων Leko σε clickable λίστα. Δυνατότητα προβολής ιστορικού συγκεκριμένου θερμοκηπίου, ή διαγραφής	Leko Manager- Leko overview- Leko report- Leko event history	lekoreturn.php
Leko overview	Γενικές πληροφορίες θερμοκηπίου Leko που επιλέχθηκε	ShowLekos	Κανένα. Ανάκτηση στοιχείων θερμοκηπίου από SQLite
Leko report	Προβολή ιστορικού περιβαλλοντικών τιμών, τελευταίων 7 ημερών	ShowLekos	history.php
Leko event history	Προβολή δραστηριότητας ενεργοποιητών, τελευταίων 7 ημερών	ShowLekos	eventhistory.php

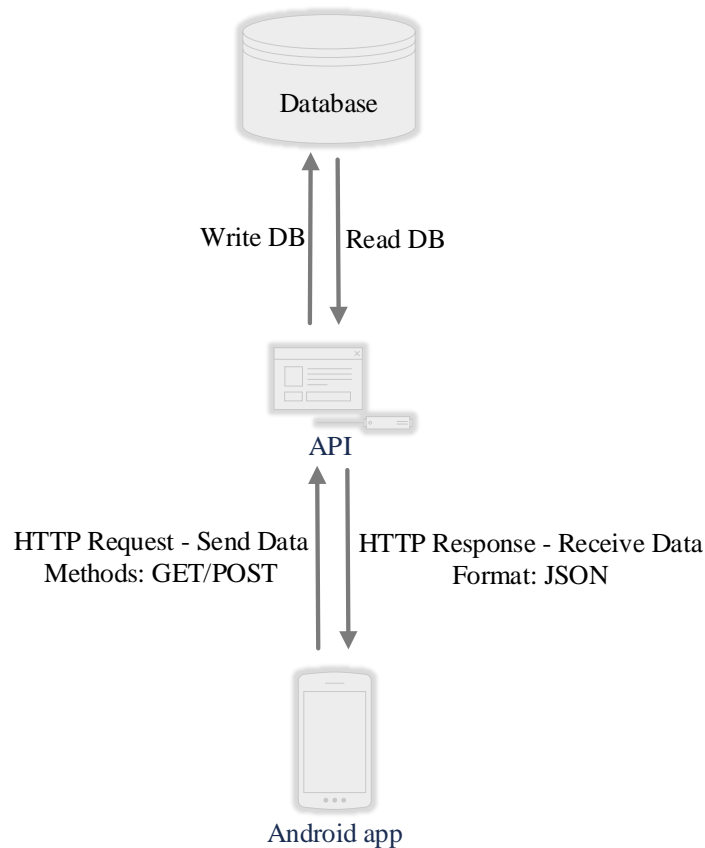
Στο Σχήμα 5-14 παρουσιάζεται το διάγραμμα ενεργειών του χρήστη μέσω της εφαρμογής, όπου διακρίνονται οι ενέργειες και οι μεταβάσεις μεταξύ των οθονών. Το διάγραμμα αυτό αποτέλεσε την αφετηρία της σχεδίασης της εφαρμογής. Οι οθόνες και οι αντίστοιχες δραστηριότητες της εφαρμογής, παρουσιάζονται στην ενότητα 6, μαζί με την επιβεβαίωση λειτουργίας ολόκληρου του συστήματος.



Σχήμα 5-14. Χάρτης ενεργειών εφαρμογής

5.3.3 Επικοινωνία εφαρμογής με Server

Η αρχιτεκτονική της επικοινωνίας μεταξύ εφαρμογής και server φαίνεται στο Σχήμα 5-15. Την επικοινωνία, διαχειρίζεται το API, που μεσολαβεί για την αλληλοεπίδραση με την βάση δεδομένων.



Σχήμα 5-15. Αρχιτεκτονική επικοινωνίας Android – Server.

Στην σχεδίαση της δικτύωσης της εφαρμογής LekoApp, επιλέχθηκε η βιβλιοθήκη Volley για όλα τα αιτήματα πλην των εβδομαδιαίων αναφορών, όπου επιλέχθηκε η Retrofit, λόγω καλύτερης απόκρισης. Η εφαρμογή στέλνει HTTP αιτήματα στο API, και οι απαντήσεις που δέχεται είναι σε μορφή JSON. Για την μετατροπή των JSON δεδομένων σε Java objects, χρησιμοποιήθηκε η βιβλιοθήκη GSON.

Για την επιτυχή επικοινωνία της εφαρμογής με το API, αποθηκεύεται στην εσωτερική μνήμη, το διακριτικό `api_token` που περιεγράφηκε στην υποενότητα **Error! Reference source not found.** και εξυπηρετεί την λειτουργία του api authentication, και εμπεριέχεται στον header κάθε αιτήματος που αποστέλλεται.

5.3.4 Γραφικές παραστάσεις εβδομαδιαίων αναφορών

Η γραφική απεικόνιση, είναι από τις πιο σημαντικές λειτουργίες της εφαρμογής. Ο χρήστης, όπως ορίστηκε και στις απαιτήσεις του συστήματος, θα πρέπει να μπορεί να διαχειρίζεται και να ενημερώνεται για το θερμοκήπιο του, με τον πιο απλό και φιλικό προς αυτόν τρόπο. Εκτιμάται λοιπόν, ότι η καλή απεικόνιση των report του θερμοκηπίου, συμβάλει σημαντικά στην αναβάθμιση του UX της διεπαφής του συστήματος. Εξυπηρετεί ταυτόχρονα, αισθητικούς αλλά και πρακτικούς σκοπούς.

Οι γραφικές παραστάσεις υλοποιήθηκαν με την χρήση της βιβλιοθήκης AnyChart. Οι γραφικές παραστάσεις είναι interactive με τον χρήστη. Ο χρήστης, μπορεί πατώντας σε κάποια από τις κορυφές της γραφικής παράστασης στον οριζόντιο άξονα των ημερών, να λάβει πιο στοχευμένες πληροφορίες. Οι δυνατότητες των γραφικών παραστάσεων, παρουσιάζονται πιο αναλυτικά στην ενότητα που ακολουθεί, μαζί με την επιβεβαίωση λειτουργίας.

5.3.5 Λήψη ειδοποιήσεων

Η συσκευή δέχεται ειδοποιήσεις, ακόμα και εάν η εφαρμογή είναι κλειστή, εάν ο χρήστης δεν έχει αποσυνδεθεί. Οι ειδοποιήσεις αποστέλλονται μέσω του Firebase API, όπως περιγράφηκε και στην υποενότητα 5.2.4. Για την αποστολή ειδοποιήσεων ενημερωτικού τύπου, σε εξατομικευμένες συσκευές, σχεδιάστηκε μία HTML διεπαφή, που παρουσιάζεται στην ενότητα που ακολουθεί.

6. Επιβεβαίωση λειτουργίας

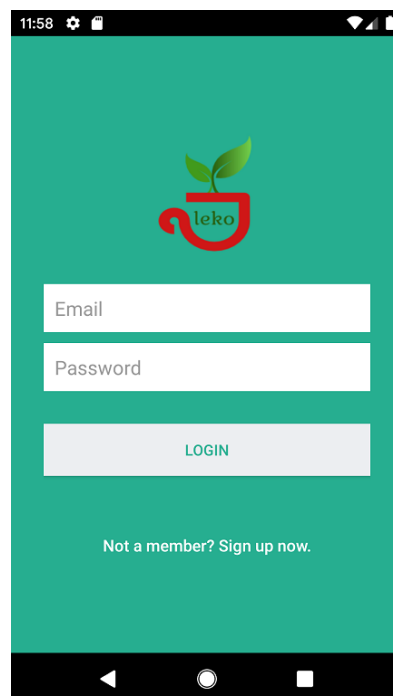
Στην παρούσα ενότητα, παρουσιάζεται ένα case study λειτουργίας του συστήματος διαχείρισης του θερμοκηπίου. Δίνεται μία εικόνα της εμπειρίας του χρήστη, καθώς και της λειτουργικότητας των διαδικασιών του συστήματος που παρουσιάστηκαν στις προηγούμενες ενότητες.

Για την προσομοίωση αυτή χρησιμοποιήθηκε ο emulator του android_studio και συγκεκριμένα η συσκευή nexus 5 με έκδοση android Q ή αλλιώς API level 29. Η βάση δεδομένων του server κατά την προσομοίωση είναι κενή, δηλαδή δεν διαθέτει κανέναν εγγεγραμμένο χρήστη ή θερμοκήπιο Leko.

Στην συνέχεια δημιουργείται ένας λογαριασμός χρήστη και προστίθεται ένα θερμοκήπιο. Κατόπιν, στέλνονται σε αυτό το θερμοκήπιο δείγματα διαστήματος μίας εβδομάδας, από εργαλείο σχεδιασμένο σε HTML που προσομοιώνει το θερμοκήπιο Leko. Τέλος, προστίθενται και άλλα θερμοκήπια στον λογαριασμό, και γίνεται έλεγχος του monitoring του συστήματος και της αποστολής των ειδοποιήσεων.

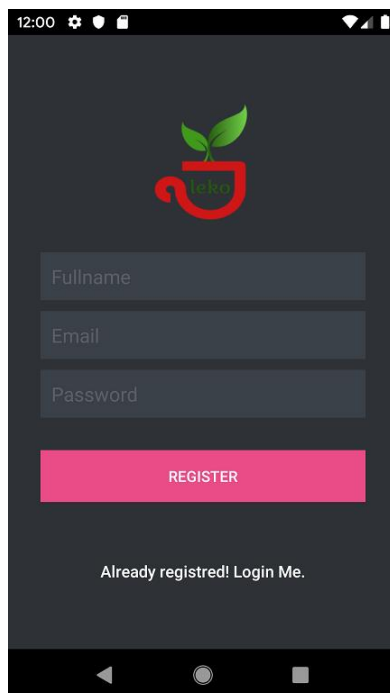
6.1 Εγγραφή χρήστη

Ο χρήστης κατά την εκκίνηση της εφαρμογής θα πρέπει να εγγραφεί στο σύστημα. Η αρχική οθόνη της εφαρμογής είναι η οθόνη εισόδου, όπως φαίνεται στην Εικόνα 6-1. Οθόνη Login



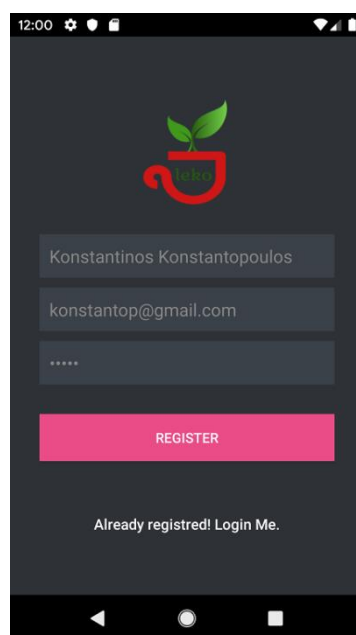
Εικόνα 6-1. Οθόνη Login

Εάν ο χρήστης δεν έχει προηγουμένως εγγραφεί στο σύστημα, θα πρέπει πρώτα να κάνει εγγραφή. Πατώντας το link text στο κάτω μέρος της οθόνης ο χρήστης μεταφέρεται στην οθόνη εγγραφής που φαίνεται παρακάτω.



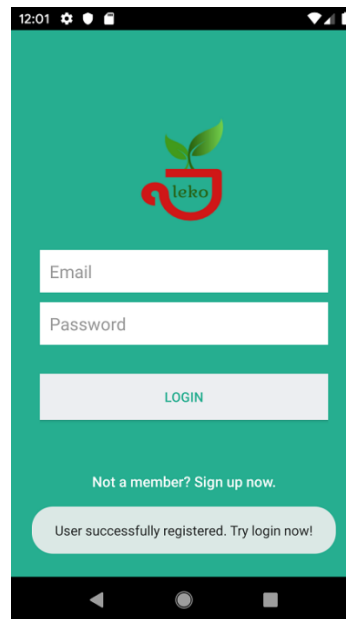
Εικόνα 6-2. Οθόνη εγγραφής

Η εγγραφή πραγματοποιείται με την συμπλήρωση του ονοματεπωνύμου του χρήστη, του email και του επιθυμητού κωδικού λογαριασμού που επιθυμεί. Στο παρόν παράδειγμα ο χρήστης είναι ο Κωνσταντίνος Κωνσταντόπουλος, με email konstantop@gmail.com και κωδικό '12345'.



Εικόνα 6-3. Στοιχεία εγγραφής χρήστη

Έπειτα από την επιτυχή εγγραφή του χρήστη, η εφαρμογή οδηγείται πάλι στην αρχική οθόνη. Αναδύεται σχετικό μήνυμα που προτρέπει τον χρήστη να κάνει login με τα στοιχεία που συμπλήρωσε κατά την εγγραφή (Εικόνα 6-4).



Εικόνα 6-4. Μήνυμα επιτυχημένης εγγραφής

Αφού ο χρήστης εγγραφεί επιτυχώς αποθηκεύεται και στην βάση δεδομένων του server στον πίνακα users του συστήματος, όπως φαίνεται στην παρακάτω εικόνα. Στα πεδία name και email αποθηκεύονται τα στοιχεία που συμπλήρωσε ο χρήστης. Τα υπόλοιπα πεδία παράγονται από το σύστημα και εξυπηρετούν την λειτουργικότητα του όπως αναλύθηκε και στις προηγούμενες ενότητες. Το πεδίο updated_at χρησιμοποιείται εάν ο χρήστης ανανεώσει κάποιο από τα στοιχεία του.

✓ Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

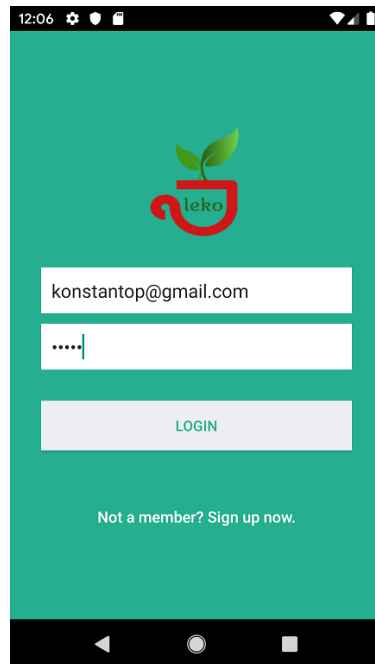
```
SELECT * FROM `users`
```

id	unique_id	name	email	encrypted_password	salt	created_at	updated_at
7	5d4e87d3ab8901.76694088	Konstantinos Konstantopoulos	konstantop@gmail.com	RRCLxUrMGUj8ok8W0z+Ggj1R0bthZjJjNGU1MzUy	af2c4e5352	2019-08-10 12:01:07	NULL

Εικόνα 6-5. Table users

6.2 Είσοδος χρήστη

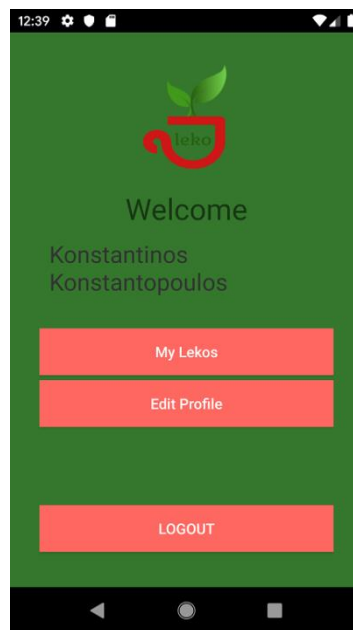
Στην συνέχεια ο χρήστης κάνει login στο σύστημα.



Εικόνα 6-6. Login εγγεγραμμένου χρήστη

Η είσοδος του χρήστη μπορεί να παρουσιάσει σφάλμα σε δύο περιπτώσεις. Εάν υπάρχει πρόβλημα στην σύνδεση είτε της εφαρμογής είτε του server, ή εάν τα στοιχεία του χρήστη είναι λανθασμένα. Και στις δύο περιπτώσεις η εφαρμογή ενημερώνει τον χρήστη με σχετικό pop up μήνυμα. Μετά την πετυχημένη είσοδο του ο χρήστης μεταφέρεται στην κεντρική οθόνη της εφαρμογής.

6.2.1 Κεντρική οθόνη IekoApp



Εικόνα 6-7. Κεντρική οθόνη εφαρμογής

Στην κεντρική οθόνη επισκόπησης ο χρήστης έχει τρεις επιλογές. Μπορεί να μεταφερθεί στον κέντρο ελέγχου των θερμοκηπίων Lekos πατώντας το κουμπί 'My Lekos', να επεξεργαστεί το προφίλ του πατώντας το κουμπί 'Edit Profile', ή να αποσυνδεθεί πατώντας το 'Logout'.

6.2.2 Αποθήκευση διακριτικού συσκευής

Κατά την είσοδο του χρήστη στο σύστημα με μία συσκευή, αποθηκεύεται στον server το διακριτικό εγγραφής της συσκευής αυτής. Το διακριτικό αποθηκεύεται στον πίνακα user_devices μαζί με το user_id του χρήστη που συνδέθηκε με αυτή.

✓ Showing rows 0 - 0 (1 total, Query took 0.0090 seconds.)

```
SELECT * FROM `user_devices`
```

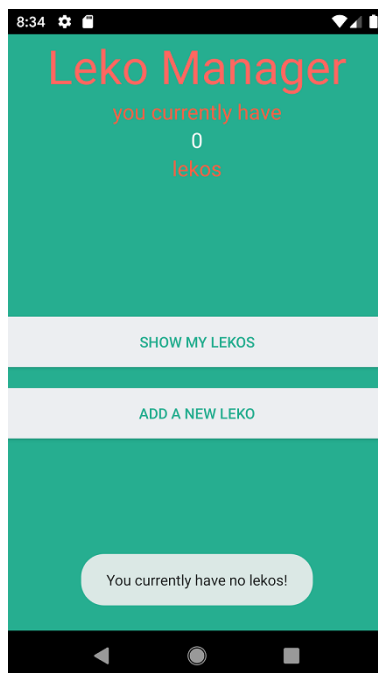
id	reg_id	user_id	created_at
63	daag9Rf1I-Q:APA91bFWKZKmEVJW_CLPAeZ3Qo6blw6Kxgwgqz...	7	2019-08-10 12:06:59

Εικόνα 6-8. Πίνακας user_devices

6.3 Διαχείριση θερμοκηπίων Leko

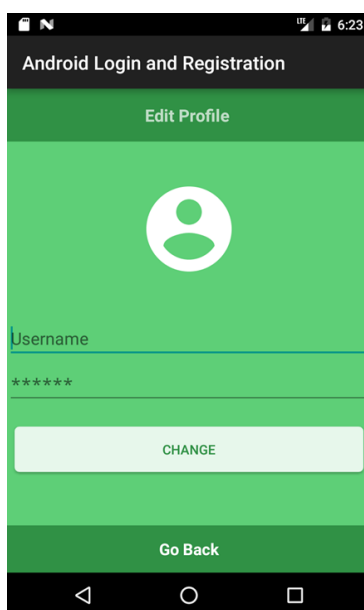
Στην οθόνη Leko Manager ο χρήστης μπορεί να διαχειριστεί τα Leko θερμοκήπιά του. Το μήνυμα στο πάνω μέρος της οθόνης επισημαίνει τον αριθμό των συνδεδεμένων θερμοκηπίων του χρήστη. Ο χρήστης στην οθόνη αυτή μπορεί να μεταφερθεί στα θερμοκήπια του ή να προσθέσει κάποιο ακόμα θερμοκήπιο.

Πατώντας το κουμπί ‘My Lekos’ ο χρήστης μεταφέρεται στο κέντρο ελέγχου των θερμοκηπίων, όπως φαίνεται στην Εικόνα 6-9. Ο χρήστης δεν έχει ακόμα προσθέσει κάποιο Leko στο σύστημα.



Εικόνα 6-9. Leko Manager

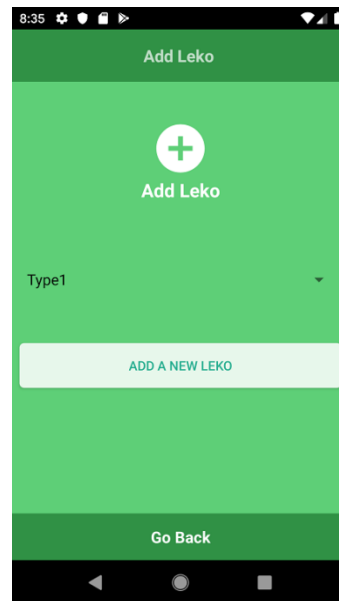
Στην επεξεργασία προφίλ ο χρήστης μπορεί να αλλάξει το υφιστάμενο username του καθώς και τον κωδικό του προφίλ του.



Εικόνα 6-10. Τροποποίηση στοιχείων προφίλ χρήστη

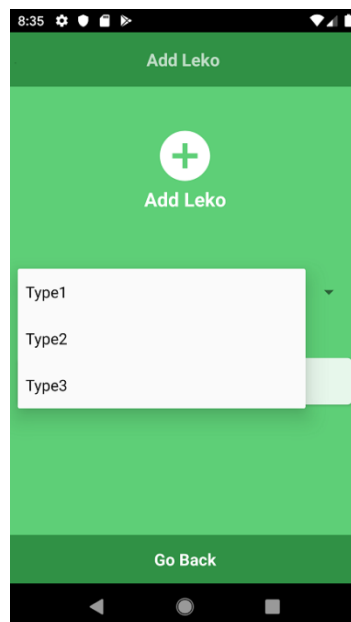
6.3.1 Προσθήκη θερμοκηπίου

Πατώντας το κουμπί ‘ADD NEW LEKO’ ο χρήστης μεταφέρεται στην οθόνη προσθήκης Leko.



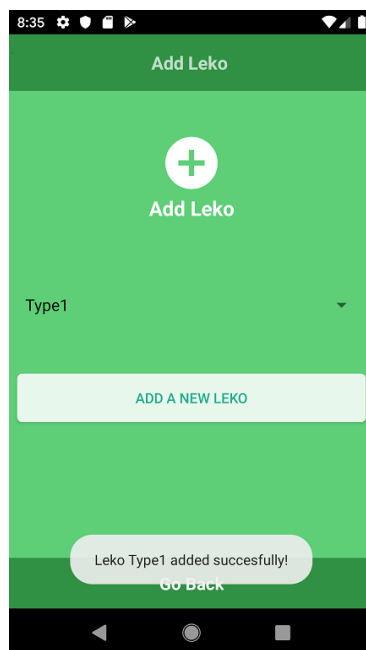
Εικόνα 6-11. Προσθήκη Leko

Ο χρήστης πρέπει να επιλέξει τον τύπο του θερμοκηπίου που θέλει να προσθέσει. Πατώντας στον τύπο εμφανίζεται η λίστα με τις επιλογές καλλιέργειας. Προεπιλογή του συστήματος είναι ο τύπος 1, δηλαδή τα ζαρζαβατικά.



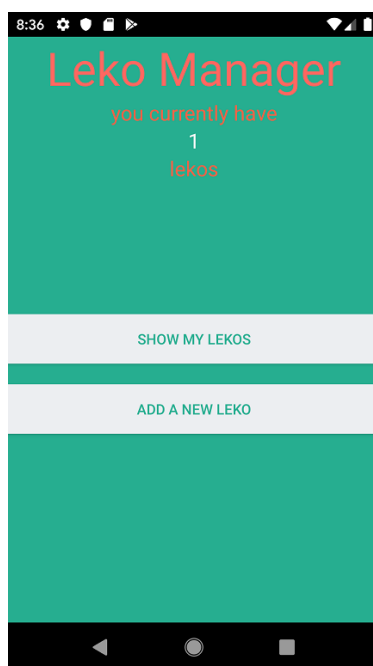
Εικόνα 6-12. Λίστα τύπων θερμοκηπίου

Ο χρήστης προσθέτει ένα θερμοκήπιο ζαρζαβατικών, τύπου 1. Εάν η προσθήκη είναι επιτυχής αναδύεται σχετικό μήνυμα, όπως φαίνεται στην Εικόνα 6-13. Σε διαφορετική περίπτωση ο χρήστης ενημερώνεται για την αποτυχία εισαγωγής.



Εικόνα 6-13. Προσθήκη θερμοκηπίου τύπου 1

Πίσω στην οθόνη ελέγχου φαίνεται ότι ο χρήστης διατηρεί πλέον ένα θερμοκήπιο στο συνδεδεμένο στο σύστημα.



Εικόνα 6-14. Κέντρο ελέγχου μετά την εισαγωγή

Το θερμοκήπιο που προστέθηκε φαίνεται και στην βάση δεδομένων του server στον πίνακα leko. Το θερμοκήπιο που δημιουργήθηκε έχει id, lekoId και το userId του χρήστη που το δημιούργησε. Όπως φαίνεται και στην Εικόνα 6-15 οι περιβαλλοντικές τιμές του θερμοκηπίου είναι κενές. Αυτό συμβαίνει γιατί το θερμοκήπιο έχει δημιουργηθεί στην βάση του server αλλά δεν έχει ακόμα συνδεθεί με το πραγματικό θερμοκήπιο του χρήστη.

✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT * FROM `leko`
```

id	lekold	userId	type	temp	humidity	light	created_at
49	5d5051e95d6315.29172982	7	1	NULL	NULL	NULL	2019-08-11 20:35:37

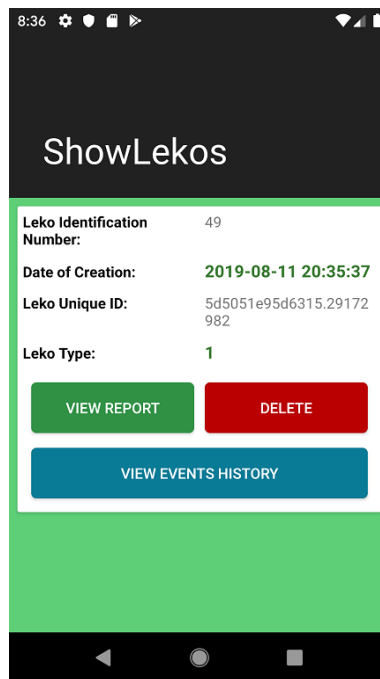
Εικόνα 6-15. Πίνακας Leko

6.3.2 Λίστα θερμοκηπίων

Για να λειτουργήσει το σύστημα, θα πρέπει το φυσικό θερμοκήπιο του χρήστη να συνδεθεί στο τοπικό δίκτυο και να επικοινωνήσει με τον κεντρικό server του συστήματος. Ο χρήστης θα πρέπει να εισάγει στο θερμοκήπιο του το lekoId που του δόθηκε από το σύστημα. Με τον τρόπο αυτό όταν το θερμοκήπιο επικοινωνήσει με τον server θα μπορεί να γίνει η ταυτοποίηση.

Αν ο χρήστης επιλέξει να δει τα θερμοκήπια του θα μεταφερθεί στην οθόνη showLekos, όπως φαίνεται και στην Εικόνα 6-16. Στην οθόνη αυτή παρουσιάζονται όλα τα θερμοκήπια του χρήστη σε μορφή λίστας, καθώς και βασικά χαρακτηριστικά που χαρακτηρίζουν το καθένα από αυτά. Τα χαρακτηριστικά αυτά είναι το Leko Id, η ημερομηνία και ώρα δημιουργίας, το Leko Unique Id και ο τύπος του φυτού που περιέχει. Σε κάθε ένα από τα θερμοκήπια, υπάρχει και κουμπί διαγραφής. Ο χρήστης εάν θέλει να χρησιμοποιήσει το θερμοκήπιο του για κάποια άλλη οικογένεια φυτών θα πρέπει να το διαγράψει και να το καταχωρήσει στο σύστημα ξανά. Αυτό έγινε για να αποφευχθεί η πιθανότητα αλλαγής κατά λάθος από την μεριά του χρήστη. Στην περίπτωση αυτή το σύστημα θα λειτουργούσε λάθος.

Ο χρήστης μπορεί να δει το LekoId του θερμοκηπίου του πατώντας το 'My Lekos'. Τα αναγνωριστικά id του θερμοκηπίου, όπως περιγράφηκε και στην υποενότητα 5.1.1 είναι δύο, για λόγους ασφαλείας. Το Leko Id και το Leko Unique ID. Στην παρακάτω εικόνα φαίνονται και τα δύο στον χρήστη. Στην πραγματικότητα ο χρήστης θα μπορεί να δει μόνο το Leko Unique ID, το οποίο και θα εισάγει στο θερμοκήπιο του.



Εικόνα 6-16. Οθόνη λίστας Lekos του χρήστη

6.4 Επικοινωνία με θερμοκήπιο Leko

Για να προσομοιωθεί το θερμοκήπιο Leko καθώς και ο τρόπος με τον οποίο αλληλοεπιδρά με το σύστημα διαχείρισης, δημιουργήθηκε μία σελίδα php/html. Μέσω της φόρμας της σελίδας στέλνονται αιτήματα στον server, όπως θα έστελνε και ένα φυσικό θερμοκήπιο Leko.

Σε κάθε ένα από τα αιτήματά αυτά, το θερμοκήπιο στέλνει στον server το unique id και τις περιβαλλοντικές του τιμές. Τα αιτήματα στέλνονται στον server κάθε ένα χρονικό διάστημα που έχει τεθεί (2 ώρες). Το θερμοκήπιο στέλνει επίσης αίτημα εάν κάποιος από τους ενεργοποιητές του ενεργοποιηθεί. Στην Εικόνα 6-17 φαίνεται το πρώτο αίτημα του θερμοκηπίου στον server με περιβαλλοντικές τιμές 32, 61 και 14 για θερμοκρασία, υγρασία εδάφους και χρόνο φωτισμού αντίστοιχα. Οι ενεργοποιητές είναι όλοι απενεργοποιημένοι.

Lekos

Leko

Leko list:
5d5051e95d6315.29172982 ▼

Leko Type:
1 ▼

Temp:
32 □

Hum:
61 □

Lighting:
14 □

Fan:
off ▼

Pump:
off ▼

Led:
off ▼

Send Request

Εικόνα 6-17. Προσομοίωση θερμοκηπίου Leko

Αφού το αίτημα ληφθεί από τον server του συστήματος, ενημερώνονται και οι νέες περιβαλλοντικές τιμές στην βάση δεδομένων.

✓ Showing rows 0 - 0 (1 total, Query took 0.0090 seconds.)

```
SELECT * FROM `leko`
```

id	lekold	userId	type	temp	humidity	light	created_at	updated_at
49	5d5051e95d6315.29172982	7	1	32	61	14	2019-08-11 20:35:37	2019-08-11 17:45:44

Εικόνα 6-18. Leko ανανέωση τιμών

Το αίτημα του θερμοκηπίου καταγράφεται επίσης στον πίνακα history της βάσης δεδομένων, καθώς και στον πίνακα events.

✓ Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT * FROM `history`
```

regID	lekotype	mDate	sTemp	sHumidity	sLight	lekold	mTime
45	1	2019-08-11	32	61	14	5d5051e95d6315.29172982	17:45:44

Εικόνα 6-19. Πίνακας history

✓ Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT * FROM `events`
```

regID	lekold	mDate	mTime	type	fan	led	pump
15	5d5051e95d6315.29172982	2019-08-11	17:45:44	1	off	off	off

Εικόνα 6-20. Πίνακας Events

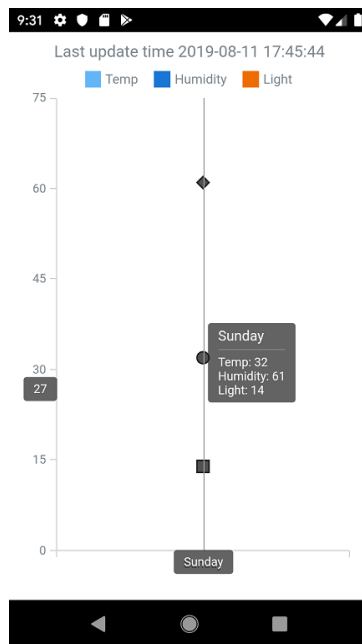
Τα στοιχεία στην λίστα των θερμοκηπίων (Εικόνα 6-16) είναι clickable. Πατώντας πάνω στο στοιχείο Leko της λίστας ο χρήστης μεταφέρεται στην οθόνη του Leko. Εκεί βρίσκονται συγκεντρωμένες οι πληροφορίες του θερμοκηπίου, καθώς και οι τελευταίες περιβαλλοντικές τιμές που ελήφθησαν.

Leko Info	
ID:	49
Lekold:	5d5051e95d6315.29172982
UserId:	7
Type:	1
Temp:	32
Humidity:	61
Light:	14
Created_at:	2019-08-11 20:35:37

Εικόνα 6-21. Οθόνη Leko

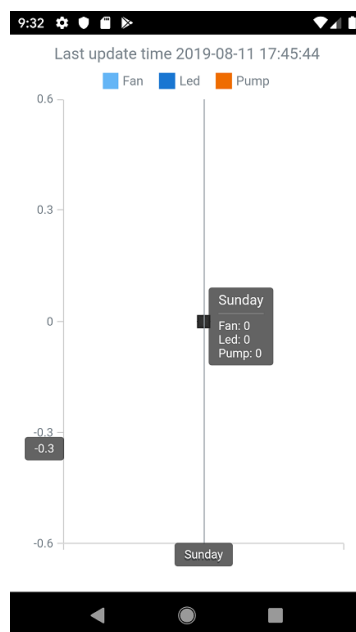
6.5 Εβδομαδιαίες αναφορές θερμοκηπίων Leko

Πατώντας το κουμπί 'View Report', ο χρήστης μπορεί να δει την εβδομαδιαία αναφορά του θερμοκηπίου του σε μορφή γραφική παράστασης. Στην γραφική παράσταση ο χρήστης μπορεί να δει την διακύμανση των περιβαλλοντικών τιμών σε συνάρτηση με το χρόνο. Στην Εικόνα 6-22 φαίνονται οι τελευταίες τιμές που έλαβε ο server. Δεν έχουν ληφθεί άλλα δείγματα αναφορών από το θερμοκήπιο προς τον server, για τον λόγο αυτό η γραφική παράσταση είναι ένα στιγμιότυπο.



Εικόνα 6-22. Αναφορά τιμών έπειτα από την πρώτη εγγραφή

Ομοίως, ο χρήστης πατώντας το κουμπί 'View Events History', μπορεί να δει την αναφορά των συμβάντων των ενεργοποιητών της περασμένης εβδομάδας. Στην παρακάτω εικόνα φαίνεται ότι στην μία εγγραφή που έχει ληφθεί δεν υπάρχει κάποιο συμβάν.



Εικόνα 6-23. Αναφορά ενεργοποιητών έπειτα από την πρώτη εγγραφή

Για την παρουσίαση των αναφορών δόθηκαν στον server δεκαεφτά δείγματα από το θερμοκήπιο σε διάστημα μίας εβδομάδας, από Σάββατο 24/8 έως Παρασκευή 30/8. Οι τιμές που δόθηκαν καθώς και η ημέρα και η ώρα που ελήφθησαν φαίνονται στον παρακάτω πίνακα. Όλες οι αναφορές οφείλονται σε κάποιο γεγονός, πότισμα, αερισμό ή φωτισμό.

Πίνακας 6-1. Δείγματα θερμοκηπίου					
Ημερομηνία δείγματος	Ωρα δείγματος	Θερμοκρασία (Celsius)	Υγρασία (%)	Φωτισμός (ώρες/ημέρα)	Γεγονός
Σάββατο 24/8	12:12	26	62	18	Ανεμιστήρας
Σάββατο 24/8	14:18	40	61	20	Ανεμιστήρας
Σάββατο 24/8	18:02	38	58	18	Ανεμιστήρας
Κυριακή 25/8	11:32	40	56	22	Ανεμιστήρας
Κυριακή 25/8	18:46	33	55	21	Led
Δευτέρα 26/8	10:41	34	59	19	Αντλία
Δευτέρα 26/8	13:08	35	65	17	Led
Δευτέρα 26/8	16:39	39	64	19	Ανεμιστήρας
Δευτέρα 26/8	19:05	36	57	17	Led
Τρίτη 27/8	14:15	33	58	18	Αντλία
Τρίτη 27/8	18:37	30	56	14	Αντλία
Τετάρτη 28/8	10:34	35	63	16	Ανεμιστήρας
Τετάρτη 28/8	13:52	32	62	17	Led
Τετάρτη 28/8	16:05	39	59	14	Αντλία
Πέμπτη 29/8	09:09	38	62	18	Αντλία
Πέμπτη 29/8	13:19	40	64	16	Led
Παρασκευή 30/8	09:02	34	55	20	Ανεμιστήρας

Στην Εικόνα 6-24. Πίνακας history και στην Εικόνα 6-25. Πίνακας Events, φαίνονται και οι εγγραφές στην βάση δεδομένων για τις παραπάνω αναφορές, στον πίνακα history και events αντίστοιχα.

✓ Showing rows 0 - 16 (17 total, Query took 0.0005 seconds.)

`SELECT * FROM `history``

regID	lekotype	mDate	sTemp	sHumidity	sLight	lekold	mTime
46	1	2019-08-24	26	62	18	5d5051e95d6315.29172982	12:12:03
47	1	2019-08-24	40	61	20	5d5051e95d6315.29172982	14:18:03
48	1	2019-08-24	38	58	18	5d5051e95d6315.29172982	18:02:03
49	1	2019-08-25	40	56	22	5d5051e95d6315.29172982	11:32:04
50	1	2019-08-25	33	55	21	5d5051e95d6315.29172982	18:46:43
51	1	2019-08-26	34	59	19	5d5051e95d6315.29172982	10:41:17
52	1	2019-08-26	35	65	17	5d5051e95d6315.29172982	13:08:41
53	1	2019-08-26	39	64	19	5d5051e95d6315.29172982	16:39:26
54	1	2019-08-26	36	57	17	5d5051e95d6315.29172982	19:05:53
55	1	2019-08-27	33	58	18	5d5051e95d6315.29172982	14:15:23
56	1	2019-08-27	30	59	14	5d5051e95d6315.29172982	18:37:08
57	1	2019-08-28	35	63	16	5d5051e95d6315.29172982	10:34:01
58	1	2019-08-28	32	62	17	5d5051e95d6315.29172982	13:52:09
59	1	2019-08-28	39	59	14	5d5051e95d6315.29172982	16:05:28
60	1	2019-08-29	38	62	18	5d5051e95d6315.29172982	09:09:56
61	1	2019-08-29	40	64	16	5d5051e95d6315.29172982	13:19:14
62	1	2019-08-30	34	55	20	5d5051e95d6315.29172982	09:02:19

Εικόνα 6-24. Πίνακας history

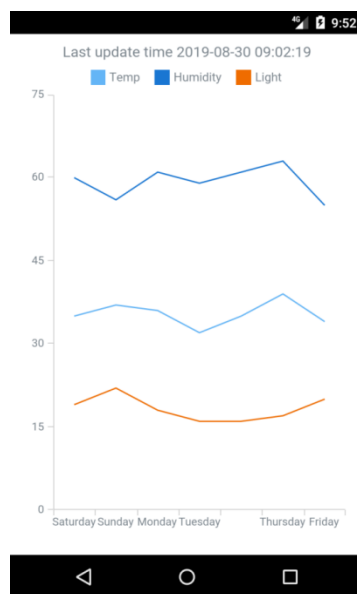
✓ Showing rows 0 - 16 (17 total, Query took 0.0004 seconds.)

`SELECT * FROM `events``

regID	lekold	mDate	mTime	type	fan	led	pump
16	5d5051e95d6315.29172982	2019-08-24	12:12:03	1	on	off	off
17	5d5051e95d6315.29172982	2019-08-24	14:18:03	1	on	off	off
18	5d5051e95d6315.29172982	2019-08-24	18:02:03	1	on	off	off
19	5d5051e95d6315.29172982	2019-08-25	11:32:04	1	on	off	off
20	5d5051e95d6315.29172982	2019-08-25	18:46:43	1	off	on	off
21	5d5051e95d6315.29172982	2019-08-26	10:41:17	1	off	off	on
22	5d5051e95d6315.29172982	2019-08-26	13:08:41	1	off	on	off
23	5d5051e95d6315.29172982	2019-08-26	16:39:26	1	on	off	off
24	5d5051e95d6315.29172982	2019-08-26	19:05:53	1	off	on	off
25	5d5051e95d6315.29172982	2019-08-27	14:15:23	1	off	off	on
26	5d5051e95d6315.29172982	2019-08-27	18:37:08	1	off	off	on
27	5d5051e95d6315.29172982	2019-08-28	10:34:01	1	on	off	off
28	5d5051e95d6315.29172982	2019-08-28	13:52:09	1	off	on	off
29	5d5051e95d6315.29172982	2019-08-28	16:05:28	1	off	off	on
30	5d5051e95d6315.29172982	2019-08-29	09:09:56	1	off	off	on
31	5d5051e95d6315.29172982	2019-08-29	13:19:14	1	off	on	off
32	5d5051e95d6315.29172982	2019-08-30	09:02:19	1	on	off	off

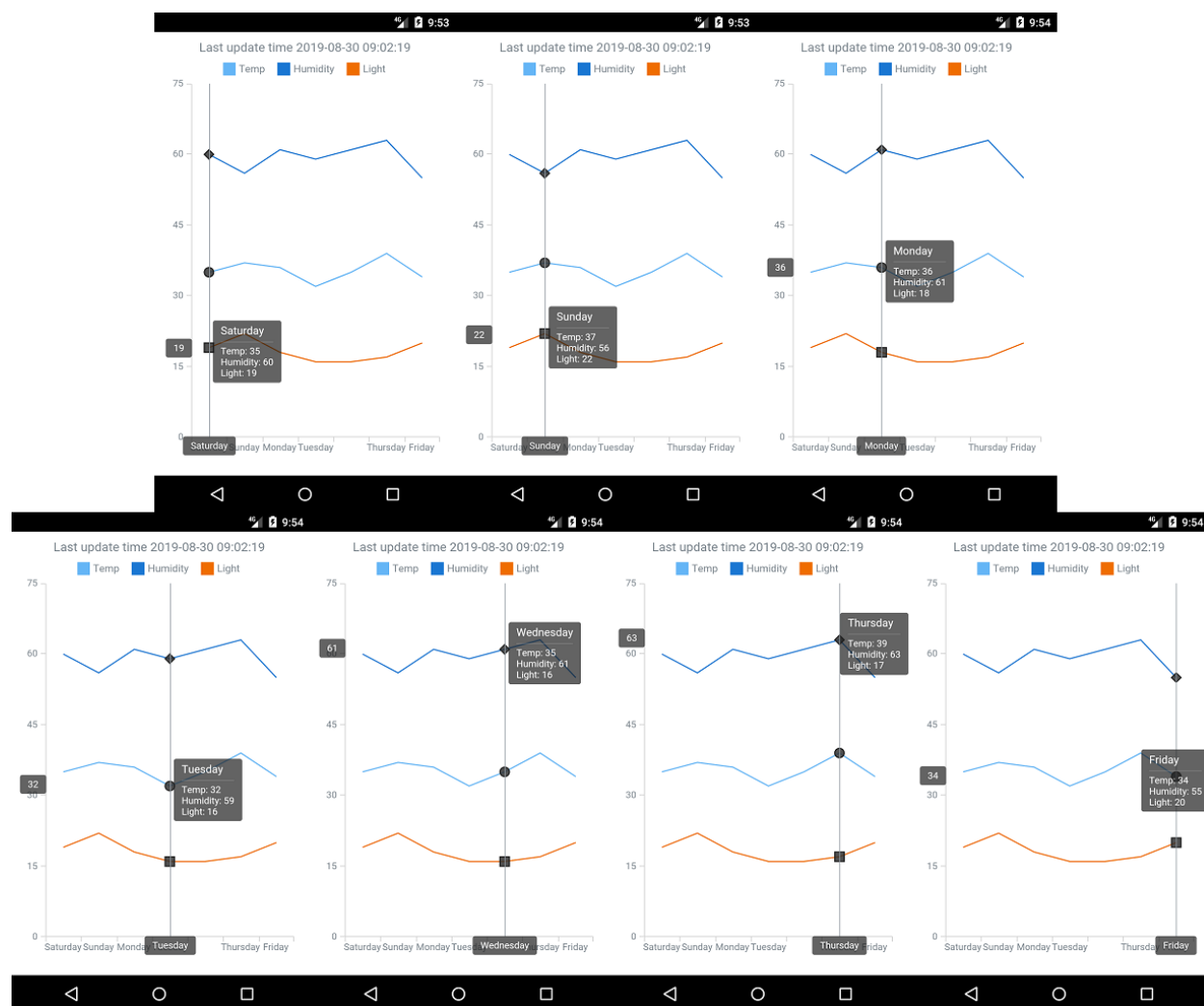
Εικόνα 6-25. Πίνακας Events

6.5.1 Γραφική παράσταση περιβαλλοντικών τιμών



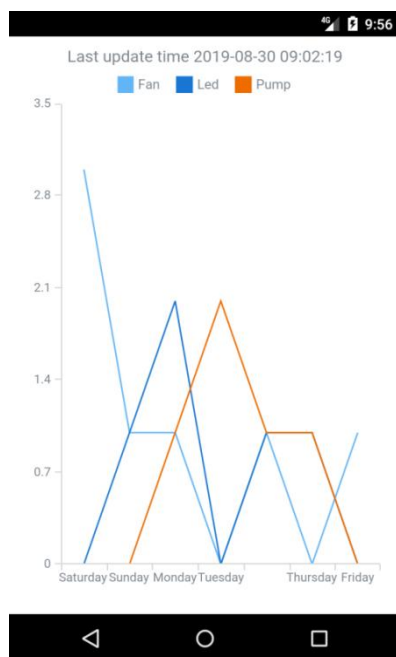
Εικόνα 6-26. Γραφική παράσταση περιβαλλοντικών τιμών εβδομάδας

Η γραφική παράσταση αναφέρεται μόνο στις τιμές της προηγούμενης εβδομάδας. Οποιαδήποτε παλαιότερη εγγραφή δεν φαίνεται στην γραφική παράσταση. Στην εικόνα 23 φαίνεται η γραφική παράσταση της εβδομαδιαίας αναφοράς, για τα δείγματα που δόθηκαν. Με γαλάζιο φαίνεται η θερμοκρασία, με μπλε η εδαφική υγρασία και με πορτοκαλί το επίπεδο φωτισμού. Πατώντας πάνω σε μία από τις ημέρες, ο χρήστης μπορεί να δει τις μέσες τιμές των δειγμάτων που ελήφθησαν, για κάθε ημέρα ξεχωριστά, όπως φαίνεται στην Εικόνα 6-27.



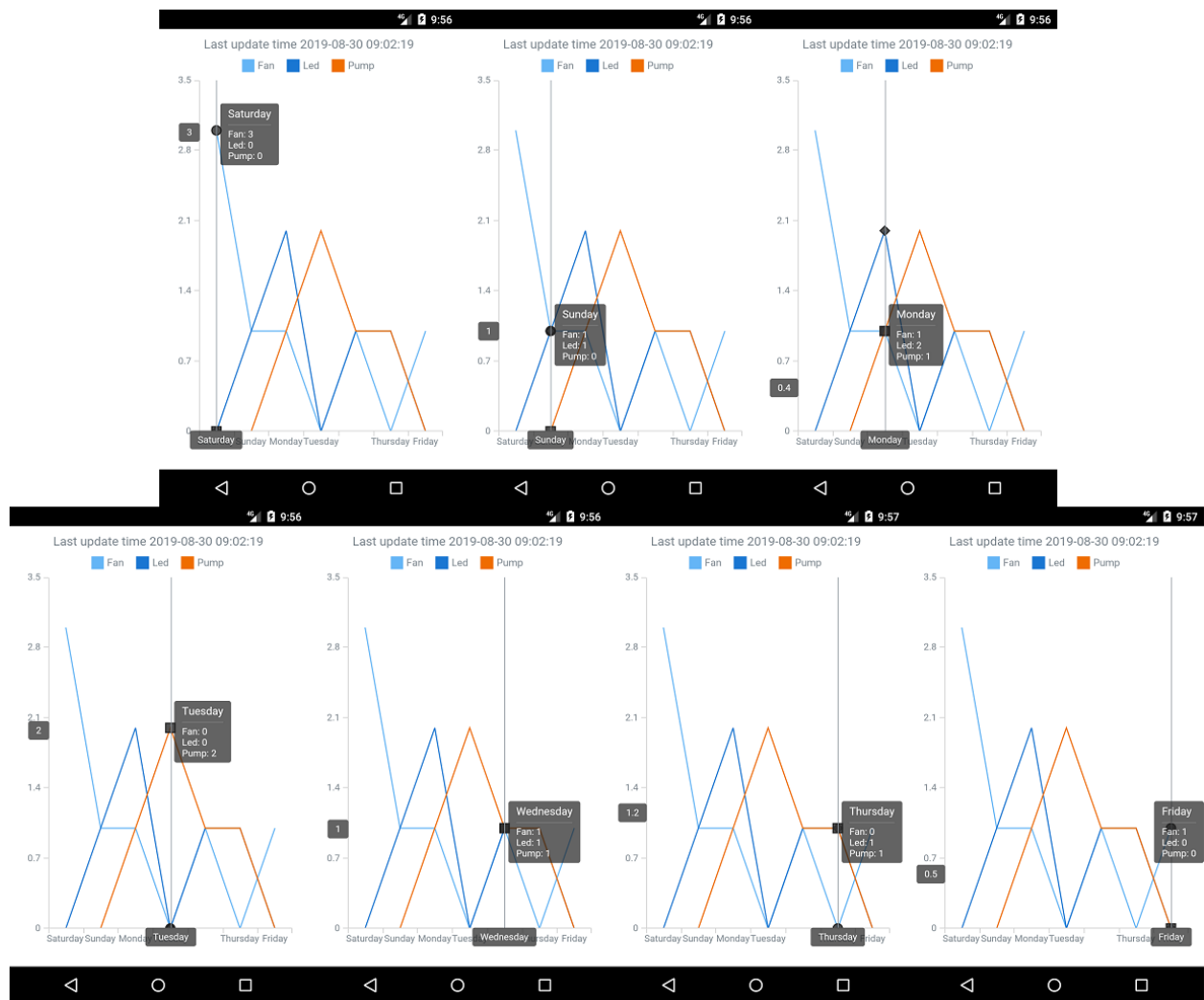
Εικόνα 6-27. Ημερήσιες αναφορές εβδομάδας

6.5.2 Γραφική παράσταση συμβάντων



Εικόνα 6-28. Γραφική παράσταση συμβάντων εβδομάδας

Στα συμβάντα της εβδομάδας δίνεται μία αναφορά των ενεργοποιητών στον χρήστη. Με γαλάζιο φαίνεται ο ανεμιστήρας, με μπλε τα φώτα Led και με πορτοκαλί η αντλία. Πατώντας πάνω σε μία από τις ημέρες, ο χρήστης μπορεί να δει τον συνολικό αριθμό των συμβάντων ανά κατηγορία, για κάθε ημέρα ξεχωριστά, όπως φαίνεται στην Εικόνα 6-29.

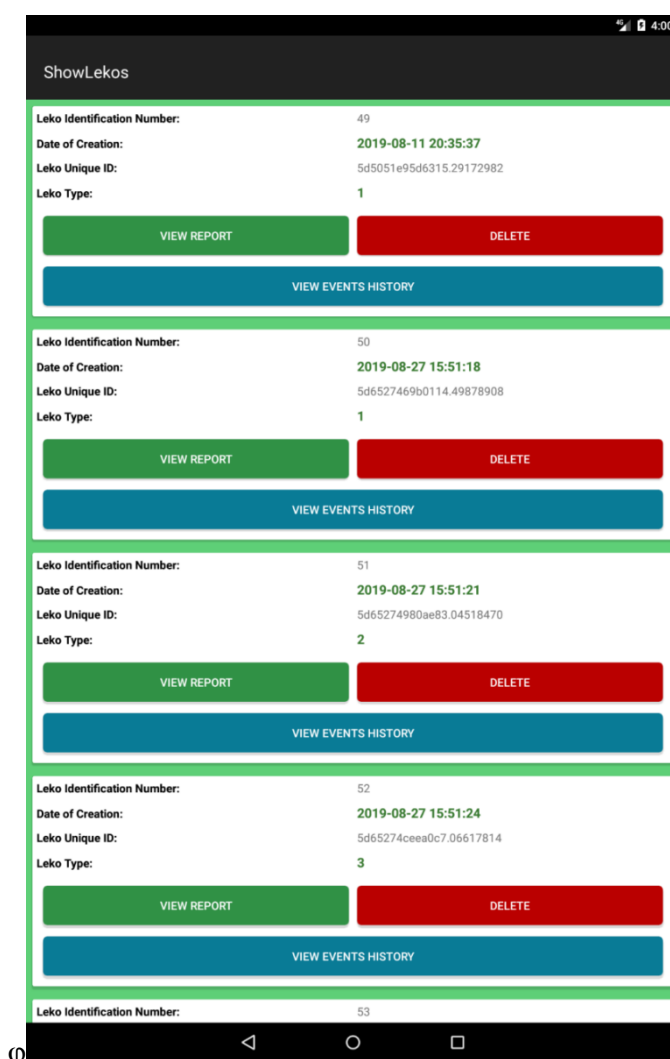


Εικόνα 6-29. Ημερήσια αναφορά συμβάντων εβδομάδας

6.6 Ειδοποιήσεις χρήστη

Για την επαλήθευση των ειδοποιήσεων, προστέθηκαν επιπλέον 4 θερμοκήπια, όπως φαίνεται και στην Εικόνα 6-30 στον λογαριασμό του χρήστη konstantop@gmail.com. Τα θερμοκήπια του χρήστη, με τις αντίστοιχες τελευταίες περιβαλλοντικές τιμές που ελήφθησαν φαίνονται στον παρακάτω πίνακα. Με κόκκινο απεικονίζονται οι τιμές που είναι εκτός των επιτρεπτών ορίων για τον ανάλογο τύπο καλλιέργειας.

Πίνακας 6-2. Θερμοκήπια χρήστη				
Id θερμοκηπίου	Είδος θερμοκηπίου	Θερμοκρασία (Celsius)	Υγρασία (%)	Φωτισμός (ώρες/ημέρα)
49	Ζαρζαβατικά (1)	34	55	20
50	Ζαρζαβατικά (1)	27	61	21
51	Παχύφυτα (2)	46	3	9
52	Καλλωπιστικά (3)	29	31	4
53	Παχύφυτα (2)	48	14	13



Εικόνα 6-30. Λίστα θερμοκηπίων που προστέθηκαν

6.6.1 Ειδοποιήσεις εσφαλμένων περιβαλλοντικών τιμών

Για τον έλεγχο των περιβαλλοντικών τιμών από τον διαχειριστή, εκτελείται η ρουτίνα 'checkLekos'. Ο έλεγχος γίνεται στις τωρινές περιβαλλοντικές τιμές του κάθε θερμοκηπίου. Η επιστροφή της εκτέλεσης της φαίνεται στην Εικόνα 6-31.

```

=====Sending Notification=====
C:\xampp64\www\android_login_api\Notifications.php:52: int 50

Leko id : 50

Sending notification to : dQ0o6eov3sk:APA91bEskKvG:GZMWmbkSLVpLNVbqMTGQF9K801fAUkGqSjkeJSS-GoYpsyu6vBtR69Hmlf.ra13YTNVRXdZC_5cPw3twu04cEdhgexp62zNPgcapRIYtbvNLn54heQNvil-pLaaA

Json sent:
C:\xampp64\www\android_login_api\Notifications.php:74:
array (size=1)
  'data' =>
    array (size=7)
      'title' => string 'Leko Notification' (length=17)
      'is_background' => boolean false
      'message' => string 'Temperature of Leko : 50 is not right, please check sensor Or fan!' (length=66)
      'image' => string '' (length=0)
      'payload' =>
        array (size=2)
          'team' => string 'Leko' (length=4)
          'score' => string '1' (length=1)
      'timestamp' => string '2019-08-28 13:07:35' (length=19)
      'lekoId' => int 50

Json Response: [{"multicast_id":4728957362342821870,"success":1,"failure":0,"canonical_ids":0,"results":[{"message_id":"0:1566997655199804%8bb492f6f9fd7ecd"}]}]

=====Sending Notification=====
C:\xampp64\www\android_login_api\Notifications.php:52: int 51

Leko id : 51

Sending notification to : dQ0o6eov3sk:APA91bEskKvG:GZMWmbkSLVpLNVbqMTGQF9K801fAUkGqSjkeJSS-GoYpsyu6vBtR69Hmlf.ra13YTNVRXdZC_5cPw3twu04cEdhgexp62zNPgcapRIYtbvNLn54heQNvil-pLaaA

Json sent:
C:\xampp64\www\android_login_api\Notifications.php:74:
array (size=1)
  'data' =>
    array (size=7)
      'title' => string 'Leko Notification' (length=17)
      'is_background' => boolean false
      'message' => string 'Check humidity sensor Or water pump of Leko :51' (length=47)
      'image' => string '' (length=0)
      'payload' =>
        array (size=2)
          'team' => string 'Leko' (length=4)
          'score' => string '1' (length=1)
      'timestamp' => string '2019-08-28 13:07:35' (length=19)
      'lekoId' => int 51

Json Response: [{"multicast_id":489787761456702055,"success":1,"failure":0,"canonical_ids":0,"results":[{"message_id":"0:1566997655487431%8bb492f6f9fd7ecd"}]}]

=====Sending Notification=====
C:\xampp64\www\android_login_api\Notifications.php:52: int 52

Leko id : 52

Sending notification to : dQ0o6eov3sk:APA91bEskKvG:GZMWmbkSLVpLNVbqMTGQF9K801fAUkGqSjkeJSS-GoYpsyu6vBtR69Hmlf.ra13YTNVRXdZC_5cPw3twu04cEdhgexp62zNPgcapRIYtbvNLn54heQNvil-pLaaA

Json sent:
C:\xampp64\www\android_login_api\Notifications.php:74:
array (size=1)
  'data' =>
    array (size=7)
      'title' => string 'Leko Notification' (length=17)
      'is_background' => boolean false
      'message' => string 'Temperature of Leko : 52 is not right, please check sensor Or fan!' (length=66)
      'image' => string '' (length=0)
      'payload' =>
        array (size=2)
          'team' => string 'Leko' (length=4)
          'score' => string '1' (length=1)
      'timestamp' => string '2019-08-28 13:07:35' (length=19)
      'lekoId' => int 52

Json Response: [{"multicast_id":6122869110532793945,"success":1,"failure":0,"canonical_ids":0,"results":[{"message_id":"0:1566997655779667%8bb492f6f9fd7ecd"}]}]

=====Sending Notification=====
C:\xampp64\www\android_login_api\Notifications.php:52: int 52

Leko id : 52

Sending notification to : dQ0o6eov3sk:APA91bEskKvG:GZMWmbkSLVpLNVbqMTGQF9K801fAUkGqSjkeJSS-GoYpsyu6vBtR69Hmlf.ra13YTNVRXdZC_5cPw3twu04cEdhgexp62zNPgcapRIYtbvNLn54heQNvil-pLaaA

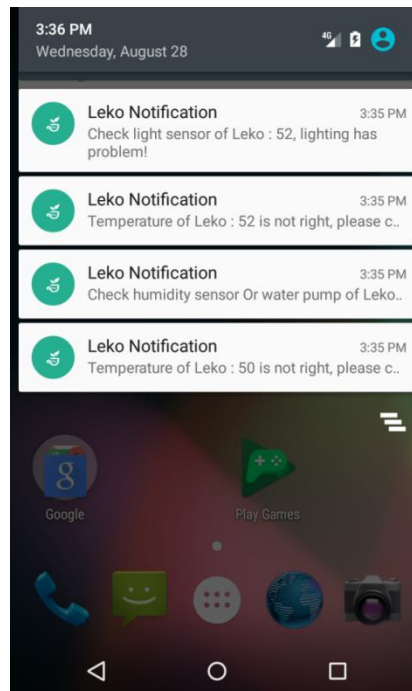
Json sent:
C:\xampp64\www\android_login_api\Notifications.php:74:
array (size=1)
  'data' =>
    array (size=7)
      'title' => string 'Leko Notification' (length=17)
      'is_background' => boolean false
      'message' => string 'Check light sensor of Leko :52, lighting has problem!' (length=53)
      'image' => string '' (length=0)
      'payload' =>
        array (size=2)
          'team' => string 'Leko' (length=4)
          'score' => string '1' (length=1)
      'timestamp' => string '2019-08-28 13:07:36' (length=19)
      'lekoId' => int 52

Json Response: [{"multicast_id":4659642627530461060,"success":1,"failure":0,"canonical_ids":0,"results":[{"message_id":"0:1566997656118783%8bb492f6f9fd7ecd"}]}]

```

Εικόνα 6-31. Εκτέλεση ρουτίνας checkLekos

Για να λάβει ο χρήστης τις ειδοποιήσεις στο κινητό του, αρκεί να μην έχει κάνει logout από τον λογαριασμό του στην συσκευή που χρησιμοποιεί. Η εφαρμογή μπορεί να είναι ανοιχτή, στο παρασκήνιο ή ακόμα και κλειστή με χρήση kill.



Εικόνα 6-32. Notifications εσφαλμένων τιμών

6.6.2 Ειδοποιήσεις σε πολλαπλές συσκευές

Εάν ο χρήστης χρησιμοποιεί την εφαρμογή IekoApp σε παραπάνω από μία συσκευές, οι ειδοποιήσεις στέλνονται σε όλες. Η εφαρμογή εγκαταστάθηκε σε μία ακόμα συσκευή (nexus 4, api level 22), μέσω της οποίας συνδέθηκε ο χρήστης. Στην Εικόνα 6-33 φαίνονται τα διακριτικά των δύο ενεργών συσκευών του χρήστη με user_id 7.

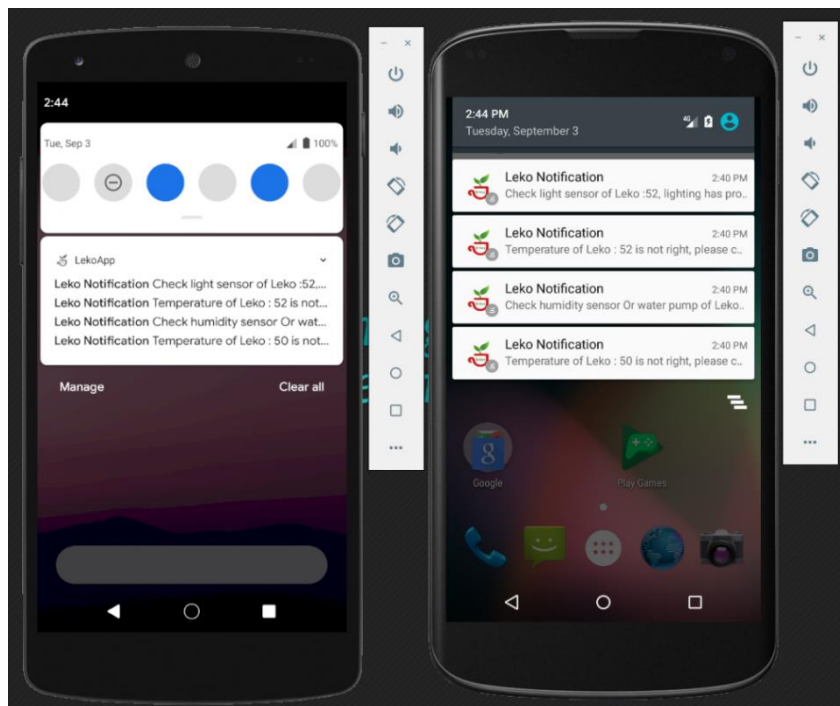
✓ Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)

```
SELECT * FROM `user_devices`
```

id	reg_id	user_id	created_at
83	fD3a_Wq1dHI:APA91bGrtGORjuT6EqmRZ0D-9cR6hXwl6dLrcw...	7	2019-09-03 13:21:58
85	daag9Rf1I-Q:APA91bFWKZKmEVJW_CLPAeZ3Qo6blw6Kxgwgqz...	7	2019-09-03 14:37:34

Εικόνα 6-33. Διακριτικά εγγραφής συσκευών στον πίνακα user_devices

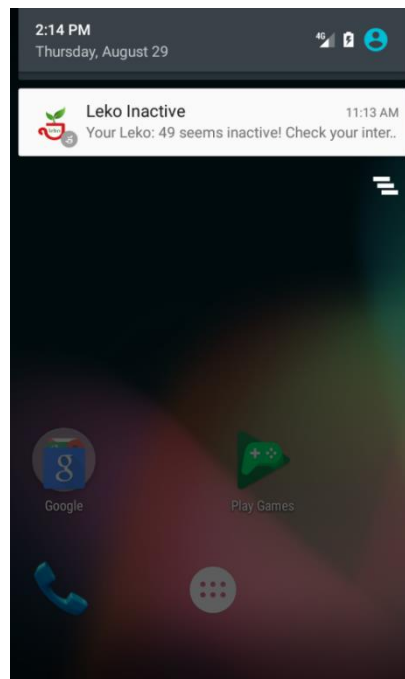
Στην Εικόνα 6-34 φαίνεται ότι οι ειδοποιήσεις λαμβάνονται ταυτόχρονα και από τις δύο αυτές ενεργές συσκευές.



Εικόνα 6-34. Ταυτόχρονη αποστολή ειδοποίησης στις ενεργές συσκευές του χρήστη

6.6.3 Ειδοποιήσεις αδράνειας

Για να διασφαλιστεί η ομαλή λειτουργία του θερμοκηπίου, η ρουτίνα `checkInactive` ενημερώνει τον χρήστη για την αδράνεια του θερμοκηπίου του. Αδράνεια, φαινομενικά σημαίνει ένα μεγάλο χρονικό διάστημα κατά το οποίο ένα ενεργό θερμοκήπιο δεν έχει ανανεώσει τις περιβαλλοντικές του τιμές. Το διάστημα αυτό έχει οριστεί στις 24 ώρες και άνω, αλλά μπορεί να τροποποιηθεί σε λιγότερο ή περισσότερο. Για να υλοποιηθεί ο έλεγχος αδράνειας, προστέθηκε ένα επιπλέον πεδίο `updated_at`, στον πίνακα `Leko` της βάσης δεδομένων του συστήματος. Το πεδίο `updated_at` ανανεώνεται αυτόματα όταν γίνεται κάποια αλλαγή στην αντίστοιχη εγγραφή.



Εικόνα 6-35. Ειδοποίηση χρήστη για αδράνεια θερμοκηπίου

6.6.4 Ειδοποιήσεις σε συγκεκριμένο θερμοκήπιο από τον browser

Μέσω της lekoNotify από το σύστημα διαχείρισης των θερμοκηπίων, μπορεί να σταλεί οποιαδήποτε ειδοποίηση σε συγκεκριμένο Leko. Με παρόμοιο τρόπο είναι εφικτό να σταλούν ειδοποιήσεις με βάση τον χρήστη ή ομάδα χρηστών, τύπο καλλιέργειας κλπ. Αυτή η δυνατότητα που προσφέρει το σύστημα δίνει την δυνατότητα στους διαχειριστές να ενημερώσουν την βάση χρηστών για οποιαδήποτε πληροφορία αφορά τα θερμοκήπιά τους, τον τύπο καλλιέργειάς τους, συμβουλές καλλιέργειας, και ανανεώσεις συστήματος.

Send to Single Device

Leko Id
Enter Leko id

Title
Enter title

Message
Notification message!

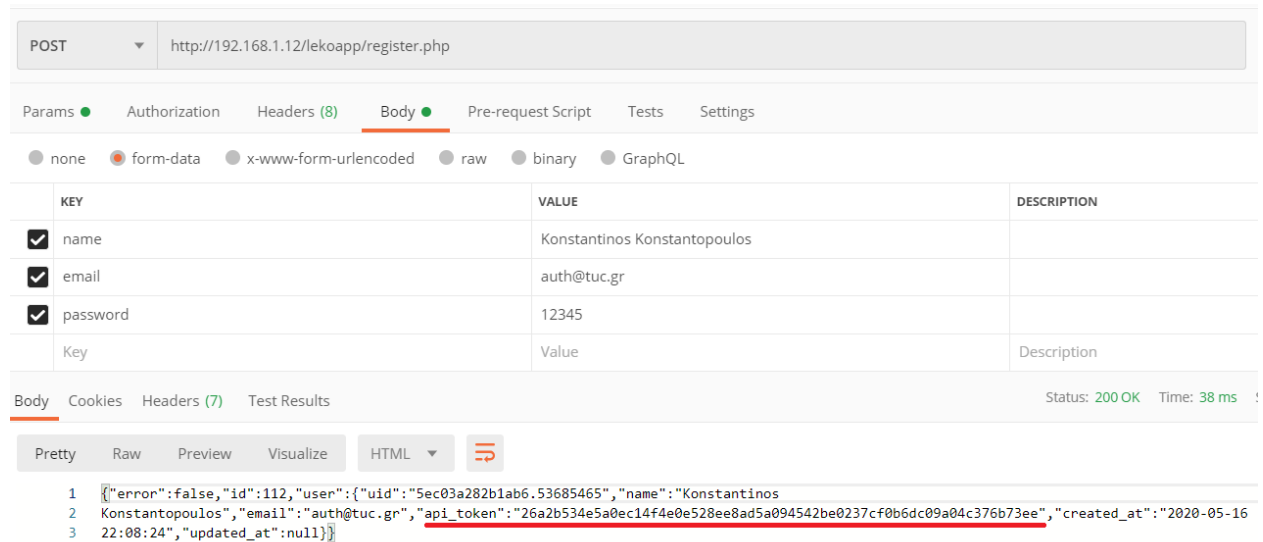
☐ Include image

Send

Εικόνα 6-36. Αποστολή ειδοποίησης για συγκεκριμένο Leko του συστήματος

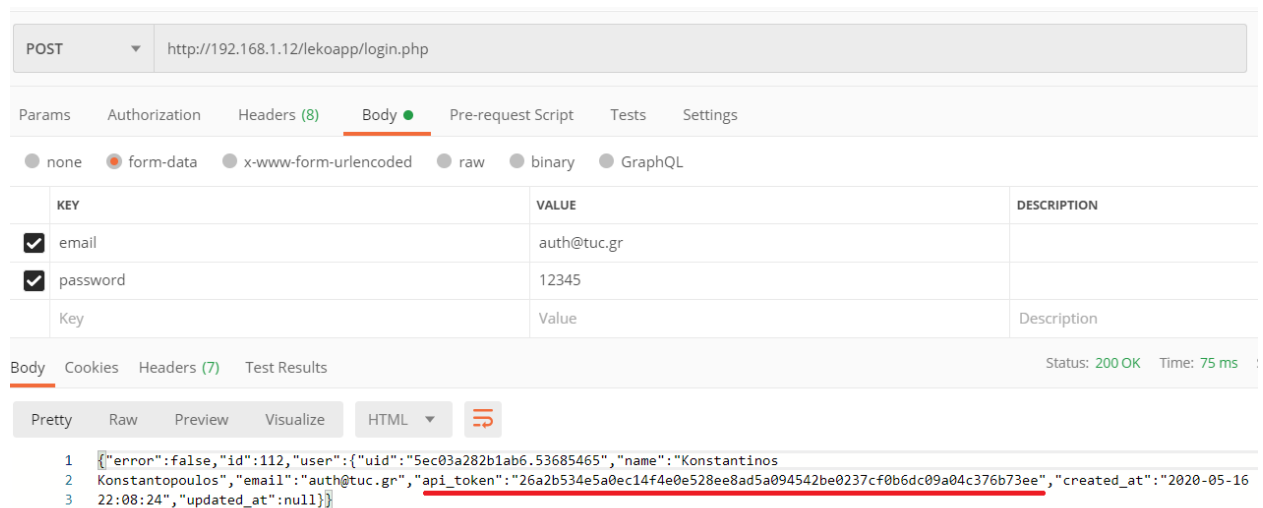
6.7 API Authentication

Για την επιβεβαίωση της λειτουργίας του συστήματος Authentication του API, χρησιμοποιήθηκε το εργαλείο Postman. Στην Εικόνα 6-37, ο χρήστης εγγράφεται στο σύστημα και λαμβάνει το API token που δημιουργήθηκε. Το διακριτικό αυτό αποθηκεύεται και στην βάση δεδομένων για μελλοντική χρήση.



Εικόνα 6-37. API token – register user.

Έπειτα, ο χρήστης, στέλνει αίτημα login στο API. Τονίζεται, ότι για να κάνει ο χρήστης login στο σύστημα δεν απαιτείται το API token.



Εικόνα 6-38. API token – login user.

Όπως φαίνεται και στην Εικόνα 6-38, ο χρήστης κάνει κανονικά login στο σύστημα, και λαμβάνει το API token που είχε δημιουργηθεί κατά την εγγραφή του. Ο χρήστης, με το διακριτικό αυτό, μπορεί να έχει πλέον πρόσβαση σε όλους τους πόρους του συστήματος. Το διακριτικό, στην εφαρμογή android αποθηκεύεται στην SQLite και προστίθεται στον header κάθε αιτήματος. Πρακτικά, η έλλειψη του token, δεν επιτρέπει πρόσβαση στους πόρους του συστήματος, όπως φαίνεται στην Εικόνα 6-39, όπου ο χρήστης στέλνει αίτημα για προσθήκη νέου θερμοκηπίου μέσω της μεθόδου `lekoadd()`.

The screenshot shows a REST client interface with a POST request to `http://192.168.1.12/lekoapp/lekoadd.php`. The 'Body' tab is selected, and the request body is `{ "error": true, "error_msg": "API Authentication Failed" }`. The status is 200 OK and the time is 46 ms.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> userId	112	
<input checked="" type="checkbox"/> type	1	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 46 ms

Pretty Raw Preview Visualize HTML

```
1 { "error": true, "error_msg": "API Authentication Failed" }
```

Εικόνα 6-39. API authentication failed

Με την προσθήκη του token στο header του request, επιτρέπεται κανονικά η πρόσβαση στους πόρους του συστήματος. Ο χρήστης, επαναλαμβάνει το request για προσθήκη θερμοκηπίου, και όπως φαίνεται στην Εικόνα 6-40, το αίτημα ολοκληρώνεται επιτυχώς.

The screenshot shows the same REST client interface, but now the 'Headers' tab is selected and circled in red. The 'api_token' header is added with a long alphanumeric value. The request body is `{ "error": false, "lekoId": "5ec03b6925c9c6.88816005", "created_at": "2020-05-16 22:13:45" }`. The status is 200 OK and the time is 63 ms.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> api_token	26a2b534e5a0ec14f4e0e528ee8ad5a094542be0237cf0b6dc09a0...	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 63 ms

Pretty Raw Preview Visualize HTML

```
1 { "error": false, "lekoId": "5ec03b6925c9c6.88816005", "created_at": "2020-05-16 22:13:45" }
```

Εικόνα 6-40. API authentication successful.

7. API Performance Test

Τα test απόδοσης, προκύπτουν από την δημιουργία εικονικών συνδέσεων μέσω load test. Τα δεδομένα που προκύπτουν από το load test, περιγράφουν την απόδοση του API σε σενάρια με αυξανόμενο traffic. Σκοπός είναι να παρατηρηθεί η συμπεριφορά συστήματος και η ικανότητα χειρισμού του σε περιόδους αιχμής. Συγκεκριμένα, θα προσδιοριστεί ο αριθμός ταυτόχρονων χρηστών και θερμοκηπίων που μπορεί να διαχειριστεί το σύστημα, και η απόδοση του συστήματος για αυτόν τον αριθμό.

Ένα performance test, θα πρέπει να αποτυπώνει για διαφορετικά traffic, τις ακόλουθες μετρήσεις: τον χρόνο απόκρισης του API, το latency, το Throughput της πληροφορίας, τα Success και error rates, καθώς και την χρήση CPU & Memory [48].

Για λόγους απλότητας, το API που θα υποβληθεί στο Performance Test, δεν θα παρέχει την λειτουργία του Authentication.

7.1 Πλάνο Testing – Φάσεις Πειράματος

Στο αρχικό setup, στο σύστημα προστέθηκαν εκατό νέοι χρήστες, με τρία θερμοκήπια ο καθένας. Για τα τριακόσια θερμοκήπια, προστέθηκε μία αναφορά για κάθε ημέρα της εβδομάδας, οδηγώντας σε 2100 εγγραφές στους πίνακες history και events. Το setup είναι απαραίτητο, για να λειτουργήσουν και τα υπόλοιπα endpoints.

Αφού γίνει το αρχικό setup και υπάρχουν διαθέσιμα δεδομένα για τα test, θα πραγματοποιηθούν 4 διαδοχικοί κύκλοι με διαφορετικούς εικονικούς χρήστες (VUs), ταυτόχρονα ενεργούς. Η μετάβαση στον επόμενο κύκλο θα γίνεται, μόνο εάν το σύστημα ανταπεξέρχεται στον προηγούμενο και διατηρεί αποδεκτό error rate. Έπειτα από κάθε κύκλο, θα εκτελούνται οι δύο monitoring λειτουργίες. Για πιο ρεαλιστικά αποτελέσματα στο monitoring, οι περιβαλλοντικές τιμές των ανανεώσεων θα είναι εκτός ορίων με ποσοστό 15%. Ομοίως, 15% των θερμοκηπίων θα προσαρμοστούν με κατάλληλο query manually, ώστε οι ανανεώσεις τους να είναι σε χρονικό διάστημα μεγαλύτερο των 12 ωρών από την εκτέλεση του monitoring συνδεσιμότητας.

Οι κύκλοι είναι οι ακόλουθοι:

- 1^{ος} κύκλος: 100 VUs
- 2^{ος} κύκλος: 200 VUs
- 3^{ος} κύκλος: 500 VUs
- 4^{ος} κύκλος: 1000 VUs

Τα transactions που θα προσομοιωθούν σε κάθε κύκλο, φαίνονται στον **Πίνακα 7-1**, και αντιστοιχούν στα endpoints του API.

Πίνακας 7-1. Test Transactions	
Transaction	Μέθοδος
Register Χρήστη	register.php
Login Χρήστη	login.php
Προσθήκη νέου Leko Χρήστη	lekoadd.php
Επιστροφή αριθμού Leko Χρήστη	lekonum.php
Επιστροφή αντικειμένων Leko Χρήστη	lekoreturn.php
Προσθήκη Διακριτικού Συσκευής Χρήστη	addDevice.php
Λήψη εβδομαδιαίας αναφοράς περ. τιμών από Χρήστη	history.php
Λήψη εβδομαδιαίας αναφοράς δραστηριότητας ενεργοποιητών από Χρήστη	eventhistory.php
Ανανέωση κατάστασης από θερμοκήπιο Leko	addevent.php
Λειτουργία Monitoring Περιβαλλοντικών τιμών θερμοκηπίου	checkLekos.php
Λειτουργία Monitoring Αδράνειας θερμοκηπίου	checkInactive.php

Οι μετρήσεις που θα ληφθούν σε κάθε κύκλο είναι οι ακόλουθες:

- Response Time
- Throughput
- Αριθμός Πακέτων
- Χρήση CPU και μνήμης
- Ανίχνευση Slow Queries

7.2 Περιβάλλον Testing

Για να ανιχνευθεί η απόδοση του API χωρίς να επηρεάζεται από την απόδοση του server, τα tests θα πραγματοποιηθούν σε τοπικό δίκτυο με την χρήση δύο υπολογιστών. Ο ένας υπολογιστής θα φέρει τον Server με το API και την βάση δεδομένων, και ο άλλος θα φέρει το εργαλείο προσομοίωσης των clients (Load Tester). Τα χαρακτηριστικά των μονάδων φαίνονται στον **Πίνακας 7-2**.

Πίνακας 7-2. Μονάδες Test			
Μονάδα	Λειτουργικό Σύστημα	CPU	Ram
Load Generator	Windows 10 Pro	Intel(R) Core(TM) i5-6200U CPU @2.30GHz 2.40GHz	8GB
Web Server	Windows 8.1	Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 2.40GHz	8GB

7.3 Εργαλεία Testing

Τα εργαλεία που θα χρησιμοποιηθούν στο test είναι τα ακόλουθα:

- **WampServer:** Θα χρησιμοποιηθεί στην μονάδα του Web Server, για την εγκατάσταση του API και της βάσης δεδομένων.
- **Jmeter:** Στην δεύτερη μονάδα, εγκαταστάθηκε το εργαλείο Jmeter. Το Jmeter, χρησιμοποιείται για την μαζική αποστολή HTTP αιτημάτων και την δημιουργία σεναρίων. Βασίζεται στην δημιουργία thread groups, που αποτελούνται από samplers και listeners, για την δημιουργία αιτημάτων και την καταγραφή τους αντίστοιχα [49].
- **PerfMon:** Για την ανάλυση της χρήσης CPU και μνήμης στην μονάδα του Web Server.
- **SQL Slow queries:** Παρέχεται σε μορφή log από το WampServer
- **Wireshark:** Για την ανάλυση του δικτύου και των πακέτων.

7.4 Εκτέλεση Performance Test

Στο Jmeter, προστέθηκαν όλα τα endpoints του API. Για την δημιουργία σεναρίων χρησιμοποιήθηκε η δυνατότητά του για μεταφορά των παραμέτρων δυναμικά από τον έναν HTTP sampler στον επόμενο. Με αυτόν τον τρόπο, ένας χρήστης, θα κάνει register, και με το id του θα δημιουργηθεί έναν νέο θερμοκήπιο, με το lekoId του θα προστεθούν τα updates της κατάστασής του κοκ.

Σε όλους τους κύκλους, προστέθηκε ένας μεταβλητός χρόνος Think Time, που προσομοιώνει την καθυστέρηση το χρήστη να μεταβεί από το ένα αίτημα στο επόμενο. Ο χρόνος αυτός, είναι τυχαίος σε συγκεκριμένο εύρος 2 sec με 3 sec. Επιπλέον σε κάθε κύκλο, εφαρμόστηκε και ξεχωριστό ramp-up time, που είναι ο χρόνος μέχρι να ξεκινήσουν όλα τα threads.

7.4.1 Εκτέλεση 1^{ου} κύκλου

Πίνακας 7-3. Παράμετροι 1 ^{ου} Κύκλου				
Αριθμός Χρηστών	εικονικών	Ramp-up time	Total test time	Think Time
100		30 sec	10 min	2-3 sec

Παρατηρήσεις:

- Το test πραγματοποιήθηκε ομαλά. Μόνο 2 requests απέτυχαν
- Η χρήση CPU και μνήμης ήταν αρκετά κάτω των ορίων
- Όλα τα SQL queries εκτελέστηκαν σε χρόνο κάτω του 1 sec
- Τα response times των requests του monitoring, για την επίβλεψη των περιβαλλοντικών τιμών και της σύνδεσης, ήταν αρκετά μεγάλα, με περίπου 17 λεπτά και 1,3 λεπτά αντίστοιχα.
- Όλα τα υπόλοιπα response times του API ήταν κάτω των 2 sec

7.4.2 Εκτέλεση 2^{ου} κύκλου

Πίνακας 7-4. Παράμετροι 2 ^{ου} Κύκλου				
Αριθμός Χρηστών	εικονικών	Ramp-up time	Total test time	Think Time
200		60 sec	10 min	2-3 sec

Παρατηρήσεις:

- Εμφανίστηκαν κάποια σφάλματα, της τάξεως του 0,34%
- Για ένα μικρό χρονικό διάστημα, η χρήση του CPU αυξήθηκε, αλλά επέστρεψε σε χαμηλά επίπεδα
- Όλα τα SQL queries εκτελέστηκαν σε χρόνο κάτω του 1 sec
- Τα response times των requests του monitoring, για την επίβλεψη των περιβαλλοντικών τιμών και της σύνδεσης, ήταν αρκετά μεγάλα, με περίπου 30 λεπτά και 1,5 λεπτά αντίστοιχα.
- Όλα τα υπόλοιπα response times του API ήταν μεταξύ 3 sec με 7 sec

7.4.3 Εκτέλεση 3^{ου} κύκλου

Πίνακας 7-5. Παράμετροι 3 ^{ου} Κύκλου				
Αριθμός Χρηστών	εικονικών	Ramp-up time	Total test time	Think Time
500		120 sec	10 min	2-3 sec

Παρατηρήσεις:

- Περισσότερα από το 60% των requests απέτυχαν
- Για ένα μικρό χρονικό διάστημα, η χρήση του CPU αυξήθηκε, αλλά επέστρεψε σε χαμηλά επίπεδα
- Όλα τα SQL queries εκτελέστηκαν σε χρόνο κάτω του 1 sec
- Τα response times των requests του monitoring, για την επίβλεψη των περιβαλλοντικών τιμών και της σύνδεσης, ήταν αρκετά μεγάλα, με περίπου 28 λεπτά και 1,5 λεπτά αντίστοιχα.
- Όλα τα υπόλοιπα response times του API ήταν μεταξύ 3 sec με 7 sec

Ο 4^{ος} κύκλος δεν πραγματοποιήθηκε λόγω του υψηλού error rate του 3^{ου} κύκλου. Από τα αποτελέσματα, προκύπτει ότι με το συγκεκριμένο setup, το σύστημα μπορεί να υποστηρίξει ένα load 250 με 300 ταυτόχρονα συνδεδεμένων χρηστών. Πρακτικά, εφόσον το σύνολο των χρηστών του συστήματος σε ρεαλιστικά σενάρια δεν θα χρησιμοποιεί τους πόρους την ίδια χρονική στιγμή, μπορούμε να πούμε ότι το σύστημα μπορεί να υποστηρίξει αρκετά περισσότερους εγγεγραμμένους χρήστες.

Κατά την διάρκεια του test, προέκυψαν τα παρακάτω σφάλματα:

- Response message: Non-HTTP response message: 192.168.1.12:80 failed to respond
- Java.net.SocketException: Connection reset by peer
- HttpHostConnectException: Connect to 192.168.1.12:80 failed: Connection refused: connect

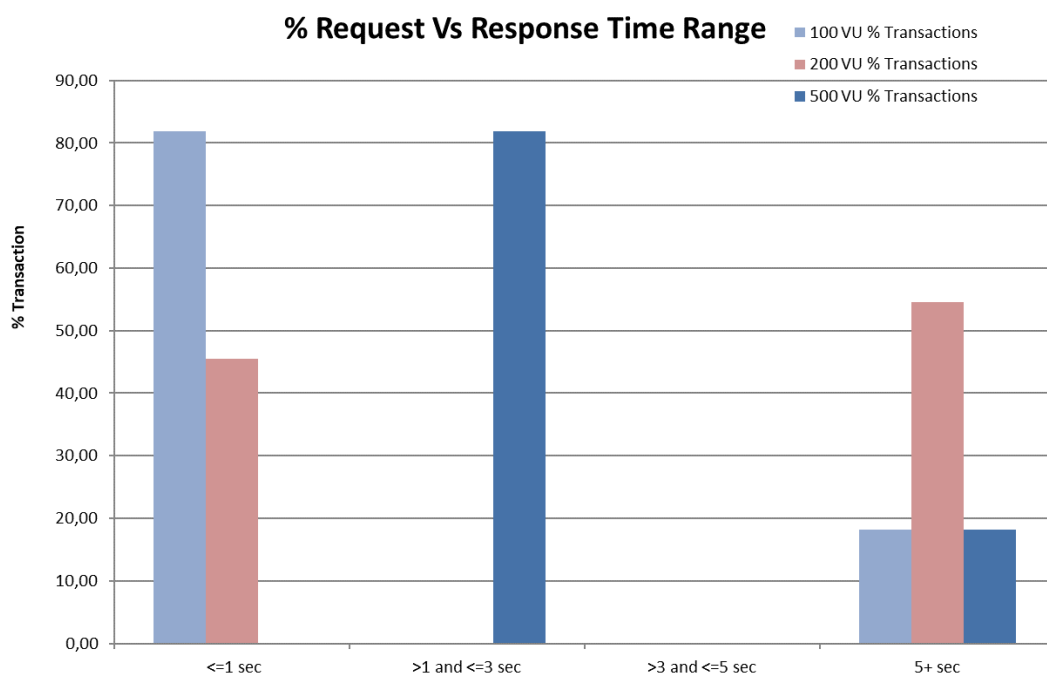
7.5 Αποτελέσματα

Κάθε κύκλος διήρκεσε 10 λεπτά. Τα συνολικά requests που εκτελέστηκαν ήταν 57793, 57992 και 158884 για τον 1ο, 2ο και 3ο κύκλο αντίστοιχα.

Πίνακας 7-6. Επισκόπηση Test Cycles				
VUs	Χρόνος Test (Λεπτά)	#Request που εκτελέστηκαν	Throughput (Req/Sec)	% Error
100	10	57793	97	0,01%
200	10	57992	97	0,34%
500	10	158884	265	62,08%

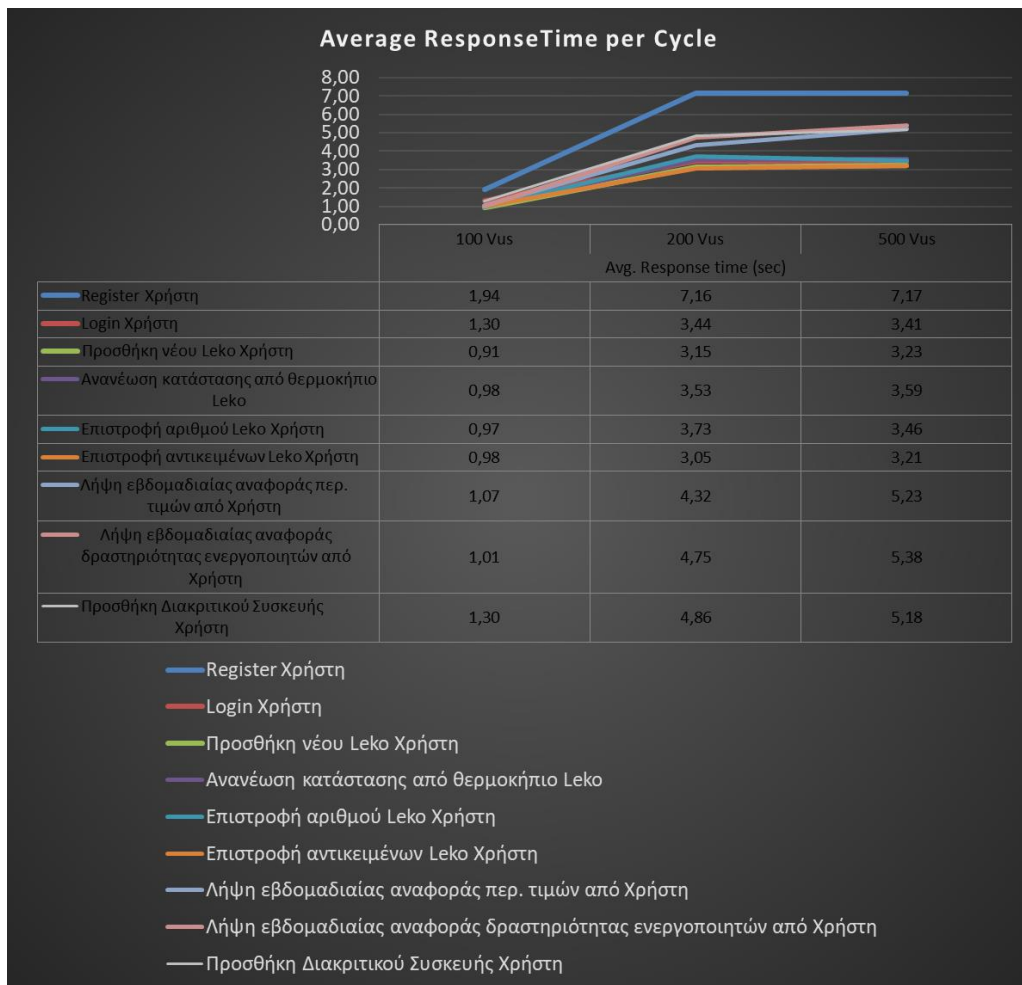
Στον **Πίνακα 7-7** φαίνεται η κατανομή του 90στου εκατοστημορίου των response times όλων των transactions για κάθε κύκλο. Τα περισσότερα transactions και στους 3 κύκλους, διήρκεσαν κάτω από 3 sec. Σε όλους τους κύκλους, τα monitoring transactions διήρκεσαν περισσότερο από 5 sec.

Πίνακας 7-7. Κατανομή Response time ανά κύκλο						
Response Time (90%) Range	100 VU		200 VU		500 VU	
	# Transactions	% Transactions	# Transactions	% Transactions	# Transactions	% Transactions
<=1 sec	9	81,82	5	45,45	0	0,00
>1 & <=3 sec	0	0,00	0	0,00	9	81,82
>3 & <=5 sec	0	0,00	0	0,00	0	0,00
5+ sec	2	18,18	6	54,55	2	18,18



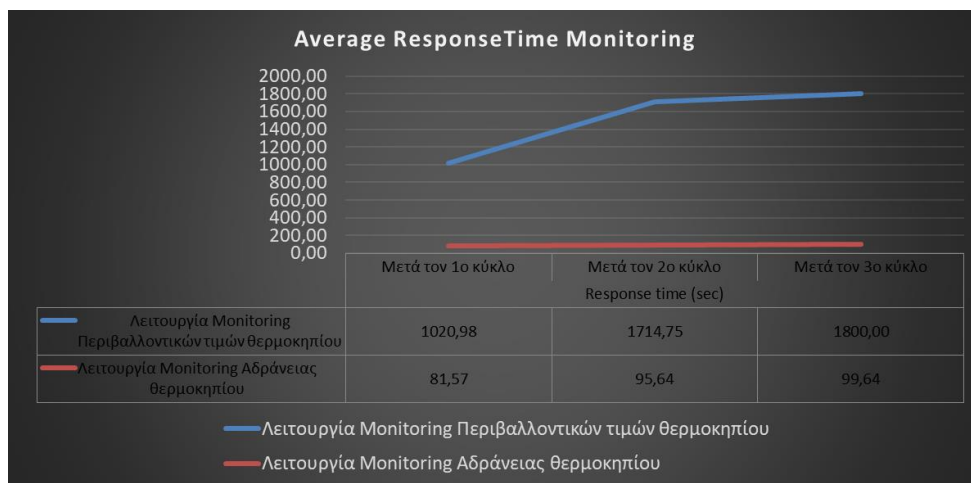
Σχήμα 7-1. Request vs 90% Response time Range.

Στο Σχήμα 7-2 φαίνεται ο πίνακας και το διάγραμμα των μέσων response time, ανά request, ανά κύκλο. Παρατηρούμε, ότι το μεγαλύτερο response time ανήκει στο register request και στους 3 κύκλους. Ακολουθούν οι εβδομαδιαίες αναφορές περιβαλλοντικών τιμών και δραστηριότητας των ενεργοποιητών.



Σχήμα 7-2. Average response time per transaction.

Στο Σχήμα 7-3 παρουσιάζονται τα response times, για το monitoring μετά από κάθε κύκλο. Παρατηρούμε, ότι τα response times του monitoring περιβαλλοντικών τιμών είναι μεγάλα. Οι χρόνοι αυξάνονται από κύκλο σε κύκλο, λόγω της αύξησης των εγγραφών στον πίνακα Leko, όπου πραγματοποιούνται και τα δύο monitoring functions.



Σχήμα 7-3. Monitoring Response times after each cycle.

7.5.1 Response Times ανά transaction

Στους Πίνακες Πίνακας 7-8, Πίνακας 7-9 Πίνακας 7-10 φαίνονται αναλυτικότερα τα response times των 3^{ων} κύκλων ανά transaction. Ενδεικτικό είναι το 90% line, που αντιπροσωπεύει το ενενηντακοστό εκατοστημόριο των αυξητικά ταξινομημένων response times. Φαίνεται ότι στις περισσότερες περιπτώσεις, ο μέσος όρος των response time αυξάνεται από το υπόλοιπο 10% με την μεγαλύτερη τιμή.

Πίνακας 7-8. Response times 1 ^{ου} κύκλου							
Requests	Request Name	1 ^{ος} Κύκλος (100 Vus)					
		#Sample Executed	Avg Response Time (In sec)	90% Line	Min	Max	Req/sec
Leko API	Register Χρήστη	1904,00	1,94	0,04	0,01	42,58	3,18
	Login Χρήστη	1890,00	1,30	0,03	0,00	42,13	3,16
	Προσθήκη νέου Leko Χρήστη	5608,00	0,91	0,03	0,01	41,79	9,41
	Ανανέωση κατάστασης από θερμικήπιο Leko	39249,00	0,98	0,04	0,01	42,61	65,41
	Επιστροφή αριθμού Leko Χρήστη	1845,00	0,97	0,03	0,00	40,97	3,12
	Επιστροφή αντικειμένων Leko Χρήστη	1839,00	0,98	0,03	0,01	40,56	3,12
	Λήψη εβδομαδιαίας αναφοράς περ. τιμών από Χρήστη	1830,00	1,07	0,07	0,02	41,15	3,11
	Λήψη εβδομαδιαίας αναφοράς δραστηριότητας ενεργοποιητών από Χρήστη	1819,00	1,01	0,07	0,02	41,60	3,11
	Προσθήκη Διακριτικού Συσκευής Χρήστη	1809,00	1,30	0,03	0,01	41,65	3,10

Πίνακας 7-9. Response times 2^{ου} κύκλου

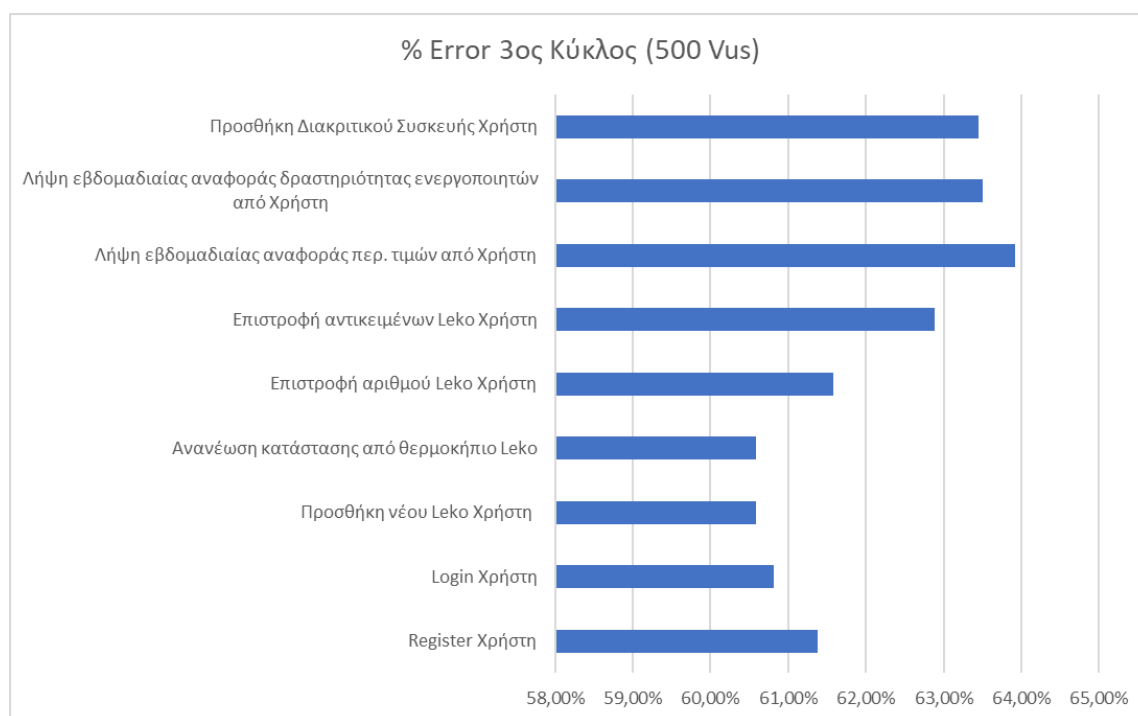
Requests	Request Name	2 ^{ος} Κύκλος (200 Vus)					
		#Sample Executed	Avg Response Time (In sec)	90% Line	Min	Max	Req/sec
Leko API	Register Χρήστη	1956	7,16	41,84	0,00	54,84	3,26
	Login Χρήστη	1939	3,44	0,13	0,00	54,20	3,23
	Προσθήκη νέου Leko Χρήστη	5702	3,15	0,09	0,00	55,57	9,55
	Ανανέωση κατάστασης από θερμοκήπιο Leko	39314	3,53	0,32	0,00	55,87	65,52
	Επιστροφή αριθμού Leko Χρήστη	1854	3,73	0,17	0,00	55,39	3,13
	Επιστροφή αντικειμένων Leko Χρήστη	1835	3,05	0,10	0,00	55,15	3,11
	Λήψη εβδομαδιαίας αναφοράς περ. τιμών από Χρήστη	1817	4,32	28,51	0,00	55,28	3,09
	Λήψη εβδομαδιαίας αναφοράς δραστηριότητας ενεργοποιητών από Χρήστη	1794	4,75	36,59	0,00	55,74	3,07
	Προσθήκη Διακριτικού Συσκευής Χρήστη	1781	4,86	38,43	0,00	55,15	3,06

Πίνακας 7-10. Response times 3^{ου} κύκλου

Requests	Request Name	3 ^{ος} Κύκλος (500 Vus)					
		#Sample Executed	Avg Response Time (In sec)	90% Line	Min	Max	Req/sec
Leko API	Register Χρήστη	5303	7,17	2,33	0,01	61,23	8,79
	Login Χρήστη	5255	3,41	2,06	0,00	61,75	8,71
	Προσθήκη νέου Leko Χρήστη	15439	3,23	2,06	0,00	61,80	25,64
	Ανανέωση κατάστασης από θερμοκήπιο Leko	108073	3,59	2,07	0,00	62,09	180,12
	Επιστροφή αριθμού Leko Χρήστη	5055	3,46	2,09	0,00	61,12	8,45
	Επιστροφή αντικειμένων Leko Χρήστη	5016	3,21	2,08	0,00	60,35	8,41
	Λήψη εβδομαδιαίας αναφοράς περ. τιμών από Χρήστη	4966	5,23	2,06	0,00	61,62	8,33
	Λήψη εβδομαδιαίας αναφοράς δραστηριότητας ενεργοποιητών από Χρήστη	4921	5,38	2,07	0,00	61,16	8,29
	Προσθήκη Διακριτικού Συσκευής Χρήστη	4856	5,18	2,05	0,00	61,12	8,19

7.5.2 Errors ανά transaction

Το error rate στους πρώτους 2 κύκλους ήταν αμελητέο. Όπως φαίνεται και στο Σχήμα 7-4. Error Rates 3^{ου} Κύκλου., στον 3^ο κύκλο το error rate έφτασε σχεδόν το 64%.



Σχήμα 7-4. Error Rates 3^{ου} Κύκλου.

7.5.3 Bytes & Request Throughput ανά transaction

Στους Πίνακες Πίνακας 7-11, Πίνακας 7-12 και Πίνακας 7-13, φαίνονται οι μετρήσεις που αφορούν το Throughput των κύκλων ανά request και ανά second. Οι μετρήσεις αφορούν τα δεδομένα που στάλθηκαν από τον Load Generator και τα δεδομένα που ελήφθησαν ως απάντηση από τον Web Server.

Το μεγαλύτερο payload ανήκει στην ανανέωση της κατάστασης του θερμοκηπίου, όπως ήταν αναμενόμενο. Η ανανέωσης της κατάστασης απαιτεί τις περισσότερες παραμέτρους από όλα τα υπόλοιπα endpoints του API (βλ. Πίνακας 5-6).

Πίνακας 7-11. Throughput analysis 1 ^{ου} κύκλου					
Requests	Request Name	1 ^{ος} Κύκλος (100 Vus)			
		Received KB/Sec	Received KB/Request	Sent KB/Sec	Sent KB/Request
Leko API	Register Χρήστη	1,50	0,47	1,17	1,71
	Login Χρήστη	1,50	0,47	0,98	1,44
	Προσθήκη νέου Leko Χρήστη	2,98	0,32	2,33	6,88
	Ανανέωση κατάστασης από θερμοκήπιο Leko	11,01	0,51	3,44	10,21
	Επιστροφή αριθμού Leko Χρήστη	0,81	0,26	0,73	0,59
	Επιστροφή αντικειμένων Leko Χρήστη	2,34	0,75	0,75	1,76
	Λήψη εβδομαδιαίας αναφοράς περ. τιμών από Χρήστη	1,67	0,54	0,86	1,44
	Λήψη εβδομαδιαίας αναφοράς δραστηριότητας ενεργοποιητών από Χρήστη	1,51	0,48	0,87	1,31
	Προσθήκη Διακριτικού Συσκευής Χρήστη	1,51	0,49	1,30	1,96
	Λειτουργία Monitoring Περιβαλλοντικών τιμών θερμοκηπίου	6,06	26910,60	0,00	0,00
	Λειτουργία Monitoring Αδράνειας θερμοκηπίου	7,10	543,25	0,00	0,00

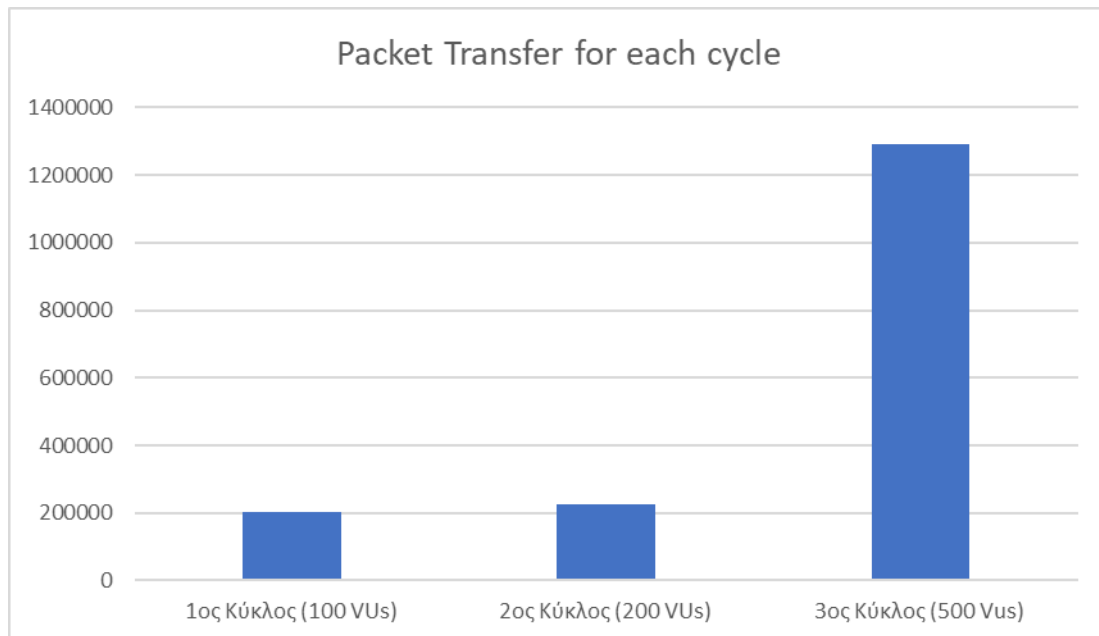
Πίνακας 7-12. Throughput analysis 2 ^{ου} κύκλου					
Requests	Request Name	2 ^{ος} Κύκλος (200 Vus)			
		Received KB/Sec	Received KB/Request	Sent KB/Sec	Sent KB/Request
Leko API	Register Χρήστη	1,56	0,48	1,14	8,38
	Login Χρήστη	1,55	0,49	0,96	3,37
	Προσθήκη νέου Leko Χρήστη	3,09	0,32	2,31	7,35
	Ανανέωση κατάστασης από θερμοκήπιο Leko	11,07	0,51	3,40	10,27
	Επιστροφή αριθμού Leko Χρήστη	0,82	0,31	0,73	2,76
	Επιστροφή αντικειμένων Leko Χρήστη	2,36	0,81	0,75	2,29
	Λήψη εβδομαδιαίας αναφοράς περ. τιμών από Χρήστη	1,69	0,58	0,85	3,67
	Λήψη εβδομαδιαίας αναφοράς δραστηριότητας ενεργοποιητών από Χρήστη	1,50	0,51	0,86	4,08
	Προσθήκη Διακριτικού Συσκευής Χρήστη	1,52	0,50	1,27	6,17
	Λειτουργία Monitoring Περιβαλλοντικών τιμών θερμοκηπίου	5,81	33638,25	0,00	0,00
	Λειτουργία Monitoring Αδράνειας θερμοκηπίου	5,68	640,63	0,00	0,00

Το υψηλό error rate του 3^{ου} κύκλου δεν επέτρεψε την καταγραφή των μετρήσεων για τα KB/request που ελήφθησαν.

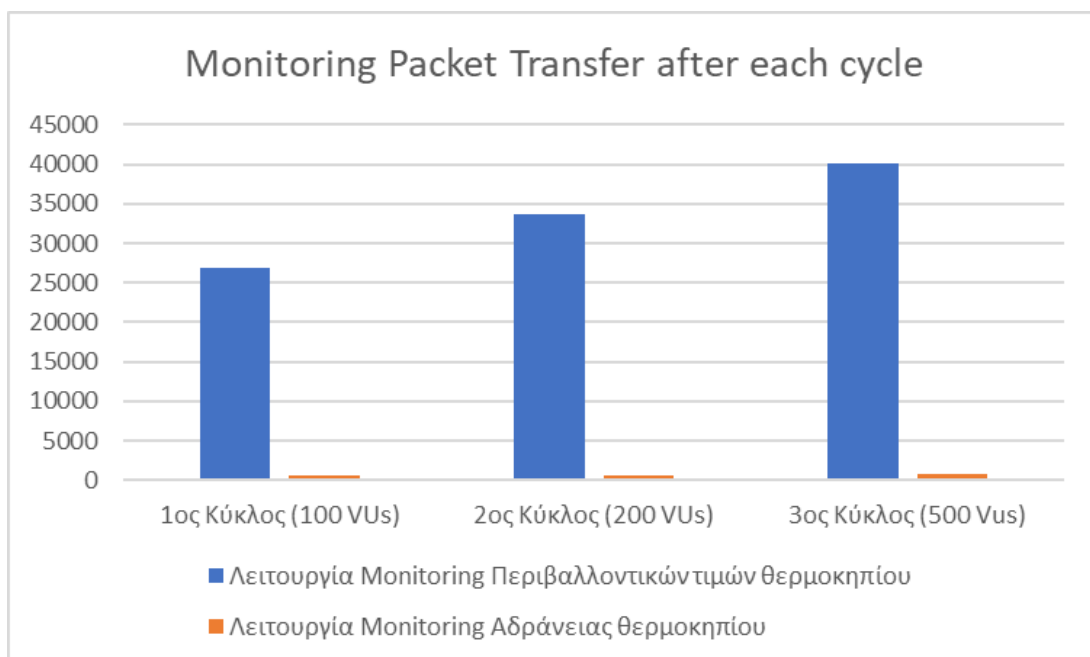
Πίνακας 7-13. Throughput analysis 3 ^{ου} κύκλου					
Requests	Request Name	3 ^{ος} Κύκλος (500 Vus)			
		Received KB/Sec	Received KB/Request	Sent KB/Sec	Sent KB/Request
Leko API	Register Χρήστη	15,90	--	1,22	6,19
	Login Χρήστη	15,58	--	1,04	3,55
	Προσθήκη νέου Leko Χρήστη	44,35	--	2,48	8,01
	Ανανέωση κατάστασης από θερμοκήπιο Leko	154,87	--	3,40	11,27
	Επιστροφή αριθμού Leko Χρήστη	14,63	--	0,77	2,66
	Επιστροφή αντικειμένων Leko Χρήστη	26,73	--	0,76	2,44
	Λήψη εβδομαδιαίας αναφοράς περ. τιμών από Χρήστη	15,73	--	0,83	2,68
	Λήψη εβδομαδιαίας αναφοράς δραστηριότητας ενεργοποιητών από Χρήστη	15,42	--	0,85	2,87
	Προσθήκη Διακριτικού Συσκευής Χρήστη	15,16	--	1,26	4,01
	Λειτουργία Monitoring Περιβαλλοντικών τιμών θερμοκηπίου	4,56	--	0,00	0,00
	Λειτουργία Monitoring Αδράνειας θερμοκηπίου	6,24	820,00	0,00	0,00

7.5.4 Packet transfer ανά κύκλο

Στα φαίνεται ο αριθμός των πακέτων της επικοινωνίας κάθε κύκλου, όπως μετρήθηκαν από το εργαλείο Wireshark. Κανένα πακέτο δεν έγινε dropped. Ουσιαστικά, το πρωτόκολλο HTTP, διεκπεραιώνει την επαναποστολή των αποτυχημένων προσπαθειών λόγω συμφόρησης του δικτύου, μεταφέροντας εν τέλει την πληροφορία, αλλά με επίπτωση στο Response time.



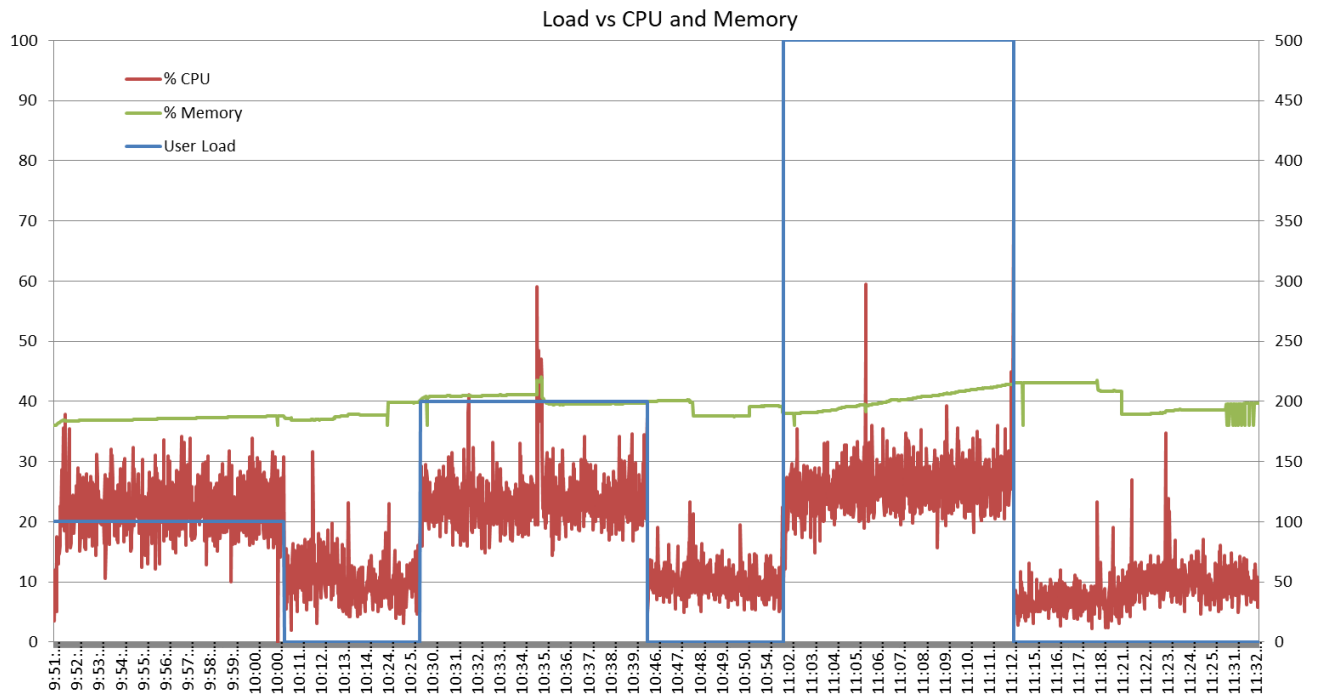
Σχήμα 7-5. Αριθμός Πακέτων ανά κύκλο.



Σχήμα 7-6. Αριθμός Πακέτων λειτουργίας Monitoring έπειτα από κάθε κύκλο.

7.5.5 Χρήση CPU & Μνήμης

Στο γράφημα φαίνεται η χρήση CPU & Memory κατά την διάρκεια και ενδιάμεσα των 3 δεκάλεπτων κύκλων. Παρατηρείται ότι δεν υπήρξε έντονο stress στους πόρους του συστήματος, παρόλο που τα resources του Web Server ήταν σχετικά περιορισμένης υπολογιστικής ικανότητας. Το API, του συστήματος δεν είναι resource intensive.



Σχήμα 7-7. CPU & Memory utilization.

7.6 Απολογισμός Αποτελεσμάτων

Το API υπεβλήθη σε διαδοχικούς κύκλους προσομοίωσης έως την αποτυχία. Από τις μετρήσεις των πόρων και του χρόνου εκτέλεσης των queries αποδείχθη ότι το API είναι αρκετά ελαφρύ. Το bottleneck του συστήματος ήταν το δίκτυο.

Από το πείραμα παρατηρήθηκε επίσης ότι οι χρόνοι του monitoring ήταν αρκετά υψηλοί. Στον 3^ο κύκλο ο έλεγχος των περιβαλλοντικών τιμών διήρκησε 30 λεπτά, για τον έλεγχο 19698 θερμοκηπίων. Μία εναλλακτική λύση για την εξάλειψη του χρόνου, θα ήταν ο έλεγχος των περιβαλλοντικών τιμών να γίνεται αυτόματα κατά την ανανέωση της κατάστασης του θερμοκηπίου. Πρακτικά, η λύση αυτή θα είναι πιο αποδοτική, καθώς θα αποφευχθεί η προσπέλαση ολόκληρου του πίνακα και ο επανέλεγχος των ίδιων εγγραφών για θερμοκήπια των οποίων οι τιμές δεν έχουν ανανεωθεί ενδιάμεσα των ελέγχων.

Στο πέρας του τρίτου κύκλου, η βάση δεδομένων του συστήματος είχε :

- 8009 χρήστες
- 19698 θερμοκήπια Leko
- 7237 συσκευές
- 130189 εγγραφές ανανέωσης κατάστασης στους πίνακες history και events

Από το Test προκύπτει, ότι ο μέσος όρος όγκου των συνολικών δεδομένων (sent/received) για κάθε request είναι περίπου 5 KB. Με βάση αυτήν την εκτίμηση, ένας χρήστης με τρία θερμοκήπια στο σύστημα, θα καταναλώσει 40KB πληροφορίας, για να κάνει login, να προσθέσει την συσκευή του, και να λάβει τις εβδομαδιαίες αναφορές περιβαλλοντικών τιμών και δραστηριότητας ενεργοποιητών για το κάθε ένα. Με την χρέωση των πάροχων κινητής να κυμαίνεται στα 7 με 9 λεπτά του ευρώ ανά MB, το κόστος που προκύπτει για τον παραπάνω υποτιθέμενο χρήστη του συστήματος θα είναι περίπου ένα τρίτο του λεπτού του ευρώ.

8. Συμπεράσματα & Μελλοντικές βελτιώσεις

Το σύστημα που υλοποιήθηκε στην παρούσα διπλωματική εργασία αποτελεί ένα αξιόπιστο σύστημα διαχείρισης του θερμοκηπίου Leko. Τα σημαντικότερα χαρακτηριστικά του συστήματος αυτού είναι η ευχρηστία και η πρόληψη πιθανών σφαλμάτων που μπορεί να προκύψουν κατά τη διάρκεια της καλλιέργειας ενός φυτού στο θερμοκήπιο. Το σύστημα διαχείρισης αυτό ολοκληρώνει ένα γενικότερο σύστημα και έναν διαφορετικό τρόπο ανάπτυξης ενός φυτού με τη λιγότερη δυνατή ασχολία από την πλευρά του χρήστη. Η δυνατότητα αυτή που παρέχει το σύστημα, το καθιστά ανταγωνιστικό στην αγορά και υποψήφιο προϊόν που μπορεί να αποτελέσει μέσο καλλιέργειας για τους ανθρώπους κυρίως της πόλης.

8.1 Συμπεράσματα και γνώση που αποκτήθηκε

Η υλοποίηση του συστήματος διαχείρισης που αφορά την παρούσα διπλωματική, μου έδωσε πολύτιμες γνώσεις, σε τέσσερα διαφορετικά επίπεδα.

Το πρώτο επίπεδο, είναι της εξερεύνησης και της αποδοτικής έρευνας. Το δεύτερο επίπεδο, είναι του σχεδιασμού ενός πλήρους συστήματος, με top-down προσέγγιση, και η σημαντικότητα αυτής της μεθόδου. Το τρίτο επίπεδο, είναι το τεχνικό, και αφορά τις τεχνικές γνώσεις προγραμματισμού και εφαρμογής τεχνολογιών που έπρεπε να αποκτήσω. Το τελευταίο επίπεδο, και ίσως το πιο σημαντικό, είναι η εξοικείωση με την επιμονή και την υπομονή που χρειάζεται κάθε προσπάθεια τέτοιας έκτασης, η αξία και η ψυχική ανταμοιβή της αυτό-μάθησης και η συμφιλίωση με την αβεβαιότητα, που από την φύση της είναι δυσάρεστη, σε κάθε ξεκίνημα που εμπεριέχει γνωσιακά κενά.

8.2 Μελλοντικές Βελτιώσεις

Παρακάτω παρατίθενται συνοπτικά μελλοντικές βελτιώσεις του συστήματος, με σκοπό να γίνει το σύστημα καλύτερο για τον τελικό χρήστη.

8.2.1 Μετατροπή σε MQTT ή CoAP

Το θερμοκήπιο Leko, έχει αναπτυχθεί σε πρωτόκολλο HTTP. Οι περισσότερες εφαρμογές IoT χρησιμοποιούν πιο ευέλικτα πρωτόκολλα όπως το MQTT και το CoAP. Τα παραπάνω πρωτόκολλα καταναλώνουν λιγότερο bandwidth στο δίκτυο, λόγω μικρότερου overhead στα πακέτα. Επίσης έχουν καλύτερη απόδοση όσο αφορά τα δίκτυα περιορισμένου bandwidth, με καλύτερα αποτελέσματα στο Packet Loss για το MQTT και χαμηλά latency για το CoAP[9]. Συνεπώς, ειδικότερα σε μία εφαρμογή όπου το θερμοκήπιο βρίσκεται σε remote τοποθεσία με ασθενές δίκτυο, η μετάβαση είναι απαραίτητη.

8.2.2 Data Analysis

Ο κόσμος της τεχνολογίας σήμερα κατακλύζεται από δεδομένα. Τα δεδομένα αυτά μπορούν να χρησιμοποιηθούν για να προβλεφθεί η να βελτιωθεί μια διαδικασία στο μέλλον. Τα δεδομένα

του συστήματος μπορούν να χρησιμοποιηθούν για να βελτιωθούν οι καλλιέργειες των φυτών σε σημαντικό βαθμό και να συλλέξουμε πληροφορίες για πολλά είδη φυτών μέσα από τη βάση δεδομένων των χρηστών του συστήματος. Η δυνατότητα αυτή είναι ελκυστική όχι μόνο για το σύστημα του θερμοκηπίου αλλά γενικότερα για την γεωπονία.

8.2.3 Ενεργειακή αναφορά θερμοκηπίου

Μία ακόμα βελτίωση είναι η έκδοση ενεργειακών αναφορών για κάθε θερμοκηπίου. Οι ενεργειακές αναφορές, θα μπορούν να βγαίνουν με βάση τα δεδομένα που θα έχουν συλλεχθεί για τους ενεργοποιητές, όπως συνολικά ανοίγματα. Ο υπολογισμός της καταναλισκόμενης ενέργειας θα μπορεί να παρουσιάσει πόση ενέργεια καταναλώθηκε για την ανάπτυξη ενός φυτού. Η μελέτη αυτή μπορεί να έχει πολύτιμη εφαρμογή σε θερμοκήπια επαγγελματικού σκοπού, καθώς θα δίνει μία καλύτερη εικόνα για το κόστος παραγωγής για κάθε τύπο φυτού.

8.2.4 Ειδοποίηση για κλάδεμα ή συγκομιδή

Μια άλλη δυνατότητα που θα μπορούσε να υλοποιηθεί στο μέλλον θα είναι ειδοποίηση για την κατάλληλη χρονική στιγμή κλαδέματος ή συγκομιδής ή ακόμα προσθήκη λιπάσματος στην καλλιέργεια του θερμοκηπίου. Για να γίνει αυτό θα πρέπει το σύστημα να εφοδιαστεί με επιπλέον αισθητήρες που θα καταγραφούν τις διαστάσεις του φυτού, θα λαμβάνουν χημικές ενδείξεις και άλλα. Χρησιμοποιώντας αυτά τα δεδομένα θα μπορούμε να καταλήξουμε και σε συμπεράσματα ιδιαίτερα ωφέλιμα για τις καλλιέργειες αυτές, όπως ποιες είναι οι καταλληλότερες περιβαλλοντικές τιμές για να επιτευχθεί η μέγιστη ανάπτυξη για το κάθε φυτό.

8.2.5 Ειδοποίηση δεξαμενής νερού

Για κάθε θερμοκήπιο αποθηκεύεται στην βάση δεδομένων ο συνολικός αριθμός ποτισμάτων. Λαμβάνοντας υπόψη ότι ο χρόνος ανοίγματος της αντλίας, άρα και η ποσότητα του νερού είναι δεδομένη, το σύστημα μπορεί να αποφανθεί εάν η δεξαμενή είναι σε χαμηλά επίπεδα και να ειδοποιήσει τον χρήστη. Για να συμβεί αυτό, θα πρέπει ο χρήστης να μπορεί να ενημερώνει το σύστημα ότι ξαναγέμισε την δεξαμενή, ούτως ώστε να μηδενίζεται η καταμέτρηση ποτισμάτων.

8.2.6 Διασύνδεση μεταξύ των χρηστών

Η δυνατότητα κοινωνικής δικτύωσης θα πρόσθετε μία επιπλέον αξία στο σύστημα, καθώς ο χρήστης θα μπορούσε να μοιραστεί την πορεία των καλλιεργειών του με τους φίλους του. Η δυνατότητα αυτή πέρα από εμπορική θα πρόσθετε και ποιοτική αξία, καθώς μέσω της ενημέρωσης για την πορεία άλλων καλλιεργειών ο χρήστης θα λάμβανε σημαντικές πληροφορίες.

Η δικτύωση των χρηστών θα μπορούσε να υλοποιηθεί στο ίδιο το σύστημα. Κάτι τέτοιο θα απαιτούσε αρκετές αλλαγές στην βάση δεδομένων και στον διαχειριστή και θα αύξανε σημαντικά την πολυπλοκότητα του συστήματος. Συνεπώς, στο άμεσο μέλλον θα υλοποιηθεί η δικτύωση με την χρήση κάποιου άλλου μέσου κοινωνικής δικτύωσης όπως το Facebook. Με την χρήση του Facebook API στο υπάρχον σύστημα ο χρήστης θα μπορεί να κάνει εγγραφή και login μέσω του λογαριασμού του σε αυτό, καθώς και να κοινοποιήσει την πορεία της καλλιέργειάς του.

8.2.7 Τροποποίηση παραμέτρων θερμοκηπίου από τον χρήστη

Στην παρούσα εργασία οι περιβαλλοντικές παράμετροι είναι σταθερές για κάθε τύπο καλλιέργειας και αποθηκευμένες στο κάθε Leko. Για να μπορεί ο χρήστης να προσθέσει μία εξατομικευμένη καλλιέργεια με διαφορετικά περιβαλλοντικά χωρία, θα πρέπει το θερμοκήπιο να

μπορεί να τα αποθηκεύσει. Θα πρέπει επίσης να αποθηκεύονται και στην βάση δεδομένων για κάθε ένα από τα εξατομικευμένα θερμοκήπια για να μπορεί ο διαχειριστής να εκτελεί την λειτουργία του monitoring.

Ομοίως για να μπορεί ο χρήστης να παρεμβαίνει στους ενεργοποιητές του θερμοκηπίου του, θα πρέπει επίσης το θερμοκήπιο να μπορεί να δέχεται τα αιτήματα αυτά από τον webserver.

9. ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ

- [1] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou and C. Lung, "Smart Home: Integrating Internet of Things with Web Services and Cloud Computing," *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, Bristol, pp. 317-320, 2013.
- [2] P. Waher, *Mastering Internet of things: design and create your own IoT applications using Raspberry Pi 3*. Birmingham, UK: Packt Publishing, 2018.
- [3] K. Ashton, "That 'Internet of Things' thing," *RFID Journal*, pp. 1–1, 2009.
- [4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswamia, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions." *arXiv.org*, 2013. [Online]. Available: <https://arxiv.org/pdf/1207.0203.pdf>. [Accessed: 11-Aug-2020].
- [5] H. Geng, *The internet of things and data analytics handbook*. Hoboken, NJ, USA: John Wiley & Sons, 2017.
- [6] P. Waher, *Learning Internet of Things: explore and learn about Internet of Things with the help of engaging and enlightening tutorials designed for Raspberry Pi*. Birmingham, UK: Packt, 2015.
- [7] V. Gazis et al., "A survey of technologies for the internet of things," *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Dubrovnik, pp. 1090-1095, 2015.
- [8] C. Sharma and N. K. Gondhi, "Communication Protocol Stack for Constrained IoT Systems," *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Bhimtal, pp. 1-6, 2018.
- [9] Y. Chen and T. Kunz, "Performance evaluation of IoT protocols under a constrained wireless access network," *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*, Cairo, pp. 1-7, 2016.
- [10] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," *2017 IEEE International Systems Engineering Symposium (ISSE)*, Vienna, pp. 1-7, 2017.
- [11] "Getting Started with MQTT," *Hivemq*, 14-Jul-2019. [Online]. Available: <https://www.hivemq.com/blog/how-to-get-started-with-mqtt>. [Accessed: 11-Aug-2020].
- [12] S. Bandyopadhyay and A. Bhattacharyya, "Lightweight Internet protocols for web enablement of sensors using constrained gateway devices," *2013 International Conference on Computing, Networking and Communications (ICNC)*, San Diego, CA, pp. 334-340, 2013.
- [13] D. Thangavel, X. Ma, A. Valera, H. Tan and C. K. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Singapore, pp. 1-6, 2014.

- [14] J. Stafford, "Essential Technology for Precision Agriculture," *Proceedings of the Third International Conference on Precision Agriculture ASA, CSSA, and SSSA Books*, pp. 593–604, 2015.
- [15] F. J. Pierce and P. Nowak, "Aspects of Precision Agriculture," *Advances in Agronomy Advances in Agronomy Volume 67*, pp. 1–85, 1999.
- [16] S. S. Reka, B. K. Chezian, and S. S. Chandra, "A Novel Approach of IoT-Based Smart Greenhouse Farming System," *Springer Transactions in Civil and Environmental Engineering Green Buildings and Sustainable Engineering*, pp. 227–235, 2018.
- [17] A. Somov, D. Shadrin, I. Fastovets, A. Nikitin, S. Matveev, I. Seledets, and O. Hrinchuk, "Pervasive Agriculture: IoT-Enabled Greenhouse for Plant Growth Control," *IEEE Pervasive Computing*, vol. 17, no. 4, pp. 65–75, 2018.
- [18] A. Carrasquilla-Batista, A. Chacón-Rodríguez and M. Solórzano-Quintana, "Using IoT resources to enhance the accuracy of overdrain measurements in greenhouse horticulture," *2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI)*, San Jose, pp. 1-5, 2016.
- [19] N. Kitpo, Y. Kugai, M. Inoue, T. Yokemura and S. Satomura, "Internet of Things for Greenhouse Monitoring System Using Deep Learning and Bot Notification Services," *2019 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, pp. 1-4, 2019.
- [20] M. Taki, S. A. Mehdizadeh, A. Rohani, M. Rahnama, and M. Rahmati-Joneidabad, "Applied machine learning in greenhouse simulation; new application and analysis," *Information Processing in Agriculture*, vol. 5, no. 2, pp. 253–268, 2018.
- [21] A. Avotins, J. Gruduls, P. Apse-Apsitis and A. Potapovs, "Crop weight measurement sensor for IoT based industrial greenhouse systems", *Dx.doi.org*, 2018. [Online]. Available: <http://dx.doi.org/10.15159/ar.18.142>. [Accessed: 11- Aug- 2020].
- [22] L. Li, M. Wang, M. Zhang, M. Li, N. Sigrimis, and A. Anastasiou, "Application of IoT Technology in Greenhouse Management," *2012 Dallas, Texas, July 29 - August 1, 2012*, 2012.
- [23] M. Jhuria, A. Kumar, and R. Borse, "Image processing for smart farming: Detection of disease and fruit grading," *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, Shimla, pp. 521-526, 2013.
- [24] M. Bhange and H. Hingoliwala, "Smart Farming: Pomegranate Disease Detection Using Image Processing," *Procedia Computer Science*, vol. 58, pp. 280–288, 2015.
- [25] N. Vljajic and D. Zhou, "IoT as a Land of Opportunity for DDoS Hackers," *Computer*, vol. 51, no. 7, pp. 26–34, 2018.
- [26] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.
- [27] L. Moroney, *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform*. Berkeley, CA: Apress, 2017.
- [28] M. Gargenta, *Learning Android*. Beijing, China: O'Reilly, 2011.
- [29] S. Komatineni, D. MacLean, and S. Y. Hashimi, *Pro Android 3*. New York, US: Springer, 2011.
- [30] P. K. Dixit, *Android*. New Delhi, India: Vikas, 2014.

- [31] S. Rautmare and D. M. Bhalerao, "MySQL and NoSQL database comparison for IoT application," *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, Coimbatore, pp. 235-238, 2016.
- [32] A. Schram and K. M. Anderson, "MySQL to NoSQL," Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity - SPLASH '12, New York, USA, pp. 191–202, 2012.
- [33] C. Györödi, R. Györödi, G. Pecherle and A. Olah, "A comparative study: MongoDB vs. MySQL," *2015 13th International Conference on Engineering of Modern Electric Systems (EMES)*, Oradea, pp. 1-6, 2015.
- [34] S. Malik and D. Kim, "A comparison of RESTful vs. SOAP web services in actuator networks," *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, Milan, pp. 753-755, 2017.
- [35] G. Mulligan and D. Gračanin, "A comparison of SOAP and REST implementations of a service based interaction independence middleware framework," Proceedings of the 2009 Winter Simulation Conference (WSC), Austin, TX, USA, pp. 1423-1432, 2009.
- [36] L. J. Mitchell, *PHP Web Services: APIs for the Modern Web*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2016.
- [37] T. Suzumura, S. Trent, M. Tatsubori, A. Tozawa and T. Onodera, "Performance Comparison of Web Service Engines in PHP, Java and C," *2008 IEEE International Conference on Web Services*, Beijing, pp. 385-392, 2008.
- [38] M. Lachgar, H. Benouda and S. Elfirdoussi, "Android REST APIs: Volley vs Retrofit," *2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, Rabat, Morocco, pp. 1-6, 2018.
- [39] J. Bhade and P. H. Yadav, "Evaluation of Android Networking Libraries," *International Journal of Scientific Research & Engineering Trends*, vol. 5, no. 4, pp. 1374–1377, 2019.
- [40] S. Z. Syed Idrus, E. Cherrier, C. Rosenberger, and J. J. Schwartzmann, "A Review on Authentication Methods," *Australian Journal of Basic and Applied Sciences*, vol. 7, no. 5, pp. 95–107, 2013.
- [41] N. Hong, M. Kim, M.-S. Jun, and J. Kang, "A Study on a JWT-Based User Authentication and API Assessment Scheme Using IMEI in a Smart Home Environment," *Sustainability*, vol. 9, no. 7, p. 1099, 2017.
- [42] S. W. Jung and S. Jung, "Personal OAuth authorization server and push OAuth for Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 13, no. 6, pp. 1–11, 2017.
- [43] H. Polat and S. Oyucu, "Token-based authentication method for M2M platforms," *Turkish Journal Of Electrical Engineering & Computer Sciences*, vol. 25, pp. 2956–2967, 2017.
- [44] J. K. Parmar and A. Desai, "IoT: Networking Technologies and Research Challenges," *International Journal of Computer Applications*, vol. 154, no. 7, pp. 1–6, 2016.
- [45] R. Mahmoud, T. Yousuf, F. Aloul and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, London, pp. 336-341, 2015.
- [46] O. Bello, S. Zeadally, and M. Badra, "Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT)," *Ad Hoc Networks*, vol. 57, pp. 52–62, 2017.

- [47] M. Andersson, "Short range low power wireless devices and Internet of Things (IoT)," *ublox*, 09-Feb-2015. [Online]. Available: <https://www.u-blox.com/en/publication/white-paper/short-range-low-power-wireless-devices-and-internet-things-iot>. [Accessed: 11-Aug-2020].
- [48] B. De, *API Management An Architect's Guide to Developing and Managing APIs for Your Organization*, 1st ed. Berkeley, CA: Apress, 2017
- [49] D. Nevedrov, "Using JMeter to Performance Test Web Services," *dev2dev*, 2016. [Online]. Available: <https://loadstorm.com/files/Using-JMeter-to-Performance-Test-Web-Services.pdf>. [Accessed: 11-Aug-2020].
- [50] E. Sarantaris, "*Autonomous greenhouse for the indoors based on a microcontroller*", Diploma Thesis, School of Electrical and Computer Engineering, Technical University of Crete, Chania, Greece, 2016.
- [51] S. R. Department, "IoT market size worldwide 2017-2025," *Statista*, 19-Feb-2020. [Online]. Available: <https://www.statista.com/statistics/976313/global-iot-market-size/>. [Accessed: 11-Aug-2020].
- [52] "Platform Architecture : Android Developers," *Android Developers*, 05-Jul-2020. [Online]. Available: <https://developer.android.com/guide/platform>. [Accessed: 11-Aug-2020].
- [53] "Android Studio," *Wikipedia*, 07-Aug-2020. [Online]. Available: https://en.wikipedia.org/wiki/Android_Studio. [Accessed: 11-Aug-2020].
- [54] "Mobile Operating System Market Share Worldwide," *StatCounter Global Stats*, Jul-2020. [Online]. Available: <https://gs.statcounter.com/os-market-share/mobile/worldwide>. [Accessed: 11-Aug-2020].
- [55] P. Ozlen, "Life of a message from FCM to the device," *The Firebase Blog*, 26-Feb-2019. [Online]. Available: <https://firebase.googleblog.com/2019/02/life-of-a-message.html>. [Accessed: 11-Aug-2020].
- [56] "Understand the Activity Lifecycle: Android Developers," *Android Developers*, 15-Apr-2020. [Online]. Available: <https://developer.android.com/guide/components/activities/activity-lifecycle>. [Accessed: 11-Aug-2020].
- [57] C. Hunt, *TCP/IP network administration*, 3rd ed. Sebastopol, CA: O'Reilly, 2002.
- [58] B. Biswas, "The Principles of REST," *Medium*, 14-Apr-2019. [Online]. Available: <https://medium.com/future-vision/the-principles-of-rest-6b00deac91b3>. [Accessed: 11-Aug-2020].
- [59] V. Bhagat, "PHP vs Python vs Ruby: Detailed Comparison", *Pixelcrayons*, 01-Jul-2019. [Online]. Available: <https://www.pixelcrayons.com/blog/php-vs-python-vs-ruby-comparison>. [Accessed: 14-Aug-2020].
- [60] A. Bletsas, P.N. Alevizos and G. Vougioukas, "The Art of Signal Processing in Backscatter Radio for μ Watt (or less) Internet-of-Things (IoT)", *IEEE Signal Processing Magazine (SPM)*, Special Issue on Internet-of-Things (IoT), Vol. 35, No. 5, pp. 28-40, 2018.
- [61] G. Vougioukas, A. Dimitriou, A. Bletsas and J.N. Sahalos, "Practical Energy Harvesting for Batteryless Ambient Backscatter Sensors", *Electronics*, Vol. 7, No. 6, 2018.

Εξωτερικοί Σύνδεσμοι

[62] Click & Grow. URL: <https://eu.clickandgrow.com/collections/products>

[63] Plantui. URL: <https://plantui.com/collections/all>

[64] Leaf. URL: <https://www.getleaf.co/#specs>

[65] Niwa. URL: <https://getniwa.com/collections/niwa-product-line>