



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Απομακρυσμένη Παρακολούθηση Ασύρματου Δικτύου  
Αισθητήρων μέσω του Διαδικτύου  
(Με μια εφαρμογή στον κλάδο της μελισσοκομίας)**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

**ΜΑΡΙΑΣ Ζ. ΚΑΡΟΥΤΑ**

**Επιβλέπων :** Ευστάθιος Συκάς  
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2016





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

**Απομακρυσμένη Παρακολούθηση Ασύρματου Δικτύου  
Αισθητήρων μέσω του Διαδικτύου  
(Με μια εφαρμογή στον κλάδο της μελισσοκομίας)**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

της

**ΜΑΡΙΑΣ Ζ. ΚΑΡΟΥΤΑ**

**Επιβλέπων :** Ευστάθιος Συκάς  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25<sup>η</sup> Οκτωβρίου 2016.

.....  
Ευστάθιος Συκάς  
Καθηγητής Ε.Μ.Π.

.....  
Μιχαήλ Θεολόγου  
Καθηγητής Ε.Μ.Π.

.....  
Γεώργιος Στασινόπουλος  
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2016

.....

Μαρία Ζ. Καρούτα

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Μαρία Ζ. Καρούτα, 2016

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## ΠΕΡΙΛΗΨΗ

---

Η παρούσα διπλωματική εργασία έχει σκοπό το σχεδιασμό και την υλοποίηση απομακρυσμένης παρακολούθησης ασύρματου δικτύου αισθητήρων (WSN) μέσω του Διαδικτύου. Ο server της εφαρμογής αναλαμβάνει τη συλλογή των μετρήσεων των αισθητήρων του ασύρματου δικτύου ανά τακτά χρονικά διαστήματα και την αποθήκευσή τους σε βάση δεδομένων. Ο χρήστης έχει τη δυνατότητα μέσω μίας διαδικτυακής εφαρμογής να ενημερώνεται για τις τρέχουσες τιμές των αισθητήρων του ασύρματου δικτύου, αλλά και να ανατρέχει στο ιστορικό των μετρήσεων. Επιπλέον, ο χρήστης, εφόσον το επιθυμεί, μπορεί να λαμβάνει απευθείας ενημερώσεις από την εφαρμογή όταν οι τιμές των αισθητήρων βγουν πέραν των προκαθορισμένων από τον ίδιο ορίων. Η εφαρμογή έχει κατασκευαστεί με τέτοιο τρόπο ώστε να υποστηρίζει περισσότερα από ένα ασύρματα δίκτυα αισθητήρων και χρήστες.

Στα πλαίσια της διπλωματικής εργασίας γίνεται αξιοποίηση ενός ασύρματου δικτύου αισθητήρων WSN που έχει εγκατασταθεί σε κυψέλες μελισσών. Τα μεγέθη που παρακολουθούνται σε κάθε κυψέλη είναι η θερμοκρασία, η υγρασία και το βάρος. Ο μελισσοκόμος μέσω της διαδικτυακής εφαρμογής έχει πλέον τη δυνατότητα να παρακολουθήσει την κατάσταση του μελισσοκομείου του από οποιοδήποτε σημείο του κόσμου, να οργανώσει τις εργασίες του και να βελτιστοποιήσει την παραγωγή του.

**Λέξεις κλειδιά:** Διαδικτυακή εφαρμογή, Ασύρματο δίκτυο αισθητήρων, WSN, GSM, Internet of Things



# ABSTRACT

---

The purpose of the current diploma thesis is the design and implementation of a wireless sensor network (WSN) remote monitoring via the Internet. The application server is responsible to collect the sensor measurements of the wireless sensor network at regular intervals and store them in a database. The user has the ability to be updated to the current values of the sensors of the wireless sensor network as well as refer to the history of measurements using a web application. Moreover, the user, if desired, can receive notifications directly from the application upon specific user specified thresholds. The application is constructed in such a way as to support more than one wireless sensor networks and users.

As part of the diploma thesis, a wireless sensor network (WSN) placed in beehives is used. The monitored variables at each beehive are temperature, humidity and weight. The beekeeper, using the web application, is now able to monitor the current state of the apiary from anywhere in the world, organize the work that has to be done and, as a result, optimize the honey production.

**Keywords:** Web Application, Wireless Sensor Network, WSN, GSM, Internet of Things





## ΕΥΧΑΡΙΣΤΙΕΣ

---

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή της διπλωματικής μου εργασίας, κ. Ευστάθιο Συκά που μου εμπιστεύτηκε την ανάληψη της εργασίας αυτής και μου έδωσε την ευκαιρία να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα. Ιδιαίτερες ευχαριστίες θα ήθελα να εκφράσω προς τον Υποψήφιο Διδάκτορα κ. Απόστολο Κοτοπούλη για τη συνεχή καθοδήγηση και την αδιάκοπη υποστήριξή του καθ' όλη τη διάρκεια εκπόνησης της εργασίας μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για την αγάπη και τη στήριξη που μου παρείχαν σε όλη τη διάρκεια των σπουδών μου.



# ΠΕΡΙΕΧΟΜΕΝΑ

---

ΠΕΡΙΛΗΨΗ.....	5
ABSTRACT.....	7
ΕΥΧΑΡΙΣΤΙΕΣ.....	9
ΠΕΡΙΕΧΟΜΕΝΑ .....	11
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ .....	14
ΚΕΦΑΛΑΙΟ 1 .....	17
ΕΙΣΑΓΩΓΗ.....	17
1.1    Πρόλογος .....	17
1.2    Σκοπός.....	18
1.3    Δομή Διπλωματικής εργασίας .....	19
ΚΕΦΑΛΑΙΟ 2.....	21
Η ΕΦΑΡΜΟΓΗ.....	21
2.1    Εισαγωγή.....	21
2.2    Οι αισθητήρες στην παρακολούθηση μιας κυψέλης.....	22
2.2.1    Αισθητήρας θερμοκρασίας.....	22
2.2.2    Αισθητήρας υγρασίας.....	23
2.2.3    Αισθητήρας βάρους.....	24
2.3    Μετάβαση από το ασύρματο δίκτυο αισθητήρων στην απομακρυσμένη παρακολούθηση από το Διαδίκτυο .....	25
2.4    Διαδικτυακή εφαρμογή .....	26
2.4.1    Πλεονεκτήματα Διαδικτυακών Εφαρμογών .....	27
2.4.2    Μειονεκτήματα Διαδικτυακών Εφαρμογών .....	29
ΚΕΦΑΛΑΙΟ 3.....	31
ΟΡΓΑΝΩΣΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ .....	31
3.1 Το σύστημα βάσεων δεδομένων Microsoft SQL Server.....	31
3.2 Περιγραφή των πινάκων της βάσης δεδομένων.....	32

3.2.1 User.....	33
3.2.2 Beehive.....	34
3.2.3 Measurement.....	35
3.2.4 Settings.....	36
3.3 Σχεσιακό σχήμα της βάσης δεδομένων.....	37
3.4 Μοντέλο Οντοτήτων – Συσχετίσεων (Entity-Relationship E-R model).....	37
ΚΕΦΑΛΑΙΟ 4.....	41
ΣΥΛΛΟΓΗ ΜΕΤΡΗΣΕΩΝ ΑΠΟ ΤΟ ΑΣΥΡΜΑΤΟ ΔΙΚΤΥΟ ΑΙΣΘΗΤΗΡΩΝ.....	41
4.1 Σχεδιαστικές και προγραμματιστικές επιλογές.....	42
4.2 Αναλυτική περιγραφή της διαδικασίας συλλογής μετρήσεων.....	43
4.2.1 Ανάκτηση από τη βάση δεδομένων όλων των ενεργών χρηστών για τους οποίους πρέπει να ληφθούν μετρήσεις.....	44
4.2.2 Λήψη μετρήσεων για κάθε μελισσοκόμο – χρήστη.....	44
ΚΕΦΑΛΑΙΟ 5.....	49
Η ΔΙΑΔΙΚΤΥΑΚΗ ΕΦΑΡΜΟΓΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΤΩΝ ΚΥΨΕΛΩΝ.....	49
5.1 Απαιτήσεις του συστήματος.....	49
5.1.1 Λειτουργικές απαιτήσεις.....	49
5.1.2 Μη λειτουργικές Απαιτήσεις.....	50
5.2 Πρόσβαση στην εφαρμογή.....	50
5.3 Login στην εφαρμογή.....	51
5.4 Κεντρική σελίδα της εφαρμογής.....	54
5.5 Γραφική αναπαράσταση των μετρήσεων.....	58
5.6 Οι ρυθμίσεις του χρήστη.....	67
5.6.1 Notification Settings.....	67
5.6.2 Schedule Settings.....	71
5.7 Το προφίλ του χρήστη.....	73
5.8 Αποσύνδεση από την εφαρμογή.....	75

5.9	Χειρισμός σφαλμάτων στην εφαρμογή.....	76
ΚΕΦΑΛΑΙΟ 6.....		77
Η ΣΥΛΛΟΓΗ ΜΕΤΡΗΣΕΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΕΠΙΠΕΔΟ .....		77
6.1	Γλώσσες προγραμματισμού και πλαίσια .....	77
6.2	Περιγραφή κλάσεων και μεθόδων .....	79
6.2.1	Program.....	79
6.2.2	Server .....	80
6.3	Μηχανισμός καταγραφής συμβάντων .....	81
ΚΕΦΑΛΑΙΟ 7.....		87
ΣΧΕΔΙΑΣΗ ΤΗΣ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ.....		87
7.1	Αρχιτεκτονική εφαρμογής .....	87
7.1.1	Μοντέλο Client-Server .....	87
7.1.2	Web Server.....	88
7.2	Γλώσσες προγραμματισμού και πλαίσια .....	89
7.2.1	Ανάπτυξη πλευράς πελάτη (front-end development) .....	89
7.2.2	Ανάπτυξη πλευράς εξυπηρετητή (back-end development).....	89
7.3	Αρχιτεκτονική σε επίπεδο κώδικα και κλάσεων .....	91
7.3.1	Data Transfer Objects (DTO) .....	91
7.3.2	Models .....	93
7.3.3	Views .....	95
7.3.4	Controllers.....	97
ΚΕΦΑΛΑΙΟ 8.....		101
ΕΠΙΛΟΓΟΣ .....		101
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		103
ΠΑΡΑΡΤΗΜΑ Οδηγίες εγκατάστασης της εφαρμογής .....		105

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

---

Εικόνα 1: Internet of things .....	17
Εικόνα 2: Οι αισθητήρες υγρασίας και θερμοκρασίας μέσα στην κυψέλη.....	23
Εικόνα 3: Ο αισθητήρας βάρους στη βάση της κυψέλης.....	25
Εικόνα 4: Το συνολικό σύστημα της εφαρμογής .....	26
Εικόνα 5: SQL Server Management Studio .....	32
Εικόνα 6: Σχεσιακό σχήμα της βάσης δεδομένων.....	37
Εικόνα 7: Μοντέλο Οντοτήτων Συσχετίσεων της εφαρμογής.....	39
Εικόνα 8: Συνοπτική περιγραφή της διαδικασίας συλλογής μετρήσεων .....	43
Εικόνα 9: GSM πομποδέκτης (SIM 900) .....	45
Εικόνα 10: Οθόνη «Login» της εφαρμογής σε κινητό τηλέφωνο .....	52
Εικόνα 11: Οθόνη «Login» της εφαρμογής σε υπολογιστή .....	52
Εικόνα 12: Η διαδικασία πιστοποίησης ενός χρήστη στην εφαρμογή .....	53
Εικόνα 13: Η κεντρική σελίδα της εφαρμογής σε υπολογιστή .....	54
Εικόνα 14: Η κεντρική σελίδα της εφαρμογής σε κινητό τηλέφωνο .....	55
Εικόνα 15: Το μενού της εφαρμογής σε κινητό τηλέφωνο .....	56
Εικόνα 16: Οθόνη αναμονής απόκρισης του server .....	58
Εικόνα 17: Θερμοκρασία των κυψελών για το διάστημα 04/2016-10/2016 .....	60
Εικόνα 18: Θερμοκρασία μίας κυψέλης (N101) για το διάστημα 04/2016-10/2016 .....	60
Εικόνα 19: Θερμοκρασία των κυψελών για το μήνα Αύγουστο 2016 .....	61
Εικόνα 20: Θερμοκρασία μίας κυψέλης (N100) για το μήνα Αύγουστο 2016.....	61
Εικόνα 21: Θερμοκρασία των κυψελών για το μήνα Σεπτέμβριο 2016 (στιγμιότυπο από κινητό τηλέφωνο).....	62
Εικόνα 22: Υγρασία σε όλες τις κυψέλες για το διάστημα 04/2016-10/2016 .....	62
Εικόνα 23: Υγρασία σε μία κυψέλη (N102) για το διάστημα 04/2016-10/2016 .....	63

Εικόνα 24: Υγρασία σε όλες τις κυψέλες για το μήνα Ιούνιο 2016.....	63
Εικόνα 25: Υγρασία σε μία κυψέλη (N100) για το μήνα Ιούνιο 2016 .....	64
Εικόνα 26: Υγρασία σε όλες τις κυψέλες για το μήνα Οκτώβριο 2016 (στιγμιότυπο από κινητό τηλέφωνο) .....	64
Εικόνα 27: Το βάρος των κυψελών για το διάστημα 04/2016-10/2016.....	65
Εικόνα 28: Το βάρος μιας κυψέλης (N100) για το διάστημα 04/2016-10/2016 .....	65
Εικόνα 29: Το βάρος των κυψελών για το μήνα Μάιο 2016.....	66
Εικόνα 30: Το βάρος μιας κυψέλης (N101) για το μήνα Μάιο 2016 .....	66
Εικόνα 31: Το βάρος των κυψελών για το μήνα Μάιο 2016 (στιγμιότυπο από κινητό τηλέφωνο).....	67
Εικόνα 32: Στιγμιότυπο οθόνης για τα Notification Settings σε υπολογιστή .....	68
Εικόνα 33: Στιγμιότυπο οθόνης για τα Notification Settings σε κινητό τηλέφωνο ....	69
Εικόνα 34: Στιγμιότυπο οθόνης κατά την επεξεργασία μίας ρύθμισης.....	70
Εικόνα 35: Στιγμιότυπο οθόνης για τα Schedule Settings σε κινητό τηλέφωνο.....	71
Εικόνα 36: Στιγμιότυπο οθόνης για τα Schedule Settings σε υπολογιστή.....	72
Εικόνα 37: Η οθόνη «Profile» της εφαρμογής σε κινητό τηλέφωνο.....	73
Εικόνα 38: Η οθόνη «Profile» της εφαρμογής σε υπολογιστή.....	74
Εικόνα 39: Η οθόνη επεξεργασίας στοιχείων του χρήστη .....	75
Εικόνα 40: Λανθασμένη εισαγωγή στοιχείων από το χρήστη .....	76
Εικόνα 41: Οθόνη εμφάνισης σφάλματος.....	76
Εικόνα 42: Το περιβάλλον προγραμματισμού Microsoft Visual Studio 2015 Community Edition .....	78
Εικόνα 43: Entity Framework.....	79
Εικόνα 44: Μοντέλο Client – Server .....	87
Εικόνα 45: Internet Information Services (IIS) Manager .....	88
Εικόνα 46: Μοντέλο Model – View – Controller .....	90
Εικόνα 47: Οργάνωση του project σε φακέλους στο Visual Studio .....	91





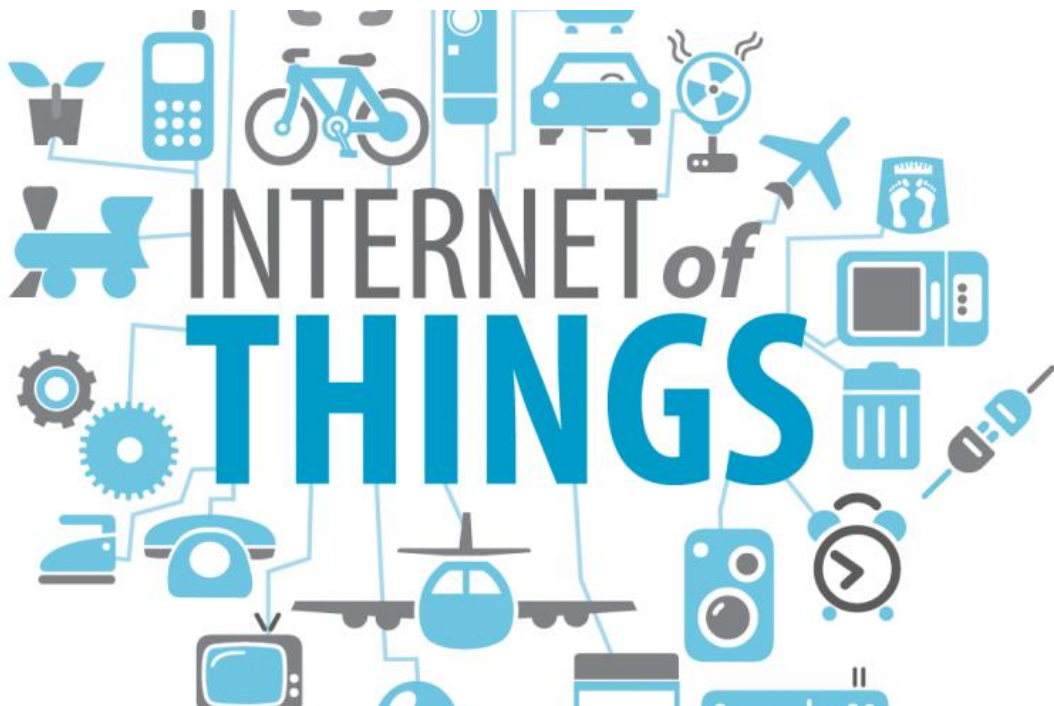
# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ

---

### 1.1 Πρόλογος

Το Internet of Things αναπτύσσεται με ταχύτατους ρυθμούς τα τελευταία χρόνια και εξαπλώνεται σε ολοένα και περισσότερα αντικείμενα της καθημερινής ζωής. Ο κόσμος του Internet of Things περιλαμβάνει αντικείμενα με δυνατότητα σύνδεσης στο Διαδίκτυο. Πάσης φύσεως αντικείμενα και αισθητήρες, όπως θερμοστάτες και οικιακές συσκευές έχουν τη δυνατότητα να ανταλλάσσουν πληροφορίες και να ελέγχονται μέσω του Διαδικτύου. Η ζωή των ανθρώπων γίνεται πολύ πιο εύκολη αφού μπορούν να ρυθμίσουν την κατάσταση ενός αντικειμένου τους ή να ελέγξουν τις συνθήκες που επικρατούν σε ένα χώρο όπου και αν βρίσκονται, χρησιμοποιώντας απλά το κινητό τους τηλέφωνο το οποίο διαθέτει δυνατότητα σύνδεσης στο Internet.



Εικόνα 1: Internet of things

## 1.2 Σκοπός

Η παρούσα διπλωματική εργασία έχει σκοπό το σχεδιασμό και την υλοποίηση απομακρυσμένης παρακολούθησης ενός ασύρματου δικτύου αισθητήρων μέσω του Διαδικτύου.

Στα πλαίσια της διπλωματικής, γίνεται αξιοποίηση ενός ασύρματου δικτύου αισθητήρων WSN που έχει εγκατασταθεί σε κυψέλες μελισσών. Το συγκεκριμένο WSN αποτελεί προϊόν μιας προηγούμενης διπλωματικής εργασίας που ολοκληρώθηκε στο προηγούμενο έτος με τίτλο «Δίκτυο ασύρματων αισθητήρων για απομακρυσμένη παρακολούθηση (Με μια εφαρμογή στο κλάδο της μελισσοκομίας)». Στην προηγούμενη διπλωματική ο μελισσοκόμος είχε τη δυνατότητα να βλέπει την τρέχουσα κατάσταση των κυψελών (θερμοκρασία, υγρασία, βάρος κλπ.) μέσω SMS από το κινητό του τηλέφωνο. Η παρούσα διπλωματική εργασία αξιοποιώντας το hardware της προηγούμενης διπλωματικής όσον αφορά τη συλλογή των μετρήσεων, προσθέτει το απαραίτητο λογισμικό και hardware έτσι ώστε να δώσει επιπλέον μία ολοκληρωμένη διαδικτυακή παρακολούθηση. Ο μελισσοκόμος θα έχει πλέον τη δυνατότητα με την παρούσα διπλωματική, μέσω μιας εφαρμογής client-server να ενημερώνεται διαδικτυακά όχι μόνο για την παρούσα κατάσταση των κυψελών αλλά και να ανατρέχει στο ιστορικό των μετρήσεων, το οποίο είναι διαθέσιμο μέσω βάσης δεδομένων που υπάρχει στο server μηχανήμα για περεταίρω ανάλυση και βελτιστοποίηση της παραγωγής. Ο μελισσοκόμος μπορεί να τρέξει την εφαρμογή είτε στον υπολογιστή του είτε στο κινητό του τηλέφωνο, αρκεί να υπάρχει διαδικτυακή σύνδεση (π.χ. ADSL ή 3G αντίστοιχα).

Ο απομακρυσμένος έλεγχος στα μελίσσια κρίνεται ιδιαίτερα σημαντικός και χρήσιμος διότι βρίσκονται τοποθετημένα εκτός πόλης σε αγροτικές εκτάσεις και συνεπώς δεν είναι βολικό για το μελισσοκόμο να μεταβαίνει συνεχώς εκεί για να ελέγξει τις συνθήκες που επικρατούν. Έτσι όχι μόνο μειώνονται τα έξοδα μετακίνησης για μετάβαση στο χώρο των κυψελών, αλλά βελτιστοποιείται και ο χρόνος απόκρισης σε ενδεχόμενα προβλήματα. Αποδεικνύεται δε ιδιαίτερα πολύτιμος ο απομακρυσμένος έλεγχος σε περιπτώσεις πρόληψης ασθενειών και άλλων έκτακτων συμβάντων. Επίσης ο μελισσοκόμος διευκολύνεται στη συγκομιδή, αφού μπορεί να γνωρίζει ή ακόμα και να υπολογίζει την καταλληλότερη στιγμή για να την πραγματοποιήσει.

Στόχος της εργασίας είναι η λήψη των μετρήσεων των αισθητήρων κάθε κυψέλης ανά τακτά χρονικά διαστήματα αλλά και οποιαδήποτε στιγμή ζητηθεί από το μελισσοκόμο, η αποθήκευσή τους και η παρουσίασή τους στο μελισσοκόμο μέσω μιας φιλικής προς το χρήστη web εφαρμογής. Επιπλέον, η εξασφάλιση της δυνατότητας απευθείας ενημέρωσης του μελισσοκόμου σε επείγουσες καταστάσεις.

### **1.3 Δομή Διπλωματικής εργασίας**

Η εργασία αυτή αποτελείται από οκτώ κεφάλαια:

#### **Κεφάλαιο 1: Εισαγωγή**

Πρόκειται για το παρόν κεφάλαιο στο οποίο γίνεται μια ευρεία περίληψη της εργασίας που περιλαμβάνει το αντικείμενο και το σκοπό της, καθώς και μια σύντομη αναφορά στο περιεχόμενο κάθε κεφαλαίου.

#### **Κεφάλαιο 2: Η εφαρμογή**

Στο κεφάλαιο αυτό περιγράφονται τα βασικά σημεία και η χρησιμότητα της εφαρμογής που κατασκευάστηκε στην προηγούμενη διπλωματική εργασία καθώς και η σύνδεσή της με την παρούσα διπλωματική. Τέλος, γίνεται μία ανάλυση των διαδικτυακών εφαρμογών ως προς τα πλεονεκτήματα και τα μειονεκτήματά τους.

#### **Κεφάλαιο 3: Οργάνωση των δεδομένων της εφαρμογής**

Εδώ παρουσιάζεται αναλυτικά ο τρόπος που οργανώθηκαν τα δεδομένα που διαχειρίζεται η εφαρμογή σε βάση δεδομένων.

#### **Κεφάλαιο 4: Συλλογή μετρήσεων από το ασύρματο δίκτυο αισθητήρων**

Στο κεφάλαιο αυτό περιγράφεται λεπτομερώς η διαδικασία την οποία ακολουθεί ο server προκειμένου να πάρει τις μετρήσεις που συλλέγουν οι κεντρικοί κόμβοι του ασύρματου δικτύου αισθητήρων που έχει εγκατασταθεί σε κυψέλες μελισσών.

#### **Κεφάλαιο 5: Η διαδικτυακή εφαρμογή παρακολούθησης των κυψελών**

Στο κεφάλαιο αυτό περιγράφεται αναλυτικά η διαδικτυακή εφαρμογή που κατασκευάστηκε, μέσα από την οποία οι μελισσοκόμοι – χρήστες έχουν τη δυνατότητα να ελέγξουν την τρέχουσα κατάσταση των κυψελών τους, αλλά και να ανατρέξουν στο ιστορικό παλαιότερων μετρήσεων των αισθητήρων.

## **Κεφάλαιο 6: Η συλλογή μετρήσεων σε προγραμματιστικό επίπεδο**

Εδώ πραγματοποιείται μια αναφορά στα εργαλεία και τα πλαίσια που χρησιμοποιήθηκαν κατά την ανάπτυξη του προγράμματος που αφορά τη συλλογή μετρήσεων από το server. Ακόμα, αναλύεται η δομή και τα συστατικά στοιχεία του κώδικα.

## **Κεφάλαιο 7: Σχεδίαση της διαδικτυακής εφαρμογής**

Στο κεφάλαιο αυτό παρουσιάζονται οι προγραμματιστικές γλώσσες και τα πλαίσια που χρησιμοποιήθηκαν κατά την ανάπτυξη της διαδικτυακής εφαρμογής, ενώ ταυτόχρονα αναλύεται η αρχιτεκτονική της σε επίπεδο κλάσεων και επιλεγμένου προγραμματιστικού μοντέλου.

## **Κεφάλαιο 8: Επίλογος**

Εδώ παρουσιάζονται ορισμένες εφαρμογές των ασύρματων δικτύων αισθητήρων στις οποίες θα μπορούσε να αξιοποιηθεί το λογισμικό που κατασκευάστηκε στην παρούσα διπλωματική εργασία.

# ΚΕΦΑΛΑΙΟ 2

## Η ΕΦΑΡΜΟΓΗ

---

Η παρούσα διπλωματική εργασία πραγματεύεται την απομακρυσμένη παρακολούθηση ασύρματου δικτύου αισθητήρων σε κυψέλες μελισσών μέσω του Διαδικτύου. Το ασύρματο δίκτυο αισθητήρων που χρησιμοποιήθηκε αποτελεί προϊόν της διπλωματικής εργασίας «Δίκτυο ασύρματων αισθητήρων για απομακρυσμένη παρακολούθηση (Με μια εφαρμογή στο κλάδο της μελισσοκομίας)». Σε αυτό το κεφάλαιο περιγράφονται τα βασικά σημεία και η χρησιμότητα της εφαρμογής που κατασκευάστηκε στην προηγούμενη εργασία καθώς και η σύνδεσή της με την παρούσα διπλωματική. Τέλος, στα πλαίσια της διαδικτυακής εφαρμογής που κατασκευάστηκε σε αυτήν την εργασία, αναλύονται τα πλεονεκτήματα και τα μειονεκτήματα των διαδικτυακών εφαρμογών.

### ***2.1 Εισαγωγή***

Η εφαρμογή προσανατολίζεται στην κάλυψη διαφόρων αναγκών ενός μελισσοκόμου. Η ανάγκη για απομακρυσμένη παρακολούθηση του μικροκλίματος των κυψελών, κατά τη διάρκεια όλου του έτους, παρουσιάζει ιδιαίτερο ενδιαφέρον για το μελισσοκόμο που ασκεί νομαδική μελισσοκομία. Το χειμώνα, που παρατηρούνται χαμηλές θερμοκρασίες, τα αποθέματα τροφής που βρίσκονται μέσα σε μία κυψέλη μειώνονται. Το μέλι μετά από μια παρατεταμένη τέτοια περίοδο κινδυνεύει να καταρρεύσει. Εκεί είναι που πρέπει να επέμβει ο μελισσοκόμος και με κατάλληλους χειρισμούς να βοηθήσει στην συντήρηση της κυψέλης. Με τη χρήση, λοιπόν, αισθητήρων και της ασύρματης τεχνολογίας μπορεί να ενημερώνεται ο μελισσοκόμος ανά πάσα στιγμή για την κατάσταση των μελισσών.

Το ασύρματο δίκτυο αισθητήρων ταιριάζει απόλυτα στις ανάγκες απομακρυσμένης παρακολούθησης κυψελών, καθώς ενσωματώνει όλους τους απαραίτητους αισθητήρες για τη σωστή παρακολούθηση μιας κυψέλης (αισθητήρες θερμοκρασίας, υγρασίας και βάρους). Η χρήση της ασύρματης τεχνολογίας των WSN στα 2.4GHz βοηθά στην παρακολούθηση όλου του μελισσοκομείου με πολύ μικρό κόστος. Με εμβέλεια επικοινωνίας 60m μπορεί να καλύψει μια περιοχή 3600m<sup>2</sup>. Ακόμα, με τη σύνδεση ενός GSM πομποδέκτη στον Gateway που συλλέγει τις μετρήσεις από τους αισθητήρες των κυψελών προσφέρεται η δυνατότητα στο μελισσοκόμο της απομακρυσμένης παρακολούθησης του μελισσοκομείου του από οποιοδήποτε σημείο του κόσμου. Ο μελισσοκόμος είναι πλέον σίγουρος για την πορεία και την κατάσταση των μελισσοσμηνών του. Μπορεί να προγραμματίζει τις εργασίες του χωρίς να χάνει χρόνο αλλά και χρήμα σε άσκοπες μετακινήσεις. Τέλος, με την άμεση ενημέρωσή του στις περιόδους μελιτοεκκρίσεων μπορεί να παίρνει σημαντικές αποφάσεις για την αύξηση των κερδών του.

## ***2.2 Οι αισθητήρες στην παρακολούθηση μιας κυψέλης***

Τοποθετώντας τους αισθητήρες στην κυψέλη μπορούμε να πάρουμε άμεσα πληροφορίες για το βάρος, την υγρασία και την θερμοκρασία που επικρατούν. Οι πληροφορίες αυτές, σε συνδυασμό με την εποχή και την εμπειρία του μελισσοκόμου, δίνουν μια πλήρη εικόνα της κατάστασης των μελισσοσμηνών. Παρακάτω παρουσιάζεται αναλυτικά η χρησιμότητα του κάθε αισθητήρα καθώς και ο τρόπος τοποθέτησής του στην κυψέλη.

### ***2.2.1 Αισθητήρας θερμοκρασίας***

Τοποθετώντας έναν αισθητήρα θερμοκρασίας μέσα στην κυψέλη μπορούμε να έχουμε τα παρακάτω συμπεράσματα:

- Το χειμώνα, που δεν έχουμε γόνο, παρατηρούμε τη θερμοκρασία που διατηρεί το μελίσσι και πόσο μπορεί να δυσκολεύεται σε σχέση με πιθανές ακραίες συνθήκες περιβάλλοντος.

- Όταν, στην αρχή της άνοιξης ή και νωρίτερα, δούμε τη θερμοκρασία να ανεβαίνει ξαφνικά στους 35 βαθμούς, σημαίνει ότι το μελίσσι άρχισε να έχει γόνο. Αυτό δείχνει ότι ξεκινάει η περίοδος ανάπτυξης του μελισσιού και η κατανάλωση τροφής μεγαλώνει πολύ. Οπότε, ο μελισσοκόμος, μόλις ο καιρός το επιτρέψει, πρέπει να ελέγξει τα αποθέματα τροφής σε όλα τα μελίσσια, γιατί αυτή η περίοδος είναι η πιο επικίνδυνη για να καταρρεύσει ένα μελίσσι από έλλειψη τροφής. Αυτό συμβαίνει διότι η κατανάλωση τροφής είναι μεγαλύτερη από την ποσότητα που μπορεί να συλλέξει και τα αποθέματα τροφής του χειμώνα μπορεί να έχουν τελειώσει.

Με αυτόν τον τρόπο, επιτυγχάνεται να έχουν τα μελίσσια τροφή για να ξεκινήσουν άμεσα την ανάπτυξη τους, χωρίς κίνδυνο λιμοκτονίας και αποφεύγοντας τους περιττούς ελέγχους αυτήν την περίοδο.

### **2.2.2 Αισθητήρας υγρασίας**

Η υγρασία μέσα στην κυψέλη αποτελεί έναν από τους μεγαλύτερους κινδύνους των μελισσιών και η καταγραφή της είναι σημαντική. Παρά το γεγονός πως η υγρασία είναι απαραίτητη για το μελίσσι και συγκεκριμένα για το γόνο του, όταν η τιμή της ξεπεράσει κάποια όρια, τότε αρχίζει και γίνεται επικίνδυνη για τις μέλισσες. Πράγματι, αυτή η ίδια η υγρασία δεν μπορεί να βλάψει ιδιαίτερα τις μέλισσες, εκτός και αν συνδυαστεί με χαμηλές θερμοκρασίες. Τότε μπορεί να ευνοήσει πολύ την ανάπτυξη κάποιων θανάσιμων εχθρών τους – εννοούμε κυρίως τη νέα Νοζεμίαση.



**Εικόνα 2: Οι αισθητήρες υγρασίας και θερμοκρασίας μέσα στην κυψέλη**

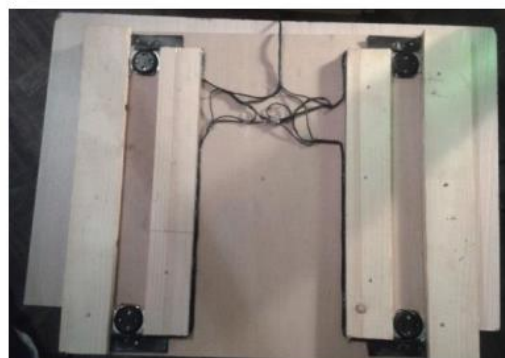
Μέσα στο χώρο της κυψέλης, όπου η θερμοκρασία συνήθως είναι μεγαλύτερη από αυτήν του περιβάλλοντος, δημιουργούνται συνθήκες υψηλής σχετικής υγρασίας. Το φαινόμενο αυτό «επιβαρύνεται» ακόμα περισσότερο και από δύο ακόμα παράγοντες. Ο ένας έχει να κάνει με την παραγωγή διοξειδίου του άνθρακα από τις μέλισσες, ιδιαίτερα όταν στρεσάρονται και ο δεύτερος με την αποβολή υγρασίας από το «υδρόφιλο» μέλι, όταν υπάρχουν ασφράγιστα κελιά.

### **2.2.3 Αισθητήρας βάρους**

Το πιο σημαντικό για να έχουμε ένα υγιές μελίσσι είναι να έχει επάρκεια τροφής ανά πάσα στιγμή. Ακόμη και ένα πολύ μικρό διάστημα να μείνει χωρίς τροφή γίνεται ευάλωτο σε ασθένειες με άμεσες αλλά και μακροπρόθεσμες συνέπειες. Οι μέλισσες είναι γενικά πιο βραχύβιες όταν μεγαλώνουν σε μελίσσια που πάσχουν από υποσιτισμό. Επίσης, διάφορα νοσήματα μπορεί να είναι αίτια του υποσιτισμού. Από την άλλη πλευρά, η χορήγηση τροφής είναι ένα από τα βασικά έξοδα του μελισσοκόμου, γι' αυτό δεν πρέπει να γίνονται υπερβολές και άσκοπα ταΐσματα. Με τον αισθητήρα βάρους παρατηρούμε τις αυξομειώσεις του βάρους μιας κυψέλης σε όλη τη διάρκεια του χρόνου αποκομίζοντας σημαντικά συμπεράσματα. Επίσης, οι βασικές ανθοφορίες είναι σε συγκεκριμένα χρονικά διαστήματα. Για παράδειγμα το έλατο, στην Ελλάδα δίνει μέλι από τέλος Μαΐου (ανάλογα με την χρονιά) και κρατάει ένα μήνα, όμως αν αρχίσουν βροχές (κάτι που είναι πολύ πιθανό αυτήν την περίοδο) μπορεί ακόμα και να το ξεπλύνει τελείως από μέλι. Άρα καταλαβαίνουμε ότι όλα τα μέρη με έλατο δεν δίνουν την ίδια ποσότητα μελιού. Οπότε, αν το μέρος που έχουν μεταφερθεί τα μελίσσια δεν αποδίδει καλά, ο μελισσοκόμος πρέπει να πάρει ένα ακόμη μεγαλύτερο ρίσκο, να ξανακάνει μεταφορά για να πάει σε άλλο μέρος που δείχνει να αποδίδει τη συγκεκριμένη χρονιά (π.χ. να τα πάρει από το Μαίναλο να τα πάει στον Ταΰγετο) μήπως προλάβει τη μελιτοέκκριση στην μέση. Οι συχνές μεταφορές φυσικά επιβαρύνουν και στρεσάρουν τα μελίσσια και επίσης μεγαλώνουν το κόστος, τον κόπο και αλλά και το άγχος του μελισσοκόμου. Από την άλλη πλευρά όμως το να μείνουν παραγωγικά μελίσσια σε ένα μέρος χωρίς τροφή μπορεί να έχει καταστροφικές συνέπειες, οπότε αναγκαστικά χρειάζεται να ταΐστουν με μεγάλη ποσότητα τροφής, μέχρι οι συνθήκες να επιτρέψουν ξανά στα μελίσσια να μαζέψουν μέλι.



Τέλος, με τον αισθητήρα βάρους μπορεί να εντοπίσει πότε η ανθοφορία φτάνει στο τέλος της και να αποσύρει τα μελίσσια εγκαίρως πριν αρχίσουν οι λεηλασίες ανάμεσα σε μελίσσια από γύρω μελισσοκομεία που βρίσκονται μαζεμένα στην περιοχή την συγκεκριμένη περίοδο. Επίσης, πιθανόν ο μελισσοκόμος να προλάβει να μεταφέρει τα μελίσσια σε μια άλλη περιοχή που γνωρίζει ότι δίνει μέλι, έχοντας εγκαταστήσει και εκεί ένα μελίσσι <δείκτη>.



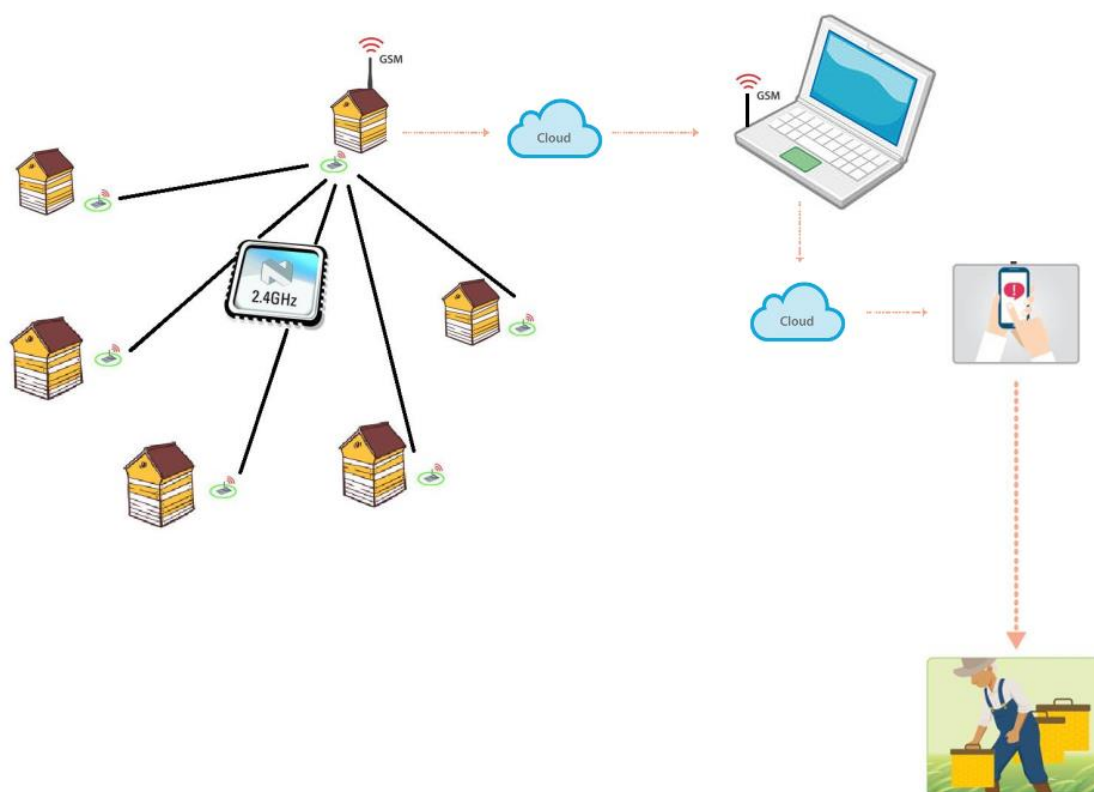
Εικόνα 3: Ο αισθητήρας βάρους στη βάση της κυψέλης.

Οι παράγραφοι 2.1 και 2.2 αναφέρονται και περιγράφουν υλικό που υλοποιήθηκε στην προηγούμενη διπλωματική «Δίκτυο ασύρματων αισθητήρων για απομακρυσμένη παρακολούθηση (Με μια εφαρμογή στο κλάδο της μελισσοκομίας)» [1]

### ***2.3 Μετάβαση από το ασύρματο δίκτυο αισθητήρων στην απομακρυσμένη παρακολούθηση από το Διαδίκτυο***

Ο Gateway συγκεντρώνει τις μετρήσεις από όλες τις κυψέλες που συνδέονται στο ασύρματο δίκτυο αισθητήρων. Διαθέτει έναν GSM πομποδέκτη (SIM 900), ο οποίος όταν λάβει κλήση από το server (η οποία πραγματοποιείται χρησιμοποιώντας ένα SIM 900 που βρίσκεται συνδεδεμένο στο server), ενεργοποιεί τον Gateway που βρίσκεται εν υπνώσει (sleep mode) και στέλνει μέσω σύνδεσης 3G τις μετρήσεις προς τον απομακρυσμένο server. Ο server αποθηκεύει τις μετρήσεις αυτές στη βάση δεδομένων του.

Ο μελισσοκόμος μπορεί να δει τις τελευταίες μετρήσεις που ελήφθησαν από τις κυψέλες του, αλλά και να ανατρέξει στο ιστορικό των μετρήσεων μέσω της διαδικτυακής μας εφαρμογής. Ακόμα, μέσω αυτής της διαδικτυακής εφαρμογής, μπορεί να ζητήσει τις τρέχουσες μετρήσεις των αισθητήρων των κυψελών ανά πάσα στιγμή και να ορίσει κανόνες ώστε να ενημερώνεται αυτόματα μέσω email/SMS όταν επικρατούν συγκεκριμένες συνθήκες στο μελίσσι του.



Εικόνα 4: Το συνολικό σύστημα της εφαρμογής

## 2.4 Διαδικτυακή εφαρμογή

Διαδικτυακή εφαρμογή (Web Application ή Web App) ονομάζεται κάθε εφαρμογή η οποία είναι διαθέσιμη στους χρήστες της μέσω του Διαδικτύου (Internet) ή του ενδοδικτύου (Intranet) μιας εταιρίας. Ο χρήστης χρησιμοποιεί μία διαδικτυακή εφαρμογή μόνο μέσω του περιηγητή του. Οι εφαρμογές αυτές συνήθως εκτελούνται σε ισχυρές υπολογιστικές μηχανές οι οποίες έχουν τον ρόλο του σταθμού εξυπηρέτησης και παρέχουν τις υπηρεσίες τους σε περισσότερους του ενός χρήστη.

Οι τοπικές εφαρμογές υπολογιστών παραμένουν μέχρι και σήμερα αρκετά δημοφιλείς και αυτό οφείλεται εν μέρει στο ότι οι τοπικές εφαρμογές ήταν και οι πρώτες εφαρμογές που δημιουργήθηκαν. Με την πάροδο του χρόνου όμως όλο και περισσότερες διαδικτυακές εφαρμογές κάνουν την εμφάνιση τους. Συγκρίνοντας τις δυο κατηγορίες εφαρμογών οι χρήστες θα εντοπίσουν τόσο θετικά όσο και αρνητικά στοιχεία και στις δυο και σε συνάρτηση με τις εργασίες που θέλουν η εφαρμογή τους να πραγματοποιεί καλούνται να αποφασίσουν ποια από τις δύο κατηγορίες είναι αυτή που τους ταιριάζει.

#### ***2.4.1 Πλεονεκτήματα Διαδικτυακών Εφαρμογών***

Τα βασικά πλεονεκτήματα των διαδικτυακών εφαρμογών είναι τα παρακάτω:

- Άμεση πρόσβαση από οποιαδήποτε συσκευή: Οι χρήστες των διαδικτυακών εφαρμογών έχουν άμεση προσβασιμότητα στις εφαρμογές που θέλουν να χρησιμοποιήσουν από οποιονδήποτε υπολογιστή ή άλλη συσκευή έχει Internet χωρίς την εγκατάσταση κάποιου επιπρόσθετου λογισμικού. Η μόνη απαραίτητη εφαρμογή είναι ο περιηγητής διαδικτύου ο οποίος είναι προεγκατεστημένος σε όλα τα λειτουργικά συστήματα ακόμα και στις φορητές συσκευές και στα κινητά τηλέφωνα. Η ιδιότητα αυτή των web εφαρμογών είναι σημαντική για μεγάλες επιχειρήσεις με πολλούς χρήστες, αφού στην περίπτωση της τοπικής εφαρμογής θα έπρεπε να εγκατασταθεί η εφαρμογή σε κάθε υπολογιστή ξεχωριστά.
- Δυνατότητα χρήσης ανεξαρτήτου τοποθεσίας: Ως συνέχεια του παραπάνω οι χρήστες των διαδικτυακών εφαρμογών μπορούν να τις χρησιμοποιούν ακόμα και αν δε βρίσκονται στο χώρο εργασίας τους. Η δυνατότητα αυτή δίνει ευελιξία στους χρήστες ώστε να χρησιμοποιούν τις εφαρμογές οπουδήποτε αυτοί επιθυμούν επιτρέποντας τους ακόμα και να εργάζονται από απομακρυσμένες περιοχές ή και από το σπίτι τους.
- Συμβατές με όλα τα λειτουργικά συστήματα: Ένα ακόμα πλεονέκτημα των διαδικτυακών εφαρμογών είναι ότι είναι συμβατές με όλα τα λειτουργικά συστήματα. Δεδομένου ότι η εφαρμογή εκτελείται μέσω του περιηγητή του διαδικτύου και όχι στον υπολογιστή του χρήστη, καθίσταται ικανή να εκτελείται σε όλα τα λειτουργικά συστήματα. Η ιδιότητα αυτή οφείλεται επίσης και στην

προτυποποίηση των γλωσσών προγραμματισμού τις οποίες χρησιμοποιεί η εφαρμογή.

- Δεν καταναλώνουν πόρους: Ως συνέχεια του παραπάνω και εφόσον οι διαδικτυακές εφαρμογές δεν εκτελούνται στον υπολογιστή του χρήστη, δεν καταναλώνουν και πόρους από το σύστημα. Για το λόγο αυτό οι εφαρμογές διαδικτύου είναι ιδιαίτερα ελαφριές για την υπολογιστική μονάδα.
- Δεν καταλαμβάνουν χώρο: Ακολουθώντας την ίδια λογική με νωρίτερα οι εφαρμογές αυτές δεν καταλαμβάνουν καθόλου ή σχεδόν καθόλου χώρο στον δίσκο του χρήστη αφού το σύνολο της εφαρμογής είναι αποθηκευμένο στον εξυπηρετητή και μόνο κατά τη χρήση της εφαρμογής μπορεί να υπάρχει μεταφορά δεδομένων προς την υπολογιστική μονάδα του χρήστη και μόνο στην περίπτωση που ο χρήστης το επιθυμεί.
- Γρήγορη αναβάθμιση: Σημαντικό πλεονέκτημα συγκριτικά με τις τοπικές εφαρμογές εμφανίζεται στις περιπτώσεις που η εφαρμογή χρειάζεται κάποια αναβάθμιση. Σε μια κλασική τοπική εφαρμογή η αναβάθμιση του συστήματος θα πρέπει να γίνει σε κάθε ένα υπολογιστή ξεχωριστά πράγμα που απαιτεί χρόνο και χρήμα. Αντίθετα σε μια διαδικτυακή εφαρμογή η αναβάθμιση πραγματοποιείται μόνο στον εξυπηρετητή που φιλοξενεί την εφαρμογή και ταυτόχρονα το αναβαθμισμένο πρόγραμμα είναι διαθέσιμο σε όλους τους χρήστες. Με τον τρόπο αυτό εξοικονομείται χρόνος ο οποίος είναι ιδιαίτερα πολύτιμος κυρίως για τις μεγάλες επιχειρήσεις ενώ ως συνέπεια του παραπάνω σημαντικά μειωμένο είναι και το κόστος της αναβάθμισης μιας και απαιτείται λιγότερο εργατικό δυναμικό για τη διεκπεραίωση της αναβάθμισης.
- Νέο, βελτιωμένο περιβάλλον: Ένα ακόμα πλεονέκτημα των διαδικτυακών εφαρμογών είναι ότι πλέον με την εμφάνιση της HTML5 είναι δυνατό ο δημιουργός της εφαρμογής να την εμπλουτίσει έτσι ώστε να είναι πιο φιλική, εύχρηστη και ευχάριστη προς τον χρήστη με εύκολο τρόπο. Παλαιότερα οι εφαρμογές αυτές υστερούσαν στην εμφάνιση, ωστόσο πλέον είναι ιδιαίτερα εύκολο να εμπλουτιστούν.
- Δυνατότητα χρήσης και εκτός διαδικτύου - ενδοδικτύου: Ένα ακόμα πλεονέκτημα των σύγχρονων διαδικτυακών εφαρμογών (εφαρμογές με χρήση HTML5) είναι η δυνατότητα της εκτός διαδικτύου χρήσης μιας διαδικτυακής εφαρμογής με την προϋπόθεση ότι η εφαρμογή έχει κατασκευαστεί με ανάλογο τρόπο.

Για παράδειγμα αν για κάποιο λόγο η σύνδεση στο διαδίκτυο διακοπεί, αυτό δεν επηρεάζει το χρήστη ο οποίος συνεχίζει να χρησιμοποιεί την εφαρμογή κανονικά. Αυτό επιτυγχάνεται από τον περιηγητή ο οποίος κρατάει ένα αντίγραφο από τα αρχεία τα οποία είναι απαραίτητα για την εκτός δικτύου χρήση της εφαρμογής, στον υπολογιστή του χρήστη, και τα χρησιμοποιεί όταν αυτό κριθεί απαραίτητο. Η συγκεκριμένη δυνατότητα δεν είναι διαθέσιμη σε όλες τις εφαρμογές που χρησιμοποιούν HTML5 αλλά μόνο σε αυτές που έχει υπάρξει πρόβλεψη για χρήση της εφαρμογής και εκτός διαδικτύου ή ενδοδικτύου.

#### ***2.4.2 Μειονεκτήματα Διαδικτυακών Εφαρμογών***

Τα σημαντικότερα μειονεκτήματα των διαδικτυακών εφαρμογών είναι τα παρακάτω:

- Χρήση της εφαρμογής εκτός διαδικτύου: Ένα μειονέκτημα που ταυτόχρονα είναι και πλεονέκτημα είναι η χρήση της εφαρμογής εκτός διαδικτύου. Προκειμένου να γίνει αυτό εφικτό θα πρέπει να έχει γίνει πρόβλεψη κατά την σχεδίαση της εφαρμογής και να έχουν ληφθεί τα κατάλληλα μέτρα. Σε περίπτωση που κάτι τέτοιο δεν έχει γίνει, η εφαρμογή δεν είναι δυνατό να χρησιμοποιηθεί χωρίς την σύνδεση του χρήστη με το Διαδίκτυο ή το ενδοδίκτυο της εταιρίας.
- Αδυναμία χρήσης χωρίς σύνδεση στο διαδίκτυο: Σαν συνέχεια του παραπάνω, εφαρμογές που δεν έχουν κατασκευαστεί με χρήση της τελευταίας έκδοσης της HTML, δηλαδή παλαιότερες εφαρμογές, δεν είναι δυνατόν να χρησιμοποιηθούν αν δεν υπάρχει σύνδεση με το διαδίκτυο ή το ενδοδίκτυο. Το παραπάνω αποτελεί και το κύριο μειονέκτημα των παλαιότερων διαδικτυακών εφαρμογών καθώς αν για κάποιο λόγο η σύνδεση του χρήστη διακοπεί τότε αυτός δεν μπορεί να χρησιμοποιήσει καθόλου την εφαρμογή. Σε περίπτωση που μια παλαιότερη διαδικτυακή εφαρμογή θελήσει να αυξήσει τις δυνατότητες της προκειμένου να μπορεί να λειτουργεί και εκτός διαδικτύου τότε αυτή θα πρέπει να ξανακατασκευαστεί σε μεγάλο μέρος της από την αρχή.
- Μη πλήρης συμβατότητα των περιηγητών: Ένα ακόμα μειονέκτημα που αφορά την τελευταία έκδοση της HTML είναι η μη πλήρης συμβατότητα των περιηγητών με την έκδοση αυτή. Αν και τα πλεονεκτήματα και οι δυνατότητες της HTML5 είναι πολλά, αρκετοί από τους περιηγητές δεν είναι ακόμα πλήρως συμβατοί με αυτά.

Έτσι δεν γίνεται πλήρη χρήση των δυνατοτήτων αυτών πράγμα που περιορίζει τους προγραμματιστές που έχουν αναλάβει ένα έργο. Επίσης, σε περίπτωση που δεν έχει προβλεφθεί η μη λειτουργία κάποιου χαρακτηριστικού της εφαρμογής σε κάποιον περιηγητή αυτό μπορεί να δημιουργήσει προβλήματα στην εφαρμογή με αποτέλεσμα να μην λειτουργεί σωστά ή να μην λειτουργεί καθόλου. Για το λόγο αυτό ο κατασκευαστής της εφαρμογής με τον πελάτη πρέπει από κοινού να αποφασίζουν ποιος περιηγητής θα είναι ο προτεινόμενος για την εφαρμογή αλλά ταυτόχρονα να προβλέπεται και η περίπτωση χρήσης άλλων περιηγητών. Ένας καλός τρόπος για να ελεγχθεί η συμβατότητα του περιηγητή μας με την HTML5 είναι τα διάφορα διαδικτυακά τεστ που αξιολογούν τις δυνατότητες του.

- Άμεση αναβάθμιση: Ένα ακόμα χαρακτηριστικό παράδειγμα πλεονεκτήματος και μειονεκτήματος ταυτόχρονα αποτελεί και η αναβάθμιση της εφαρμογής. Στην περίπτωση της τοπικής εφαρμογής μια επιχείρηση μπορεί να αναβαθμίσει την εφαρμογή που χρησιμοποιεί όποτε αυτή το κρίνει αναγκαίο κρίνοντας το κόστος αναβάθμισης, την αξιοπιστία της νέας εφαρμογής αλλά και το χρόνο που θα χρειαστούν οι υπάλληλοί της ώστε να προσαρμοστούν στη νέα έκδοση. Αντίθετα στις διαδικτυακές εφαρμογές η αναβάθμιση γίνεται χωρίς πρώτα να ερωτηθούν όλοι οι χρήστες. Για παράδειγμα στην περίπτωση που η επερχόμενη αναβάθμιση μιας τοπικής εφαρμογής έχει σφάλματα τότε μπορεί κάποιος χρήστης (εταιρία) να μην πραγματοποιήσει την αναβάθμιση έως ότου διορθωθούν αυτά. Στην περίπτωση όμως της διαδικτυακής εφαρμογής ο χρήστης (εταιρία) δεν μπορεί να αποτρέψει την αναβάθμιση αυτή.
- Πιθανή μη συμβατότητα κάποιων στοιχείων της εφαρμογής με μια μελλοντική έκδοση του περιηγητή: Τέλος, μειονέκτημα αποτελεί και η πιθανή μη συμβατότητα κάποιων στοιχείων του προγράμματος με μια μελλοντική έκδοση του περιηγητή. Αυτό συμβαίνει διότι συχνά παρατηρείται το φαινόμενο να εγκαταλείπονται κάποια υποστηριζόμενα στοιχεία από τους περιηγητές αν η δημιουργός εταιρία κρίνει ότι αυτά δεν έχουν μέλλον στις εφαρμογές διαδικτύου. Το αποτέλεσμα αυτών των αποφάσεων είναι η δυσλειτουργία κάποιων παλαιότερων εφαρμογών καθιστώντας αναγκαία την αναβάθμιση τους. [2]

# ΚΕΦΑΛΑΙΟ 3

## ΟΡΓΑΝΩΣΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

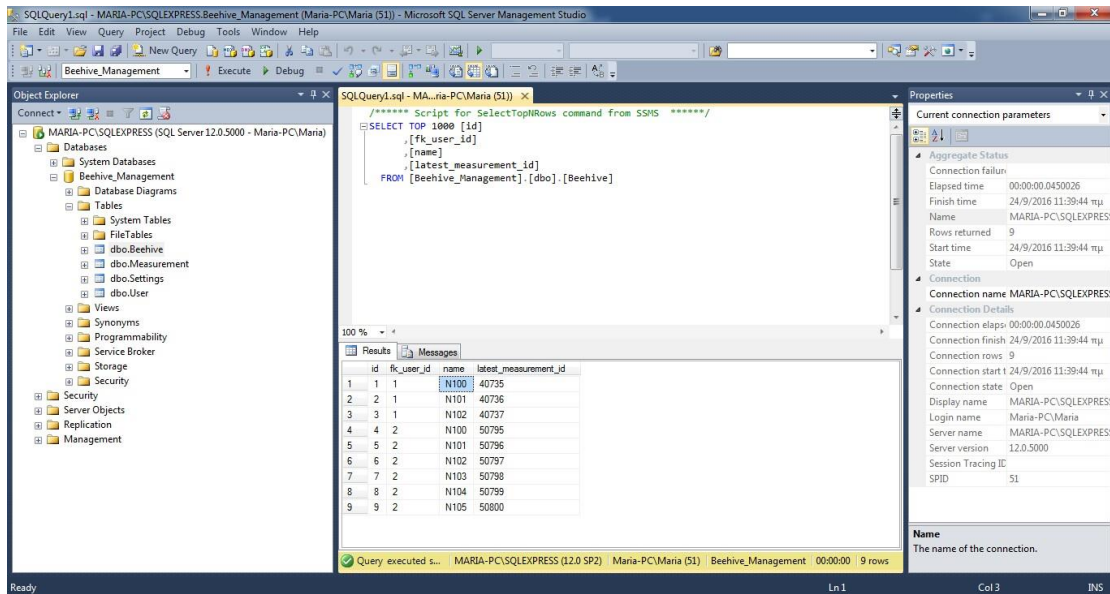
---

Βασικό στοιχείο της εφαρμογής αυτής είναι η δυνατότητα παρουσίασης μετρήσεων στο χρήστη οι οποίες ελήφθησαν από τους αισθητήρες των κυψελών μέσα στο πέρασμα του χρόνου. Απαιτείται λοιπόν η αποθήκευση και διαχείριση του όγκου των δεδομένων με ευκολία και αποτελεσματικότητα. Καθίσταται συνεπώς αναγκαία για την εφαρμογή η χρήση ενός συστήματος διαχείρισης βάσης δεδομένων (Database Management System).

Στο κεφάλαιο αυτό περιγράφεται το σύστημα βάσεων δεδομένων που χρησιμοποιεί η εφαρμογή. Στη συνέχεια περιγράφεται αναλυτικά το σχήμα της βάσης δεδομένων μαζί με τα συστατικά του στοιχεία. Τέλος, παρουσιάζεται το μοντέλο οντοτήτων-συσχετίσεων της εφαρμογής.

### ***3.1 Το σύστημα βάσεων δεδομένων Microsoft SQL Server***

Το σύστημα που χρησιμοποιήθηκε για τη βάση δεδομένων της εφαρμογής είναι ο SQL Server της Microsoft. Πρόκειται για ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS – *Relational Database Management System*) που αναπτύχθηκε από τη Microsoft. Ο SQL Server, ως εξυπηρετητής βάσης δεδομένων, είναι ένα προϊόν λογισμικού με κύρια λειτουργία την αποθήκευση και ανάκτηση των δεδομένων η οποία ζητείται από άλλες εφαρμογές λογισμικού. Οι εφαρμογές αυτές μπορεί να τρέχουν είτε στον ίδιο υπολογιστή, είτε σε έναν άλλον υπολογιστή μέσα στο δίκτυο (συμπεριλαμβανομένου και του διαδικτύου). [3] Η έκδοση που χρησιμοποιήθηκε είναι «SQL Server 2014 Express Edition», η οποία διατίθεται δωρεάν.



Εικόνα 5: SQL Server Management Studio

### 3.2 Περιγραφή των πινάκων της βάσης δεδομένων

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν τέσσερις πίνακες στη βάση δεδομένων: User, Beehive, Measurement και Settings. Πρέπει να επισημάνουμε σε αυτό το σημείο ότι κατά την εγγραφή ενός νέου χρήστη στο σύστημα δημιουργούνται από τον database administrator του συστήματος χειροκίνητα: α. μία εγγραφή στον πίνακα User με τα στοιχεία του χρήστη, β. Χ εγγραφές στον πίνακα Beehive, όπου Χ το πλήθος των κυψελών του χρήστη – μελισσοκόμου και γ. τρεις εγγραφές στον πίνακα Settings, μία για κάθε μέγεθος (θερμοκρασία, υγρασία, βάρος).

Στις υποενότητες που ακολουθούν περιγράφονται αναλυτικά οι πίνακες της βάσης δεδομένων ως προς τα πεδία τους και τη χρησιμότητά τους.



### 3.2.1 User

Στον πίνακα «User» της βάσης δεδομένων συγκεντρώνονται όλα τα προσωπικά στοιχεία που αφορούν τους χρήστες της εφαρμογής, δηλαδή τους μελισσοκόμους. Πιο συγκεκριμένα, οι πληροφορίες που αποθηκεύονται στον πίνακα User για κάθε μελισσοκόμο είναι οι ακόλουθες:

- **id:** Μοναδικό αναγνωριστικό κάθε εγγραφής στον πίνακα User και πρωτεύον κλειδί.
- **username:** Το όνομα χρήστη που χρησιμοποιείται για τη σύνδεση του μελισσοκόμου στη Web εφαρμογή.
- **password:** Ο κωδικός πρόσβασης που χρησιμοποιείται για τη σύνδεση του μελισσοκόμου στη Web εφαρμογή.
- **wsn:** Είναι ένας τετραψήφιος αριθμός που αποτελεί το μοναδικό αναγνωριστικό του Gateway του ασυρμάτου δικτύου αισθητήρων του εν λόγω χρήστη. Με την αποστολή αυτού του αναγνωριστικού από τον Gateway κατά την αποστολή μετρήσεων, πιστοποιείται από το server ποιον χρήστη αφορούν οι μετρήσεις αυτές.
- **name:** Το όνομα του μελισσοκόμου.
- **surname:** Το επώνυμο του μελισσοκόμου.
- **email:** Το προσωπικό email του μελισσοκόμου. Θα χρησιμοποιηθεί για την επικοινωνία μαζί του, εφόσον ο ίδιος το επιθυμεί, σε περιπτώσεις ειδικών συνθηκών στο μελίσσι του. Ο χρήστης μπορεί να αλλάξει αυτήν την τιμή μέσα από τον ιστότοπο της εφαρμογής.
- **useEmail:** Αυτή η λογική τιμή (true/false) δείχνει εάν θα χρησιμοποιείται το email του χρήστη για επικοινωνία μαζί του. Η προκαθορισμένη τιμή της είναι false. Ο χρήστης μπορεί να αλλάξει αυτήν την τιμή μέσα από τον ιστότοπο της εφαρμογής.
- **mobile:** Ο αριθμός κινητού τηλεφώνου του μελισσοκόμου. Θα χρησιμοποιηθεί για να του αποσταλεί SMS, εφόσον ο ίδιος το επιθυμεί, σε περιπτώσεις ειδικών συνθηκών στο μελίσσι του. Ο χρήστης μπορεί να αλλάξει αυτήν την τιμή μέσα από τον ιστότοπο της εφαρμογής.
- **useMobile:** Αυτή η λογική τιμή (true/false) δείχνει εάν θα χρησιμοποιείται το κινητό τηλέφωνο του χρήστη για επικοινωνία μαζί του. Η προκαθορισμένη τιμή της είναι false. Ο χρήστης μπορεί να αλλάξει αυτήν την τιμή μέσα από τον ιστότοπο της εφαρμογής.

- **wsn\_mobile:** Πρόκειται για τον αριθμό τηλεφώνου που αντιστοιχεί στο modem με το οποίο συνδέεται ο Gateway του ασύρματου δικτύου αισθητήρων του μελισσοκόμου. Ο αριθμός αυτός καλείται από τον server ώστε να «αφυπνιστεί» ο Gateway και να του στείλει μετρήσεις, όταν αυτό είναι επιθυμητό.
- **city:** Η πόλη στην οποία βρίσκονται τοποθετημένες οι κυψέλες του μελισσοκόμου, και άρα το ασύρματο δίκτυο αισθητήρων.
- **prefecture:** Ο νομός στον οποίο ανήκει η προαναφερθείσα πόλη (city).
- **schedule\_hours:** Το πεδίο αυτό δείχνει κάθε πόσες ώρες πρέπει να λαμβάνονται μετρήσεις από τις κυψέλες του μελισσοκόμου. Η προκαθορισμένη τιμή είναι κάθε 8 ώρες. Ο χρήστης μπορεί να αλλάξει αυτήν την τιμή μέσα από τον ιστότοπο της εφαρμογής.
- **latest\_measurement\_at:** Το πεδίο αυτό δείχνει την τελευταία χρονική στιγμή την οποία ελήφθησαν μετρήσεις για το WSN. Χρησιμοποιείται σε συνδυασμό με το schedule\_hours πεδίο, ούτως ώστε κάθε μία ώρα που τρέχει ο server το task για τη λήψη μετρήσεων από όλα τα WSN, να μπορεί να εξαχθεί συμπέρασμα εάν πρέπει να ληφθούν μετρήσεις και για το WSN αυτό, ανάλογα εάν έχουν περάσει οι χ schedule\_hours από τη χρονική στιγμή latest\_measurement\_at. Η τιμή του πεδίου αυτού ανανεώνεται αμέσως μετά τη λήψη μετρήσεων, με βάση την τρέχουσα ώρα.
- **active:** Αυτή η λογική τιμή (true/false) υποδηλώνει εάν ο χρήστης είναι ενεργός, με την έννοια ότι εξακολουθεί να χρησιμοποιεί το σύστημα λήψης και αποθήκευσης μετρήσεων στο σύστημα βάσης δεδομένων καθώς και τη web εφαρμογή. Μη ενεργοί χρήστες δε λαμβάνονται υπόψη στη λήψη μετρήσεων και δεν έχουν δυνατότητα σύνδεσης στον ιστότοπο της εφαρμογής.

### 3.2.2 *Beehive*

Ο πίνακας αυτός συγκεντρώνει τα βασικά στοιχεία κάθε κυψέλης για όλα τα μελισσοκομεία. Πιο συγκεκριμένα, οι πληροφορίες που διατηρούνται είναι οι εξής:

- **id:** Μοναδικό αναγνωριστικό κάθε εγγραφής στον πίνακα Beehive και πρωτεύον κλειδί.
- **fk\_user\_id:** Το πεδίο αυτό υποδηλώνει το id του χρήστη στον οποίο ανήκει η συγκεκριμένη κυψέλη. Ειδικότερα, αποτελεί ένα ξένο κλειδί (foreign key) προς τον πίνακα User, στο πεδίο id.

- **name:** Το όνομα της κυψέλης. Οι κυψέλες ονοματίζονται από το ασύρματο δίκτυο αισθητήρων χρησιμοποιώντας το N αρχικά και κατόπιν έναν αύξοντα αριθμό, ξεκινώντας από το 100. Έτσι για παράδειγμα, για έναν μελισσοκόμο που διαθέτει τρεις κυψέλες, αυτές θα έχουν ονόματα N100, N101, N102. Το όνομα αυτό, σε συνδυασμό με το wsn αναγνωριστικό του ασύρματου δικτύου αισθητήρων στο οποίο ανήκει, προσδιορίζουν μοναδικά μία κυψέλη.
- **latest\_measurement\_id:** Προσδιορίζει το id της τελευταίας εγγραφής που εισήχθη στον πίνακα Measurement για τη συγκεκριμένη κυψέλη. Αυτό είναι πάρα πολύ χρήσιμο, διότι μπορεί να εξαχθούν γρήγορα και εύκολα οι τελευταίες μετρήσεις που ελήφθησαν για μία κυψέλη, χωρίς να χρειάζεται εξαντλητική αναζήτηση στον πίνακα των μετρήσεων (Measurement). Αποτελεί ένα ξένο κλειδί (foreign key) προς τον πίνακα Measurement, στο πεδίο id.

### 3.2.3 Measurement

Ο πίνακας αυτός διατηρεί στιγμιότυπα της κατάστασης των κυψελών, δηλαδή αποθηκεύει ποιες ήταν οι τιμές των αισθητήρων θερμοκρασίας, υγρασίας και βάρους μιας κυψέλης σε μια συγκεκριμένη χρονική στιγμή. Τα πεδία αυτού του πίνακα πιο αναλυτικά είναι:

- **id:** Μοναδικό αναγνωριστικό κάθε εγγραφής στον πίνακα Measurement και πρωτεύον κλειδί.
- **fk\_beehive\_id:** Το πεδίο αυτό υποδηλώνει το id της κυψέλης στην οποία ανήκει η συγκεκριμένη μέτρηση. Ειδικότερα, αποτελεί ένα ξένο κλειδί (foreign key) προς τον πίνακα Beehive, στο πεδίο id.
- **temperature:** Η τιμή της θερμοκρασίας σε βαθμούς Κελσίου στο στιγμιότυπο της κατάστασης στην κυψέλη.
- **humidity:** Η τιμή του αισθητήρα υγρασίας σε ποσοστό επί τοις εκατό στο στιγμιότυπο της κατάστασης στην κυψέλη.
- **weight:** Η τιμή του αισθητήρα βάρους σε κιλά στο στιγμιότυπο της κατάστασης στην κυψέλη.

- **date\_of\_measurement:** Η χρονική στιγμή κατά την οποία ελήφθη το στιγμιότυπο της κατάστασης στην κυψέλη. Πρόκειται για τη χρονική στιγμή που έλαβε μετρήσεις ο Gateway από τους αισθητήρες του ασύρματου δικτύου, και όχι για τη χρονική στιγμή που απέστειλε αυτές τις μετρήσεις στο server.

### 3.2.4 Settings

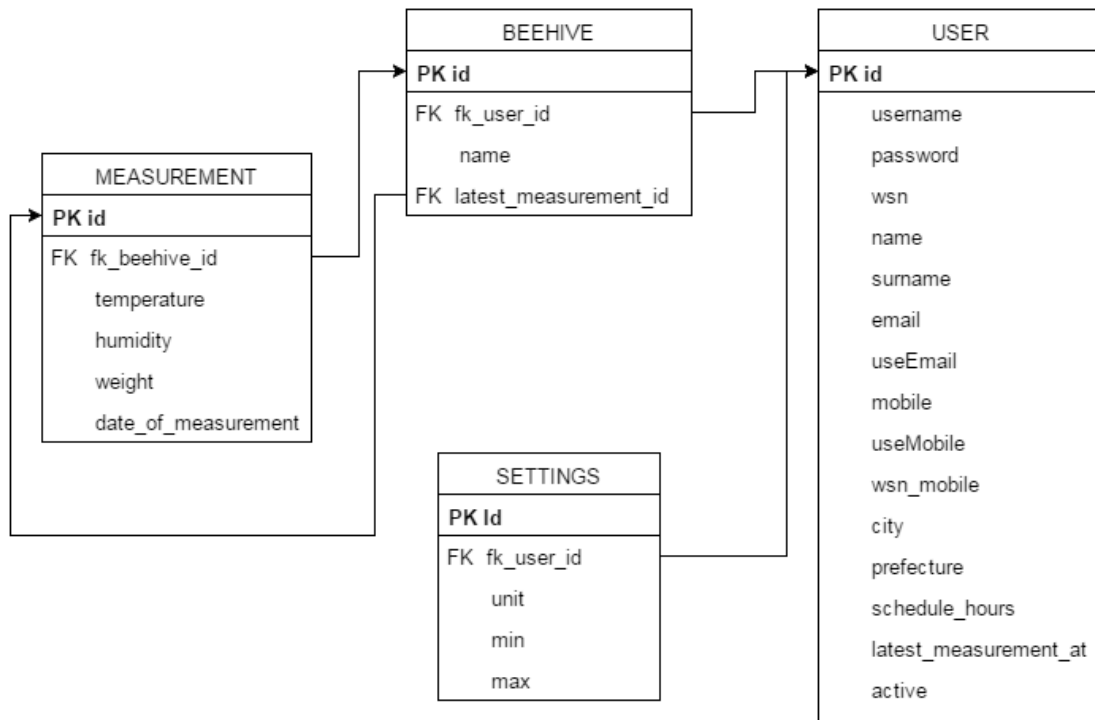
Στον πίνακα αυτό αποθηκεύονται οι κανόνες που έχει ορίσει ο χρήστης για ειδικές – έκτακτες καταστάσεις για τις κυψέλες του. Με βάση αυτούς τους κανόνες ο χρήστης ειδοποιείται στο κινητό του τηλέφωνο ή/και στο email του όταν επικρατούν συγκεκριμένες συνθήκες. Ειδικότερα, για κάθε ένα από τα μετρούμενα μεγέθη (θερμοκρασία, υγρασία, βάρος) υπάρχει ένας κανόνας που ορίζει ποια είναι η τιμή κάτω από την οποία ο χρήστης επιθυμεί να ειδοποιηθεί καθώς και ποια είναι η τιμή που όταν ξεπεραστεί ο χρήστης θα ειδοποιηθεί. Οι τιμές αυτές για κάθε χρήστη αρχικά είναι κενές. Εκείνος, εφόσον το επιθυμεί, θα τις ρυθμίσει μέσα από τον ιστότοπο της εφαρμογής. Με την εγγραφή ενός νέου χρήστη στην εφαρμογή δημιουργούνται τρεις εγγραφές στον πίνακα Settings που έχουν τα ακόλουθα πεδία:

- **id:** Μοναδικό αναγνωριστικό κάθε εγγραφής στον πίνακα Settings και πρωτεύον κλειδί.
- **fk\_user\_id:** Το πεδίο αυτό υποδηλώνει το id του χρήστη στον οποίο ανήκει η συγκεκριμένη ρύθμιση. Ειδικότερα, αποτελεί ένα ξένο κλειδί (foreign key) προς τον πίνακα User, στο πεδίο id.
- **unit:** Το πεδίο αυτό υποδηλώνει ποιο μέγεθος (temperature, humidity, weight) αφορά η συγκεκριμένη ρύθμιση. Για κάθε χρήστη υπάρχει μία εγγραφή στον πίνακα Settings για κάθε ένα από τα τρία μεγέθη.
- **min:** Στο πεδίο αυτό αποθηκεύεται η μικρότερη τιμή του συγκεκριμένου μεγέθους (unit) την οποία θεωρεί ο μελισσοκόμος ως αποδεκτή και φυσιολογική. Σε περίπτωση λοιπόν που η τιμή της μέτρησης του συγκεκριμένου μεγέθους βρεθεί κάτω από αυτήν την τιμή, ο μελισσοκόμος, εφόσον το επιθυμεί, θα ειδοποιηθεί.
- **max:** Στο πεδίο αυτό αποθηκεύεται η μεγαλύτερη τιμή του συγκεκριμένου μεγέθους (unit) την οποία θεωρεί ο μελισσοκόμος ως αποδεκτή και φυσιολογική.

Σε περίπτωση λοιπόν που η τιμή της μέτρησης του συγκεκριμένου μεγέθους ξεπεράσει αυτήν την τιμή, ο μελισσοκόμος, εφόσον το επιθυμεί, θα ειδοποιηθεί.

### 3.3 Σχεσιακό σχήμα της βάσης δεδομένων

Στην Εικόνα 6 παρουσιάζεται το σχεσιακό σχήμα της βάσης δεδομένων της εφαρμογής, όπως παράγεται από τους πίνακες και τις μεταξύ τους σχέσεις.



Εικόνα 6: Σχεσιακό σχήμα της βάσης δεδομένων

### 3.4 Μοντέλο Οντοτήτων – Συσχετίσεων (*Entity-Relationship E-R model*)

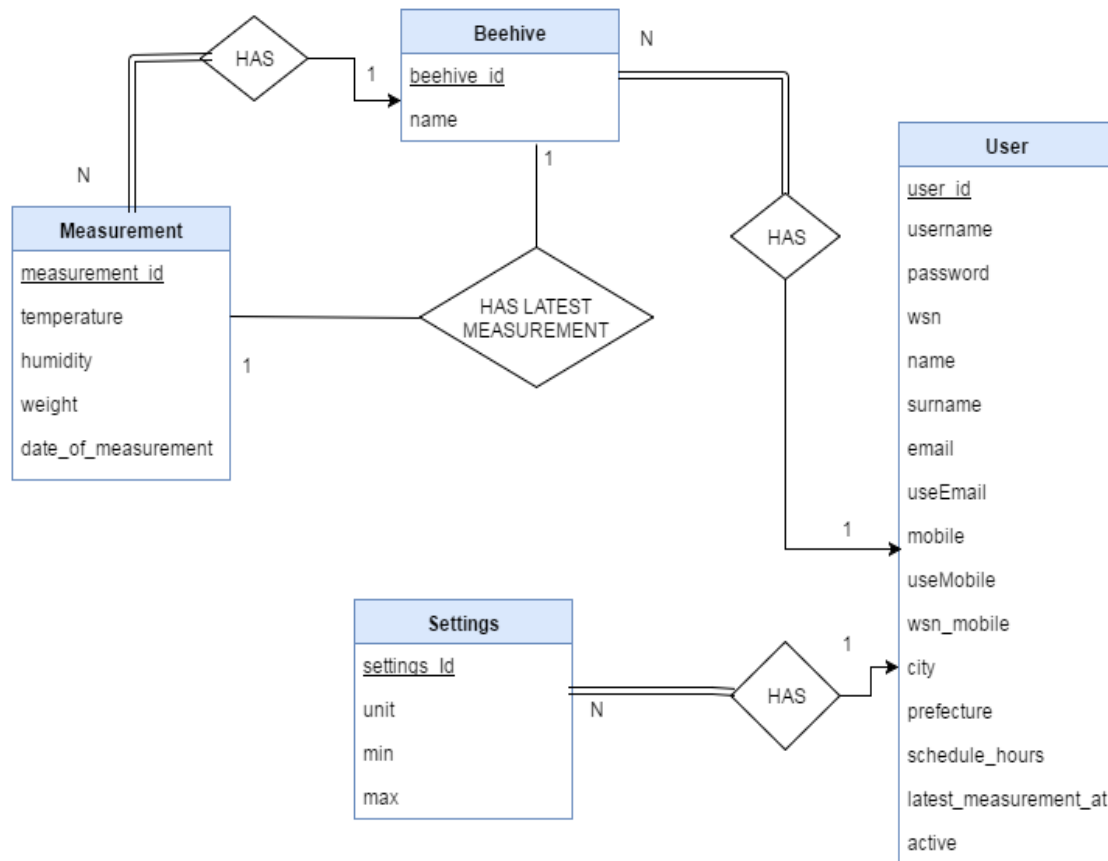
Το μοντέλο δεδομένων σύμφωνα με το οποίο σχεδιάστηκε η βάση δεδομένων της εφαρμογής ακολουθεί το μοντέλο Οντοτήτων-Συσχετίσεων (Entity-Relationship, E-R). Το μοντέλο αυτό χρησιμοποιεί ένα σύνολο από απλά αντικείμενα, που ονομάζονται οντότητες και τις σχέσεις μεταξύ αυτών των αντικειμένων. [4] Τις οντότητες του μοντέλου E-R στην παρούσα εφαρμογή αποτελούν οι πίνακες της βάσης δεδομένων, όπως περιγράφηκαν αναλυτικά στην προηγούμενη ενότητα. Στη συγκεκριμένη εφαρμογή λοιπόν οι οντότητες (entities) είναι οι εξής:

- **User:** Οι χρήστες της εφαρμογής, δηλαδή οι μελισσοκόμοι.
- **Beehive:** Οι κυψέλες των μελισσοκόμων.
- **Measurement:** Τα στιγμιότυπα των μετρήσεων που ελήφθησαν από τις κυψέλες.
- **Settings:** Κανόνες που ορίζουν ποιες συνθήκες στις κυψέλες θεωρούνται ως ειδικές – έκτακτες από το μελισσοκόμο.

Οι σχέσεις (relationships) που συνδέουν τις προαναφερθείσες οντότητες μεταξύ τους είναι οι ακόλουθες:

- Η οντότητα User συνδέεται με την οντότητα Beehive, μέσω της σχέσης Has, με συσχέτιση 1 προς N, αφού ένας χρήστης μπορεί να έχει περισσότερες κυψέλες, ενώ μία κυψέλη ανήκει σε έναν μοναδικό χρήστη. Η σχέση είναι ολική ως προς τις κυψέλες, αφού κάθε κυψέλη πρέπει να ανήκει σε κάποιον χρήστη.
- Η οντότητα User συνδέεται με την οντότητα Settings, μέσω της σχέσης Has, με συσχέτιση 1 προς N, αφού ένας χρήστης έχει περισσότερα settings-κανόνες και πιο συγκεκριμένα, ακριβώς τρία, ένα για κάθε μετρούμενο μέγεθος (θερμοκρασία, υγρασία, βάρος). Αντίστοιχα, μία εγγραφή Settings ανήκει σε έναν μοναδικό χρήστη. Η σχέση είναι ολική ως προς τα Settings αφού κάθε ένα από αυτά πρέπει να ανήκει σε κάποιον χρήστη.
- Η οντότητα Beehive συνδέεται με την οντότητα Measurement μέσω της σχέσης Has, με συσχέτιση 1 προς N, αφού για μία κυψέλη υπάρχουν πολλά στιγμιότυπα μετρήσεων, ενώ ένα στιγμιότυπο μετρήσεων ανήκει σε μία κυψέλη. Είναι ολική ως προς τα στιγμιότυπα των μετρήσεων, αφού κάθε στιγμιότυπο πρέπει να αφορά κάποια κυψέλη.
- Η οντότητα Beehive συνδέεται με την οντότητα Measurement μέσω της σχέσης HAS\_LATEST\_MEASUREMENT, με συσχέτιση 1 προς 1, αφού για μία κυψέλη είναι μοναδικό το τελευταίο στιγμιότυπο μετρήσεων και αντίστοιχα ένα στιγμιότυπο μετρήσεων μπορεί να αποτελεί το πιο πρόσφατο για μία μόνο κυψέλη.

Το διάγραμμα του μοντέλου οντοτήτων-συσχετίσεων της παρούσας εφαρμογής φαίνεται στην Εικόνα 7.



Εικόνα 7: Μοντέλο Οντοτήτων Συσχετίσεων της εφαρμογής





# ΚΕΦΑΛΑΙΟ 4

## ΣΥΛΛΟΓΗ ΜΕΤΡΗΣΕΩΝ ΑΠΟ ΤΟ ΑΣΥΡΜΑΤΟ ΔΙΚΤΥΟ ΑΙΣΘΗΤΗΡΩΝ

---

Στο κεφάλαιο αυτό περιγράφεται η διαδικασία την οποία ακολουθεί ο server προκειμένου να πάρει τις μετρήσεις που συλλέγουν οι κεντρικοί κόμβοι των μελισσοκόμων.

Βασική συνιστώσα της διαδικασίας συλλογής των μετρήσεων αποτελεί ο Gateway του ασυρμάτου δικτύου αισθητήρων. Πρόκειται για τη συσκευή που σχηματίζει το ασύρματο δίκτυο των Endevices. Τα Endevices είναι τα κυκλώματα τα οποία τοποθετούνται στις κυψέλες και έχουν τη δυνατότητα συλλογής τιμών από τους αισθητήρες που είναι συνδεδεμένοι πάνω τους αλλά και τη δυνατότητα επικοινωνίας με τον Gateway, χωρίς να μπορούν όμως να διαδώσουν την πληροφορία αλλού στο δίκτυο. Ο Gateway μπορεί να αποτελέσει γέφυρα διασύνδεσης με τερματικά και άλλα δίκτυα, όπως το δίκτυο GSM. Επίσης, διατηρεί πληροφορίες για το δίκτυο, όπως ποια Endevices είναι κάθε στιγμή συνδεδεμένα στο δίκτυο καθώς και τις τελευταίες τιμές μετρήσεων του δικτύου. [1]

Αξιοποιώντας τη δυνατότητα του Gateway να συνδεθεί με το GSM δίκτυο, καθίσταται εφικτό να αποσταλούν οι μετρήσεις που συλλέγουν τα Endevices εκτός του δικτύου των αισθητήρων, και πιο συγκεκριμένα προς το server μέσω του Διαδικτύου. Όπως προαναφέρθηκε, ο Gateway διατηρεί αποθηκευμένες τις τελευταίες τιμές των μετρήσεων, συνεπώς είναι δυνατόν να αποσταλούν οποτεδήποτε ζητηθούν. Ο server αναλαμβάνει την «αφύπνιση» του Gateway ώστε να λαμβάνει μετρήσεις μόνο όταν αυτό είναι επιθυμητό. Ακολουθεί αποθήκευση των μετρήσεων σε βάση δεδομένων και επεξεργασία τους για την άμεση ενημέρωση του μελισσοκόμου στο προσωπικό του τηλέφωνο ή/και email εφόσον το επιθυμεί.

## **4.1 Σχεδιαστικές και προγραμματιστικές επιλογές**

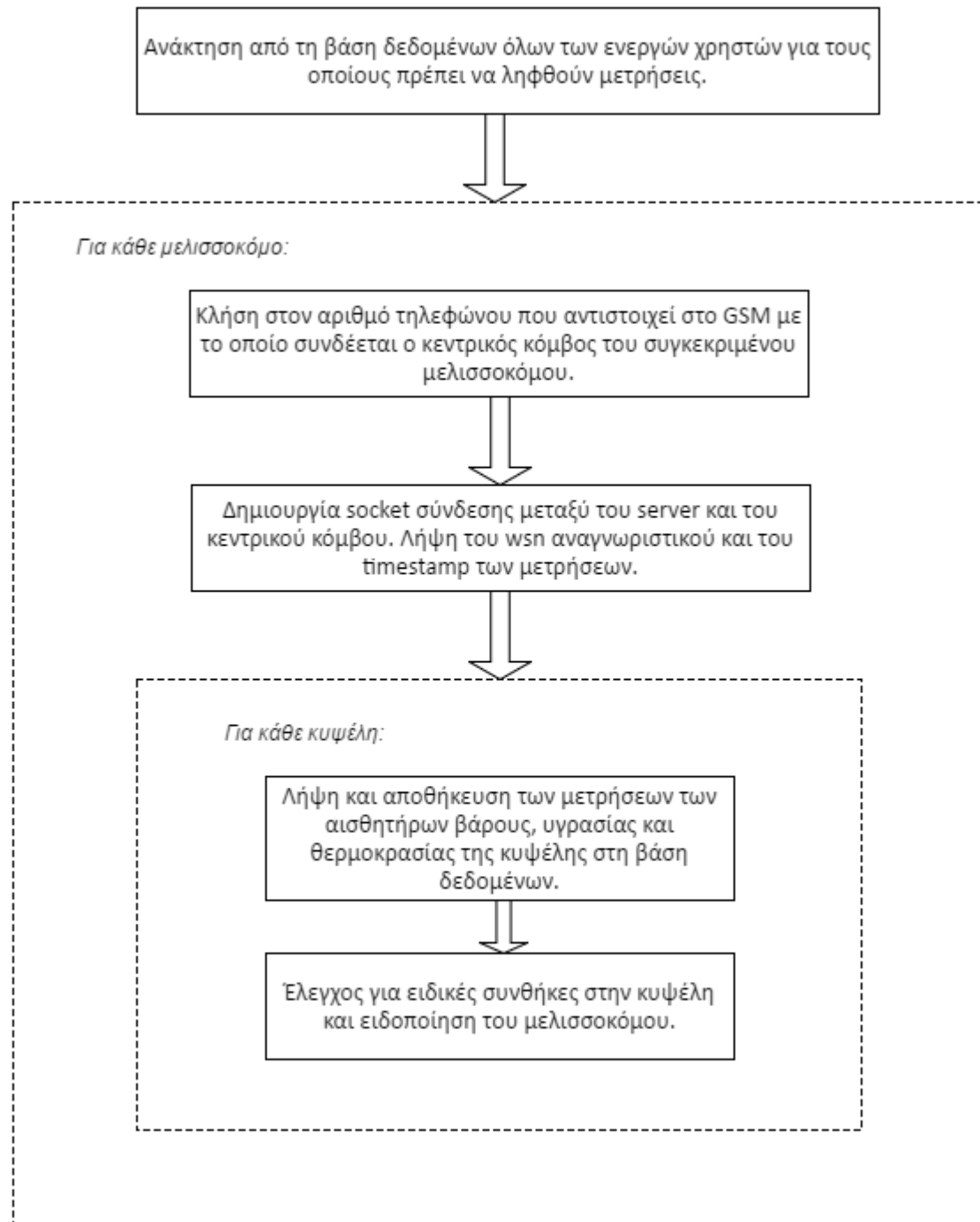
Η διαδικασία λήψης των μετρήσεων πραγματοποιείται μέσα από ένα Console Application. Επιλέχθηκε η δημιουργία exe αρχείου έτσι ώστε να είναι δυνατό να εκτελεστεί αυτόματα μέσα από τον Task Scheduler των Windows προγραμματισμένες ημέρες και ώρες. Για κάθε χρήστη-μελισσοκόμο επιλέξαμε να λαμβάνονται μετρήσεις κάθε οκτώ ώρες, όμως ο ίδιος μπορεί να αλλάξει την επιλογή αυτή μέσα από το λογαριασμό του στη web εφαρμογή, ορίζοντας ακέραιο θετικό αριθμό ωρών. Η διαδικασία λήψης των μετρήσεων εκτελείται καθημερινά κάθε μία ώρα μέσα από τον Task Scheduler και λαμβάνονται μετρήσεις για όσους χρήστες χρειάζεται κάθε φορά.

Όπως αναφέρθηκε παραπάνω, ο server είναι εκείνος που αναλαμβάνει την αφύπνιση του Gateway προκειμένου να στείλει μετρήσεις. Αυτό επιλέξαμε να γίνεται με αναπάντητη τηλεφωνική κλήση από το server προς το GSM modem με το οποίο συνδέεται ο Gateway, το οποίο αναλαμβάνει να αφυπνίσει τον Gateway ώστε να στείλει τις μετρήσεις. Η τηλεφωνική κλήση από τον server επίσης πραγματοποιείται χρησιμοποιώντας GSM modem. Η εναλλακτική λύση θα ήταν ο Gateway να έχει διαρκώς ανοιχτή TCP σύνδεση και να αναμένει τη λήψη κάποιου συγκεκριμένου χαρακτήρα από το server ώστε να ξεκινήσει την αποστολή των μετρήσεων. Κάτι τέτοιο όμως θα είχε ως αποτέλεσμα να καταναλώνονται τα δεδομένα του Internet με επιπλέον κόστος για το χρήστη. Το ίδιο ισχύει και για την κατανάλωση ενέργειας απ' την πλευρά του Gateway. Για λόγους οικονομίας λοιπόν τόσο δικτυακών δεδομένων (ογκοχρέωση) όσο και ενεργειακών δεδομένων (μπαταρία Gateway) δεν επιλέχθηκε αυτή η λύση.

Η επικοινωνία μεταξύ Gateway – server για την αποστολή των μετρήσεων γίνεται με sockets. Ο server λειτουργεί ως TCP listener ο οποίος αποδέχεται συνδέσεις από τους Gateways. Ο Gateway χρησιμοποιώντας το Dynamic DNS όνομα του server και μία πόρτα μπορεί να συνδεθεί μαζί του και κατόπιν να στείλει δεδομένα. Εναλλακτικά, θα μπορούσε να πραγματοποιείται αποστολή SMS από τον Gateway προς το server μέσα στο οποίο θα περιέχονταν οι μετρήσεις. Σε αυτήν την περίπτωση όμως θα είχαμε περιορισμό ως προς το πλήθος των χαρακτήρων που θα εμπεριείχε το SMS καθώς και μεγαλύτερο κόστος σε σχέση με τα δεδομένα σύνδεσης στο Διαδίκτυο.

## 4.2 Αναλυτική περιγραφή της διαδικασίας συλλογής μετρήσεων

Στην Εικόνα 8 παρουσιάζεται ένα διάγραμμα το οποίο δείχνει επιγραμματικά τα βήματα της διαδικασίας που ακολουθείται για τη συλλογή μετρήσεων από τους Gateways των μελισσοκόμων.



Εικόνα 8: Συνοπτική περιγραφή της διαδικασίας συλλογής μετρήσεων

#### **4.2.1 Ανάκτηση από τη βάση δεδομένων όλων των ενεργών χρηστών για τους οποίους πρέπει να ληφθούν μετρήσεις**

Η εφαρμογή χρησιμοποιεί βάση δεδομένων για την αποθήκευση όλων των δεδομένων της εφαρμογής. Στον πίνακα με τα στοιχεία των χρηστών (User) είναι αποθηκευμένη η πληροφορία της χρονικής στιγμής κατά την οποία έλαβε τελευταία φορά μετρήσεις ο server για τις κυψέλες του μελισσοκόμου αυτού (latest\_measurement\_at), καθώς και κάθε πόσες ώρες θα πρέπει να λαμβάνει μετρήσεις για το συγκεκριμένο χρήστη (schedule\_hours). Με βάση τις δύο αυτές πληροφορίες, εάν έχουν περάσει οι προγραμματισμένες ώρες από την τελευταία λήψη μετρήσεων, τότε θα πρέπει να ληφθούν ξανά μετρήσεις για το χρήστη αυτό. Με αυτήν τη μέθοδο δημιουργείται μία λίστα με τους μελισσοκόμους για τους οποίους πρέπει να ληφθούν μετρήσεις στη συγκεκριμένη εκτέλεση του task.

#### **4.2.2 Λήψη μετρήσεων για κάθε μελισσοκόμο – χρήστη**

Για κάθε έναν από τους μελισσοκόμους της λίστας που κατασκευάστηκε στο προηγούμενο βήμα λαμβάνονται μετρήσεις με την εξής διαδικασία:

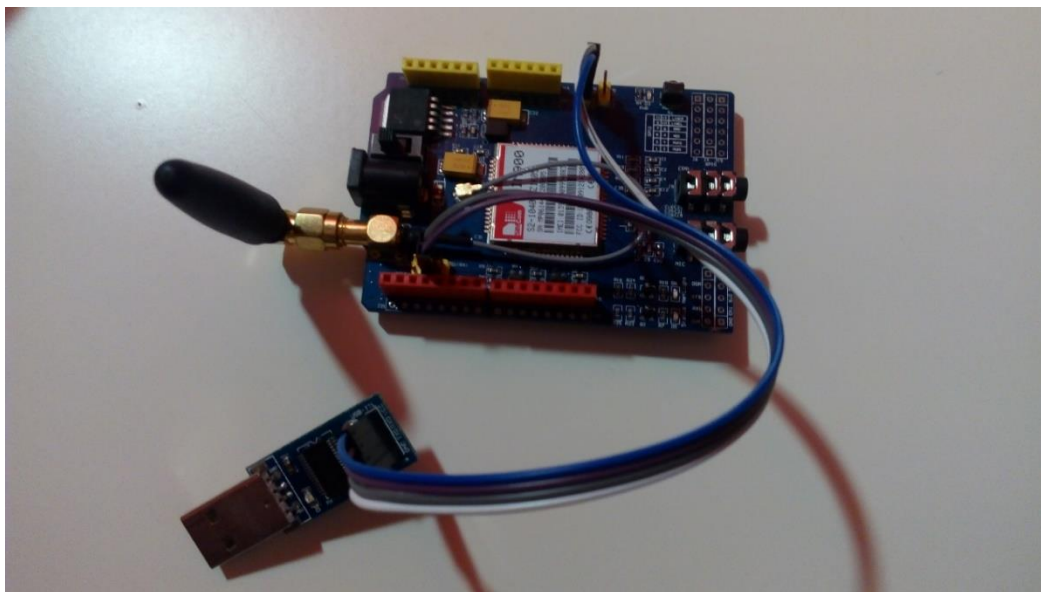
1. *Κλήση στον αριθμό τηλεφώνου που αντιστοιχεί στο GSM με το οποίο συνδέεται ο κεντρικός κόμβος του μελισσοκόμου.*
2. *Δημιουργία socket σύνδεσης μεταξύ του server και του κεντρικού κόμβου του ασύρματου δικτύου αισθητήρων.*
  - 2.1. *Επεξεργασία των μετρήσεων των αισθητήρων των κυψελών.*
  - 2.2. *Έλεγχος για ειδικές συνθήκες και ειδοποίηση του μελισσοκόμου.*

Τα παραπάνω βήματα αναλύονται ακολούθως.

1. *Κλήση στον αριθμό τηλεφώνου που αντιστοιχεί στο GSM με το οποίο συνδέεται ο κεντρικός κόμβος του μελισσοκόμου.*

Ο Gateway κάθε μελισσοσμήνης έχει προγραμματιστεί κατάλληλα έτσι ώστε όταν κληθεί ο αριθμός τηλεφώνου του GSM modem με το οποίο συνδέεται, τότε να τρέχει ένα πρόγραμμα το οποίο να αποστέλλει μετρήσεις προς το server. Επομένως, ο server είναι εκείνος που θα πρέπει να καλέσει τον αριθμό αυτό με αυτοματοποιημένο τρόπο.

Για το σκοπό αυτό έχουμε συνδέσει ένα GSM modem πάνω στο server μέσω σειριακής θύρας (USB). Πρόκειται για το SIM 900 (Εικόνα 9). Το πρόγραμμα που τρέχει λοιπόν στο server, ανακτά τον αριθμό τηλεφώνου του GSM modem που αντιστοιχεί στο WSN του συγκεκριμένου μελισσοκόμου από τη βάση δεδομένων (wsn\_mobile πεδίο στον πίνακα User) και στέλνει τις κατάλληλες εντολές προς το modem, μέσω της σειριακής θύρας στην οποία είναι συνδεδεμένο, οι οποίες θα προκαλέσουν την κλήση.



**Εικόνα 9: GSM πομποδέκτης (SIM 900)**

2. *Δημιουργία socket σύνδεσης μεταξύ του server και του κεντρικού κόμβου του ασύρματου δικτύου αισθητήρων.*

Αμέσως μετά την ολοκλήρωση της αναπάντητης κλήσης προς την απομακρυσμένη συσκευή, ξεκινά να τρέχει ένα πρόγραμμα το οποίο θα στείλει τις τελευταίες μετρήσεις στο server. Η επικοινωνία μεταξύ των δύο επιτυγχάνεται μέσω socket. Ο server λειτουργεί ως TCP Listener και αποδέχεται σύνδεση από οποιαδήποτε IP στην πόρτα 5000. Με το Dynamic DNS, η απομακρυσμένη συσκευή επιχειρεί να συνδεθεί με το server χρησιμοποιώντας το DDNS όνομά του στην πόρτα 5000. Προκειμένου η σύνδεση αυτή να είναι επιτυχής, πρέπει να προστεθεί Inbound Rule στο firewall του server, το οποίο θα επιτρέπει την εισερχόμενη TCP σύνδεση στην πόρτα 5000. Επιπλέον, πρέπει να γίνουν οι κατάλληλες ρυθμίσεις στο router που θα δρομολογήσουν την προσπάθεια σύνδεσης από την απομακρυσμένη συσκευή προς το server.

Πρώτον, χρειάζεται να δοθεί στο server στατική IP και στη συνέχεια να προστεθεί κανόνας στο router που θα δρομολογεί την εισερχόμενη TCP κίνηση στην πόρτα 5000 προς τη στατική IP που έχει ανατεθεί στο server στην πόρτα 5000. Μόλις επιτευχθεί η μεταξύ τους σύνδεση, η απομακρυσμένη συσκευή στέλνει ένα πακέτο που έχει την παρακάτω μορφή:

```
7001
timestamp 16/08/16,21:28:23
N100 H=60% T=28C W=12,7KG
N101 H=35% T=28C W=30,0KG
N102 H=42% T=30C W=22,0KG
N103 H=62% T=28C W=12,7KG
N104 H=35% T=29C W=30,0KG
N105 H=46% T=29C W=22,1KG
```

Στην 1<sup>η</sup> γραμμή δίνεται το WSN αναγνωριστικό του ασύρματου δικτύου αισθητήρων, στη 2<sup>η</sup> γραμμή το timestamp των μετρήσεων και στις επόμενες γραμμές οι μετρήσεις που έχουν ληφθεί για κάθε κυψέλη ξεχωριστά. Το pattern που ακολουθείται είναι:

Όνομα κυψέλης – Υγρασία (ποσοστό επί τοις εκατό, χωρίς δεκαδικά ψηφία) – Θερμοκρασία (σε βαθμούς Κελσίου, χωρίς δεκαδικά ψηφία) – Βάρος (σε κιλά, με ακρίβεια ενός δεκαδικού ψηφίου).

Ο server διαβάζει και επεξεργάζεται κάθε μία γραμμή του μηνύματος ξεχωριστά. Για τις γραμμές που αφορούν τις μετρήσεις των αισθητήρων ακολουθεί κάποια βήματα επεξεργασίας των πληροφοριών.

### *2.1. Επεξεργασία των μετρήσεων των αισθητήρων των κυψελών.*

Οι μετρήσεις των αισθητήρων για κάθε κυψέλη της μελισσοσμήνης αποστέλλονται σε μία τυποποιημένη αλλά και κατανοητή μορφή. Με αυτόν τον τρόπο καθίσταται δυνατό στον αναγνώστη του μηνύματος να κατανοήσει τις μετρήσεις, αλλά και στο πρόγραμμα του server να τις επεξεργαστεί.

Ο καλύτερος δυνατός τρόπος εξαγωγής πληροφοριών από το string που περιγράφει τις μετρήσεις μίας κυψέλης είναι οι κανονικές εκφράσεις, τα «Regular Expressions». Μία κανονική έκφραση είναι μία ακολουθία χαρακτήρων που ορίζουν ένα μοτίβο αναζήτησης. [5] Με τη χρήση των κανονικών εκφράσεων μπορούμε να ανακτήσουμε εύκολα λοιπόν τις τιμές των μεγεθών της υγρασίας, της θερμοκρασίας και του βάρους.

Προκειμένου να είναι εύκολη και γρήγορη η ανάκτηση των πιο πρόσφατων μετρήσεων των αισθητήρων μιας κυψέλης από τη βάση δεδομένων, διατηρείται στις πληροφορίες της κάθε κυψέλης το Id της τελευταίας μέτρησης που ελήφθη για αυτήν. Όταν ληφθούν λοιπόν νέες μετρήσεις από τους αισθητήρες, ελέγχεται εάν είναι πιο πρόσφατες από τις τελευταίες που ελήφθησαν ώστε να αποφευχθεί η δημιουργία διπλοεγγραφών στον πίνακα των μετρήσεων Measurement. Αυτός ο έλεγχος γίνεται με βάση το timestamp που αποστέλλεται από τον Gateway. Άρα, αν το νέο timestamp είναι μικρότερο ή ίσο από εκείνο των τελευταίων μετρήσεων, η διαδικασία επεξεργασίας των μετρήσεων της συγκεκριμένης κυψέλης ολοκληρώνεται εδώ, αφού οι ληφθείσες μετρήσεις είναι παρωχημένες.

Στην αντίθετη περίπτωση, οι μετρήσεις είναι έγκυρες οπότε πρέπει να γίνουν κάποιες ενέργειες. Αρχικά αποθηκεύεται μία νέα εγγραφή στον πίνακα των μετρήσεων Measurement. Στη συνέχεια, η εγγραφή της κυψέλης στον πίνακα Beehive ενημερώνεται ώστε η τελευταία της μέτρηση (latest\_measurement\_id) να δείχνει στο Id της καινούριας μέτρησης στον πίνακα Measurement.

## *2.2. Έλεγχος για ειδικές συνθήκες και ειδοποίηση του μελισσοκόμου.*

Για να ολοκληρωθεί η διαδικασία επεξεργασίας των μετρήσεων των αισθητήρων, θα πρέπει να ελεγχθούν οι ρυθμίσεις ειδοποίησης που έχει καθορίσει ο χρήστης για ειδικές ή επείγουσες καταστάσεις στις κυψέλες. Ανακτώνται λοιπόν από τον πίνακα Settings της βάσης δεδομένων οι ρυθμίσεις του συγκεκριμένου User. Αν η μέτρηση ενός ή παραπάνω μεγεθών εκ των τριών (υγρασία, θερμοκρασία, βάρος) έχει βγει εκτός των ορίων που έχει θέσει ο χρήστης, τότε θα ειδοποιηθεί με SMS ή/και email, ανάλογα εάν έχει δηλώσει στις προτιμήσεις του ότι επιθυμεί να ενημερώνεται μέσω SMS ή/και email από την εφαρμογή.

Οι πληροφορίες αυτές βρίσκονται αποθηκευμένες στην εγγραφή του χρήστη στον πίνακα User της βάσης δεδομένων (πεδία useMobile και useEmail αντίστοιχα).

Η αποστολή του SMS πραγματοποιείται μέσω του GSM modem SIM 900, το οποίο είναι συνδεδεμένο πάνω στο server μέσω σειριακής θύρας. Μέσα από τον κώδικα αποστέλλονται οι κατάλληλες εντολές στη σειριακή θύρα που θα προκαλέσουν την αποστολή του SMS στον παραλήπτη.

Όσον αφορά την αποστολή email, κατασκευάστηκε ένας λογαριασμός στο Yahoo Mail ο οποίος λειτουργεί ως αποστολέας του email. Το email αποστέλλεται χρησιμοποιώντας τον SMTP server της Yahoo στην πόρτα 587, με Secure Sockets Layer για κρυπτογράφηση της σύνδεσης και με τα credentials του λογαριασμού που φτιάχτηκε στο Yahoo Mail. Συντάσσεται κατάλληλο μήνυμα που ενημερώνει το μελισσοκόμο ποιο μέγεθος, με ποια τιμή και σε ποια κυψέλη βγήκε εκτός των ορίων που έχουν τεθεί από τον ίδιο. Αυτό είναι το περιεχόμενο του SMS αλλά και του body του email που θα σταλεί:

*«WEIGHT/TEMPERATURE/HUMIDITY IN BEEHIVE NXXX HAS REACHED A VALUE OF YY KG/C/%»*



# ΚΕΦΑΛΑΙΟ 5

## Η ΔΙΑΔΙΚΤΥΑΚΗ ΕΦΑΡΜΟΓΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΤΩΝ ΚΥΨΕΛΩΝ

---

Στο κεφάλαιο αυτό περιγράφεται αναλυτικά ο ιστότοπος της εφαρμογής που κατασκευάστηκε, μέσα από την οποία οι μελισσοκόμοι – χρήστες έχουν τη δυνατότητα να ελέγξουν την τρέχουσα κατάσταση των κυψελών τους, αλλά και να ανατρέξουν στο ιστορικό παλαιότερων μετρήσεων. Αρχικά αναλύονται οι λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος. Κατόπιν, περιγράφονται αναλυτικά όλες οι υπηρεσίες που παρέχονται μέσα από τις σελίδες της διαδικτυακής εφαρμογής και παρατίθενται ενδεικτικά στιγμιότυπα.

### ***5.1 Απαιτήσεις του συστήματος***

#### ***5.1.1 Λειτουργικές απαιτήσεις***

Οι λειτουργικές απαιτήσεις (functional requirements) ενός συστήματος περιγράφουν τι πρέπει να κάνει ένα σύστημα. [6] Στη συγκεκριμένη εφαρμογή οι λειτουργικές απαιτήσεις είναι:

1. Σύνδεση ενός εγγεγραμμένου χρήστη στην εφαρμογή.
2. Προβολή των τελευταίων μετρήσεων που ελήφθησαν για τις κυψέλες του μελισσοκόμου.
3. Δυνατότητα του χρήστη να ζητήσει τις τρέχουσες μετρήσεις των κυψελών.
4. Παρουσίαση γραφημάτων στα οποία ο χρήστης μπορεί να μελετήσει την εξέλιξη ενός εκ των τριών μεγεθών (βάρος, θερμοκρασία, υγρασία) για μία ή και περισσότερες κυψέλες για χρονικό διάστημα που θα επιλέξει ο ίδιος.
5. Δυνατότητα προβολής και επεξεργασίας των μεθόδων και στοιχείων επικοινωνίας του χρήστη για περιπτώσεις έκτακτης ανάγκης (email, mobile).

6. Δυνατότητα προβολής και επεξεργασίας των επιλογών του χρήστη για τις τιμές που θεωρούνται οριακές για τα τρία μεγέθη.
7. Δυνατότητα προβολής και επεξεργασίας της τιμής των προγραμματισμένων ωρών που μεσολαβούν ανάμεσα σε δύο λήψεις μετρήσεων.
8. Αποσύνδεση συνδεδεμένου χρήστη από την εφαρμογή.

### **5.1.2 Μη λειτουργικές Απαιτήσεις**

Οι μη λειτουργικές απαιτήσεις περιγράφουν ιδιότητες του συστήματος που συνήθως εκφράζονται βάσει χαρακτηριστικών της μορφής: απόδοση, χρηστικότητα, ασφάλεια, νομιμότητα και ιδιωτικότητα. Με άλλα λόγια, περιγράφουν το πώς (ή πόσο καλά) θα υποστηρίξει το σύστημα τις λειτουργικές απαιτήσεις. [6] Στη συγκεκριμένη εφαρμογή οι μη λειτουργικές απαιτήσεις είναι:

1. Προσβασιμότητα της εφαρμογής μέσω του Διαδικτύου.
2. Ευκολία στην εκμάθηση και τη χρήση της εφαρμογής.
3. Ανταποκρισιμότητα (responsiveness) της ιστοσελίδας ώστε να είναι εύχρηστη από κάθε είδους συσκευή (desktop, laptop, tablet, mobile).
4. Ικανοποιητικός χρόνος απόκρισης της εφαρμογής σε βασικές λειτουργίες.
5. Χρήση βιβλιοθήκης ανοιχτού κώδικα για τη δημιουργία γραφημάτων.

## **5.2 Πρόσβαση στην εφαρμογή**

Όπως αναφέρθηκε στην προηγούμενη ενότητα, σημαντική απαίτηση της εφαρμογής αποτελεί η προσβασιμότητά της μέσω του Διαδικτύου. Οι χρήστες της θα πρέπει να μπορούν να έχουν πρόσβαση σε αυτήν οπουδήποτε βρίσκονται από το κινητό τους τηλέφωνο ή τον υπολογιστή ώστε να ενημερωθούν για την κατάσταση στις κυψέλες τους. Συνεπώς, θα πρέπει να αντιστοιχίζεται στη διαδικτυακή εφαρμογή μοναδική διεύθυνση την οποία θα γράφουν οι χρήστες στον browser.

Μερικές από τις διαθέσιμες επιλογές για την επίτευξη αυτού του στόχου ήταν:

- Static IP ή DNS name. Η αγορά μίας στατικής IP ή ενός dns ονόματος αποτελεί μία εύκολη λύση. Έχει χρέωση μηνιαία ή ετήσια και ποικίλλει ανάλογα με την εταιρία που πουλά.
- Φιλοξένηση της εφαρμογής σε μια cloud πλατφόρμα επί πληρωμή.

- Χρήση του cloud service GRNET okeanos. Η ακαδημαϊκή κοινότητα παρέχει στους φοιτητές sources για τη δημιουργία Virtual Machine στον okeano, αλλά και μία μοναδική IPv4 διεύθυνση την οποία μπορούν να αναθέσουν σε ένα μηχάνημά τους. Έτσι, αν η εφαρμογή που εξετάζεται ήταν hosted σε ένα VM στον okeano με μοναδική IP, τότε θα ήταν προσβάσιμη μέσω του Διαδικτύου ανά πάσα στιγμή.
- Χρήση Dynamic DNS. Πρόκειται για μία μέθοδο αυτόματης ενημέρωσης της IP του ονόματος ενός server στο Domain Name System (DNS). Η υπηρεσία αυτή παρέχεται από διάφορους φορείς είτε δωρεάν είτε επί πληρωμή.

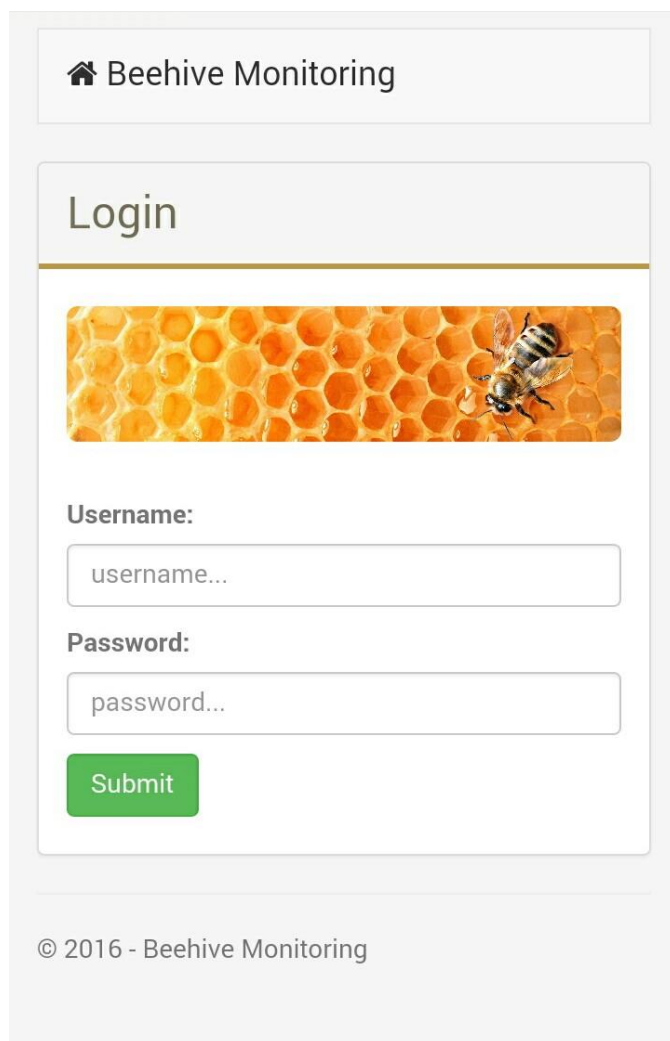
Στην παρούσα εφαρμογή επιλέχθηκε η χρήση Dynamic DNS. Ο πάροχος που χρησιμοποιείται είναι το «no-ip». Ο χρήστης μπορεί να εγγραφεί στο no-ip.com και να δημιουργήσει το πολύ τρία δωρεάν Dynamic DNS ονόματα. Με τη χρήση μιας απλής console εφαρμογής (DUC), το no-ip αναγνωρίζει την τρέχουσα public ip του router και την αντιστοιχίζει στο/στα Dynamic DNS ονόματα του χρήστη.

Η web εφαρμογή «Beehive Monitoring» είναι προσβάσιμη μέσω του Διαδικτύου στο URL: <http://beehivemonitoring.ddns.net:8080>.

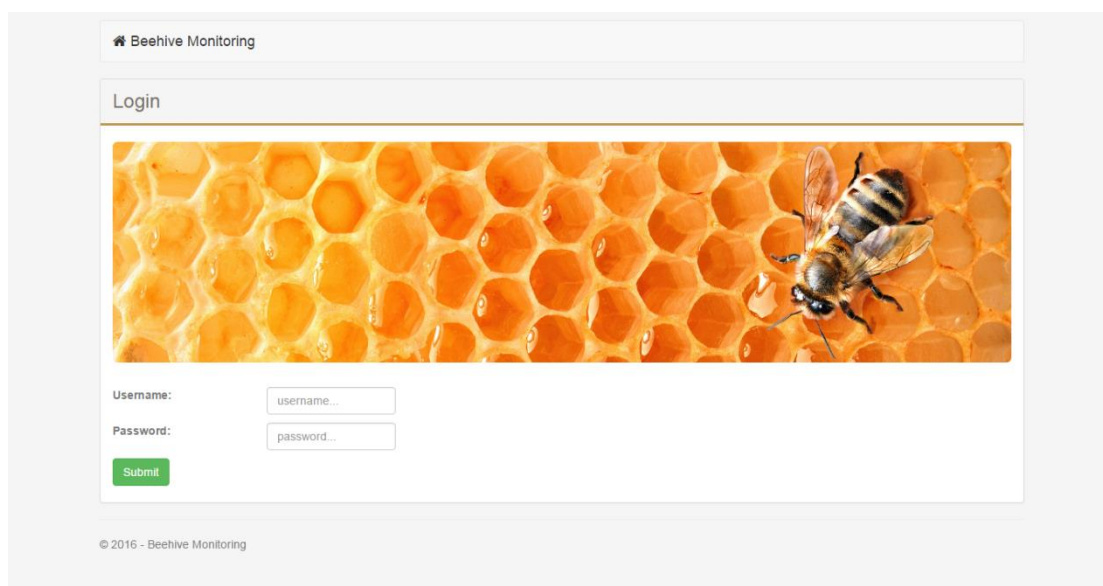
### **5.3 Login στην εφαρμογή**

Με τη χρήση του url που αναφέρθηκε στην προηγούμενη ενότητα, ο χρήστης ανακατευθύνεται στη Login σελίδα της εφαρμογής. Σε αυτήν τη σελίδα εμφανίζεται μία φόρμα στην οποία ο χρήστης πρέπει να συμπληρώσει τα στοιχεία του προκειμένου να συνδεθεί. Πιο συγκεκριμένα, πρέπει να δοθούν το username και το password του χρήστη. Τα στοιχεία αυτά είναι υποχρεωτικά για τη σύνδεση στην εφαρμογή και η ορθότητά τους ελέγχεται από το server με βάση τα στοιχεία των εγγεγραμμένων χρηστών στη βάση δεδομένων. Σε περίπτωση εισαγωγής λανθασμένων στοιχείων εμφανίζεται κατάλληλο μήνυμα στο χρήστη ώστε να προσπαθήσει ξανά. Επιπλέον, μη εγγεγραμμένοι ή μη ενεργοί χρήστες δεν μπορούν να συνδεθούν στην εφαρμογή. Εάν τα στοιχεία του χρήστη είναι ορθά, τότε ανακατευθύνεται στην κεντρική σελίδα της εφαρμογής «Home Page».

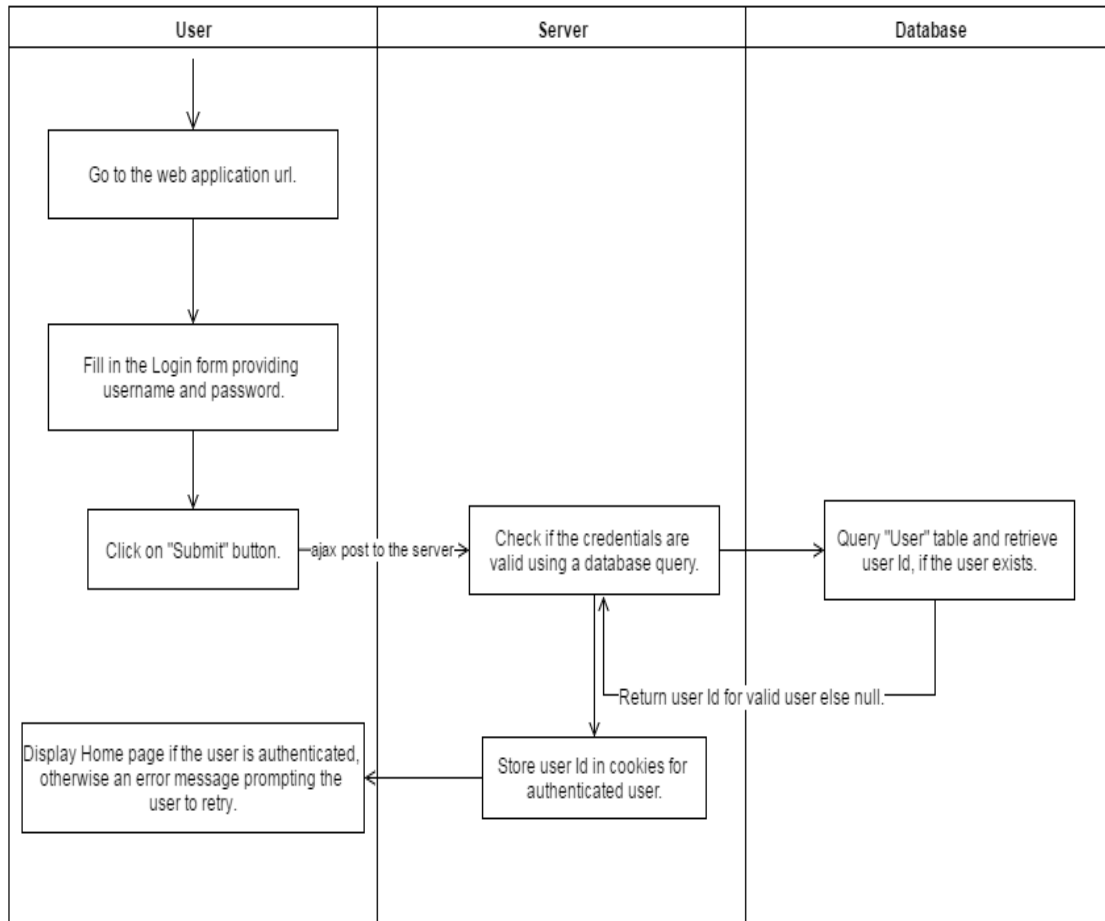
Παρακάτω φαίνεται η οθόνη της Login σελίδας σε κινητό τηλέφωνο (Εικόνα 10) και υπολογιστή (Εικόνα 11) και κατόπιν ένα διάγραμμα μεταξύ των εμπλεκόμενων οντοτήτων για τη login διαδικασία (Εικόνα 12).



Εικόνα 10: Οθόνη «Login» της εφαρμογής σε κινητό τηλέφωνο



Εικόνα 11: Οθόνη «Login» της εφαρμογής σε υπολογιστή



**Εικόνα 12: Η διαδικασία πιστοποίησης ενός χρήστη στην εφαρμογή**

Αξίζει να σημειωθεί ότι η μετάδοση των ευαίσθητων πληροφοριών του χρήστη (username, password) από το User Interface προς το server πραγματοποιείται με Ajax Post κλήση για λόγους προστασίας και ασφάλειας δεδομένων.

Η πιστοποίηση (authentication) του χρήστη γίνεται χρησιμοποιώντας τη «Forms Authentication» μέθοδο που υποστηρίζει το .NET περιβάλλον. Με αυτήν τη μέθοδο ο χρήστης πιστοποιείται από ένα forms authentication ticket για οποιαδήποτε ενέργεια πραγματοποιεί εντός του website. Αυτό το ticket θα αποστέλλεται στο server σε οποιοδήποτε request του client. Αυτό πραγματοποιείται τις περισσότερες φορές με την τοποθέτηση του ticket στα cookies του χρήστη. Τα cookies είναι μικρά αρχεία κειμένου που βρίσκονται στον υπολογιστή του χρήστη και μεταδίδονται στις κεφαλίδες HTTP για κάθε αίτηση στην ιστοσελίδα που δημιούργησε το cookie. Ως εκ τούτου, εφόσον έχει δημιουργηθεί μία φορά το ticket που αποθηκεύεται στα cookies του browser, κάθε επόμενη επίσκεψη σε αυτήν την τοποθεσία στέλνει το ticket, μαζί με το request, προσδιορίζοντας έτσι το χρήστη. [7]

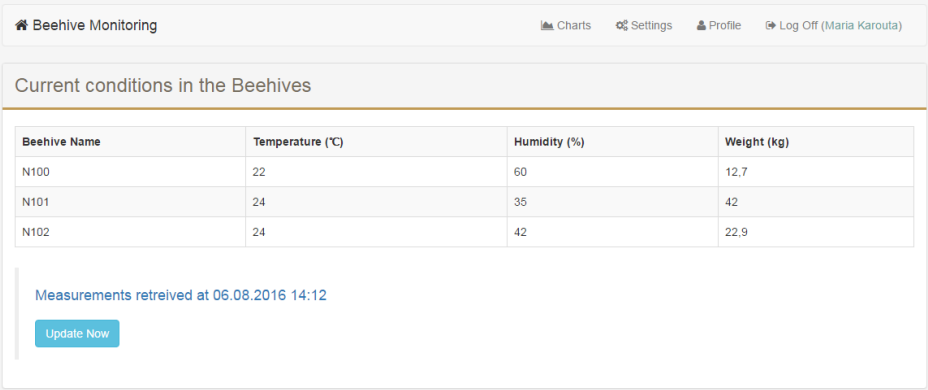
Στην παρούσα εφαρμογή χρησιμοποιήθηκε το cookie ως ticket την πιστοποίηση του χρήστη. Επιπλέον, στα cookies της εφαρμογής αποθηκεύονται μεταβλητές ώστε να χρησιμοποιηθούν σε μετέπειτα κλήσεις προς το server. Η αποθήκευση του user id στα cookies διευκολύνει την υλοποίηση της πλοήγησης του χρήστη στις σελίδες της εφαρμογής, αφού είναι ανά πάσα στιγμή διαθέσιμη η τιμή του ώστε να ανακτηθούν από τη βάση δεδομένων πληροφορίες που τον αφορούν (μετρήσεις, προσωπικές πληροφορίες και λοιπά).

Ο χρήστης παραμένει πιστοποιημένος (authenticated) στην εφαρμογή μέχρι να συμβεί ένα από τα παρακάτω:

1. Λήξη του session.
2. Κλείσιμο του browser. Εάν ο χρήστης κλείσει μόνο την αντίστοιχη καρτέλα, τότε παραμένει πιστοποιημένος στην εφαρμογή. Εάν όμως κλείσει τον browser τότε το cookie διαγράφεται.
3. Logout από το χρήστη μέσω της εφαρμογής. Περιγράφεται σε επόμενη ενότητα.

## 5.4 Κεντρική σελίδα της εφαρμογής

Σε αυτήν τη σελίδα ανακατευθύνεται ο χρήστης αμέσως μετά την επιτυχημένη σύνδεση (login). Στη Home Page της εφαρμογής παρουσιάζεται ένας πίνακας με τις τελευταίες μετρήσεις που ελήφθησαν από τις κυψέλες του χρήστη. Επιπλέον, δίνεται η δυνατότητα στο χρήστη να επιβλέψει την τρέχουσα κατάσταση στις κυψέλες του.



Beehive Monitoring

Charts Settings Profile Log Off (Maria Karouta)

Current conditions in the Beehives

Beehive Name	Temperature (°C)	Humidity (%)	Weight (kg)
N100	22	60	12,7
N101	24	35	42
N102	24	42	22,9

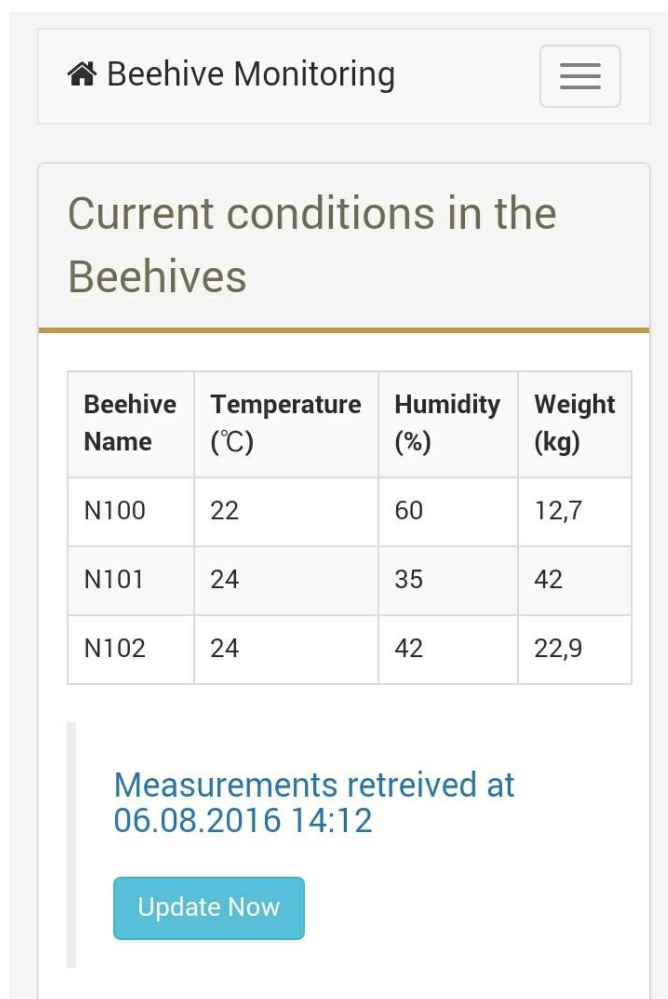
Measurements retrieved at 06.08.2016 14:12

Update Now

© 2016 - Beehive Monitoring

**Εικόνα 13:** Η κεντρική σελίδα της εφαρμογής σε υπολογιστή

Με τη λειτουργία «Update Now» μπορεί να ζητήσει να ληφθούν μετρήσεις από τις κυψέλες του εκείνη τη στιγμή, χωρίς να επηρεαστεί το χρονοδιάγραμμα που ο ίδιος έχει ορίσει για τη λήψη των μετρήσεων.

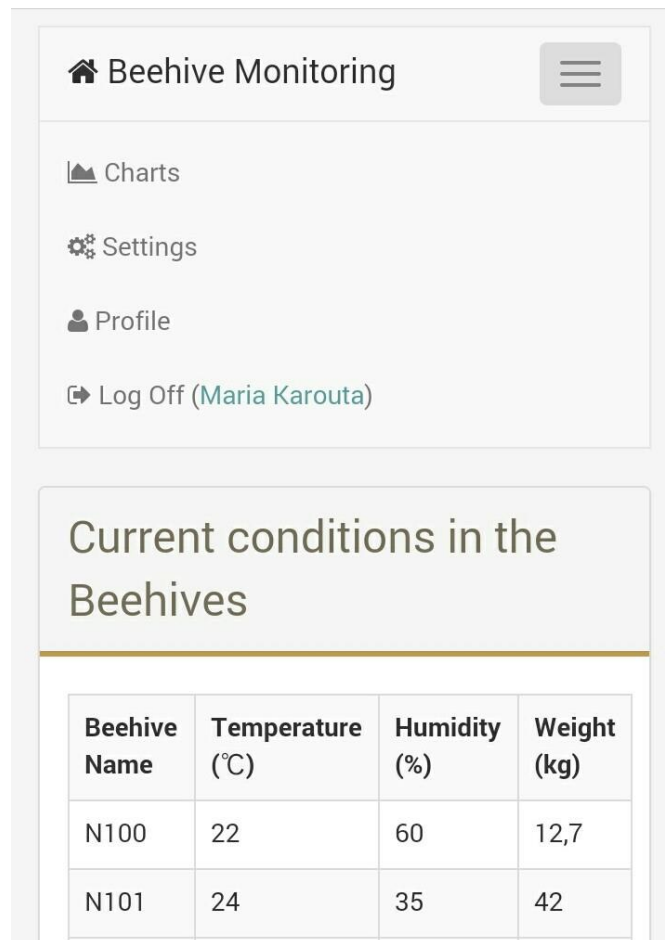


**Εικόνα 14: Η κεντρική σελίδα της εφαρμογής σε κινητό τηλέφωνο**

Στην κορυφή της ιστοσελίδας εμφανίζεται πλέον το μενού, το οποίο όπως είναι φυσικό δεν ήταν ορατό σε μη συνδεδεμένους χρήστες στη Login σελίδα. Περιλαμβάνει τις εξής σελίδες:

- **Charts.** Σε αυτήν τη σελίδα ο χρήστης μπορεί να μελετήσει διαγράμματα της εξέλιξης των μεγεθών βάρους, υγρασίας και θερμοκρασίας στις κυψέλες του στο πέρασμα του χρόνου.
- **Settings.** Σε αυτήν τη σελίδα ο χρήστης μπορεί να εξετάσει τις ρυθμίσεις του και να τις επεξεργαστεί.
- **Profile.** Σε αυτήν τη σελίδα ο χρήστης μπορεί να αλλάξει ορισμένα από τα στοιχεία του.

- Log Off. Με αυτό το link μπορεί να αποσυνδεθεί από την εφαρμογή.
- Logo – «Beehive Monitoring». Οδηγεί στη Home Page της εφαρμογής.



**Εικόνα 15: Το μενού της εφαρμογής σε κινητό τηλέφωνο**

Στο στιγμιότυπο οθόνης από το κινητό τηλέφωνο το μενού είναι μη ορατό. Με το κουμπί πάνω δεξιά στην οθόνη εμφανίζεται/εξαφανίζεται, όπως φαίνεται στην Εικόνα 15.

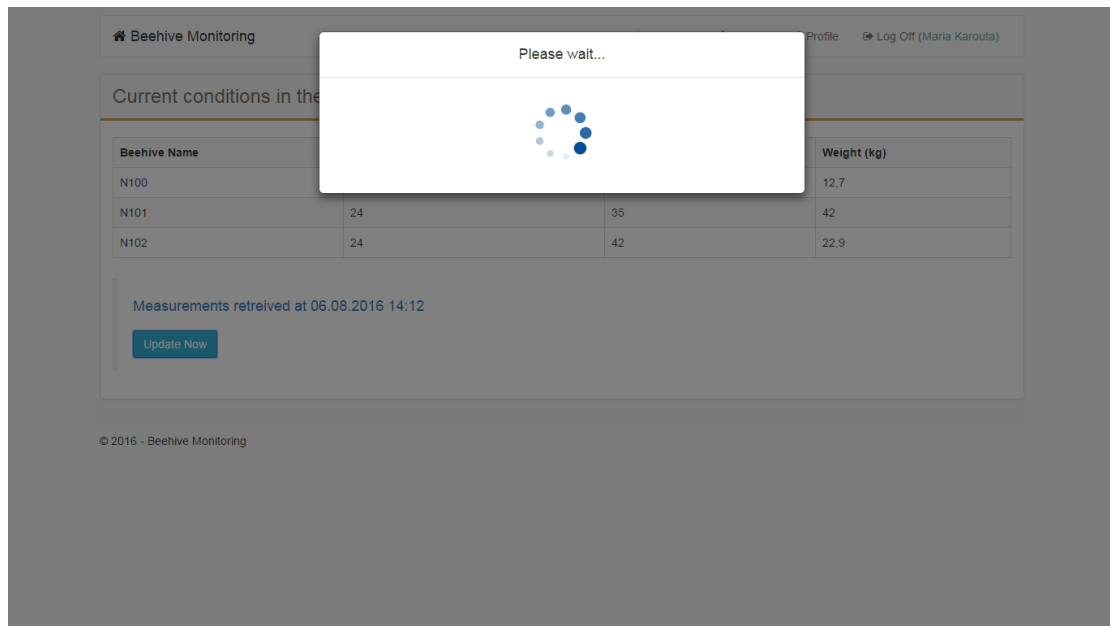
Το μοντέλο της Home Page είναι μία κλάση που περιλαμβάνει μία λίστα από κυψέλες (Beehives). Κάθε μία κυψέλη διατηρεί τις εξής πληροφορίες: όνομα και πληροφορίες για τις τελευταίες μετρήσεις που ελήφθησαν σε αυτή, δηλαδή θερμοκρασία (σε βαθμούς Κελσίου), υγρασία (σε ποσοστό επί τοις εκατό), βάρος (σε κιλά) καθώς και τη χρονική στιγμή που ελήφθησαν οι μετρήσεις από τους αισθητήρες και στάλθηκαν στον Gateway της μελισσοσμήνης.



Το «Update Now» αποτελεί μία από τις σημαντικότερες λειτουργικότητες της εφαρμογής, αφού δίνει στο μελισσοκόμο τη δυνατότητα να εξετάσει την τρέχουσα κατάσταση των κυψελών του οποιαδήποτε στιγμή, όπου και αν βρίσκεται. Με το πάτημα του κουμπιού, πραγματοποιείται μία ajax κλήση προς το server ο οποίος αναλαμβάνει να εκκινήσει τη διαδικασία λήψης των μετρήσεων. Η διαδικασία αυτή είναι σχεδόν όμοια με τη διαδικασία που ακολουθείται κατά την προγραμματισμένη λήψη των μετρήσεων από τον server. Αρχικά, ανακτάται από τη βάση δεδομένων το wsn mobile του χρήστη και πραγματοποιείται αναπάντητη κλήση προς αυτόν τον αριθμό. Η κλήση αυτή έχει ως αποτέλεσμα την αφύπνιση του Gateway ο οποίος θα επιχειρήσει socket connection με το server προκειμένου να του στείλει τις μετρήσεις. Η socket σύνδεση επιτυγχάνεται με το Dynamic DNS. Ο Gateway χρησιμοποιώντας το DDNS όνομα σε συνδυασμό με μία προκαθορισμένη θύρα μπορεί να συνδεθεί με το server. Αφού λοιπόν πραγματοποιηθεί η μεταξύ τους σύνδεση, ξεκινά η αποστολή μετρήσεων από τον Gateway και κατόπιν η λήψη και αποθήκευσή τους από το server στη βάση δεδομένων. Μόλις ολοκληρωθεί η διαδικασία αυτή ανανεώνεται αυτόματα η Home Page δείχνοντας τις νέες μετρήσεις.

Αξίζει να σημειωθεί σε αυτό το σημείο ότι το schedule του χρήστη για λήψη μετρήσεων δεν επηρεάζεται από τη λήψη μετρήσεων κατ' απαίτησή του μέσα από αυτήν τη σελίδα. Σε αντίθεση με την προγραμματισμένη λήψη μετρήσεων, σε αυτήν δεν ελέγχονται τα settings του χρήστη ώστε να ενημερωθεί για επείγουσες καταστάσεις, αφού θα ελέγξει εκείνη τη στιγμή ο ίδιος την κατάσταση των κυψελών.

Η λήψη των μετρήσεων διαρκεί ένα λεπτό περίπου λόγω των αρκετών ενεργειών που πρέπει να διεξαχθούν. Κρίθηκε λοιπόν αναγκαίο σε αυτήν τη στιγμή να εμφανίζεται ένα παράθυρο στο χρήστη που θα τον αποτρέπει από το να κάνει οποιαδήποτε ενέργεια μέσα στην ιστοσελίδα, έως ότου ολοκληρωθεί η εν λόγω διαδικασία. Χρησιμοποιώντας τις ποικίλλες δυνατότητες που παρέχει το bootstrap, εμφανίζεται ένα modal όσο εξελίσσεται η διαδικασία αυτή, όπως φαίνεται στην Εικόνα 16.



**Εικόνα 16: Οθόνη αναμονής απόκρισης του server**

Η διαδικασία λήψης μετρήσεων μέσα από το Web Interface μπορεί να αποτύχει. Αυτό γιατί υπάρχει πιθανότητα ο server να είναι απασχολημένος εκείνη τη στιγμή εκτελώντας το προγραμματισμένο schedule λήψης των μετρήσεων όλων των χρηστών, όπως περιγράφηκε σε προηγούμενο κεφάλαιο. Σε αυτήν την περίπτωση εμφανίζεται μήνυμα στο χρήστη ότι ο server είναι απασχολημένος και να προσπαθήσει ξανά αργότερα. Οι προσπάθειες των χρηστών να λάβουν μετρήσεις μέσω του Web Interface καταγράφονται σε log αρχείο στο server, έτσι ώστε να ελέγχεται από τον administrator του συστήματος κατά πόσο είναι επιτυχής αυτή η λειτουργικότητα. Σε περίπτωση που υπάρχει συνωστισμός στο δίκτυο γιατί πολλοί χρήστες επιχειρούν ταυτόχρονα λήψη μετρήσεων θα υπάρξει επανασχεδιασμός αυτής της λειτουργίας, ώστε να μπορούν να εξυπηρετηθούν πολλοί χρήστες μαζί.

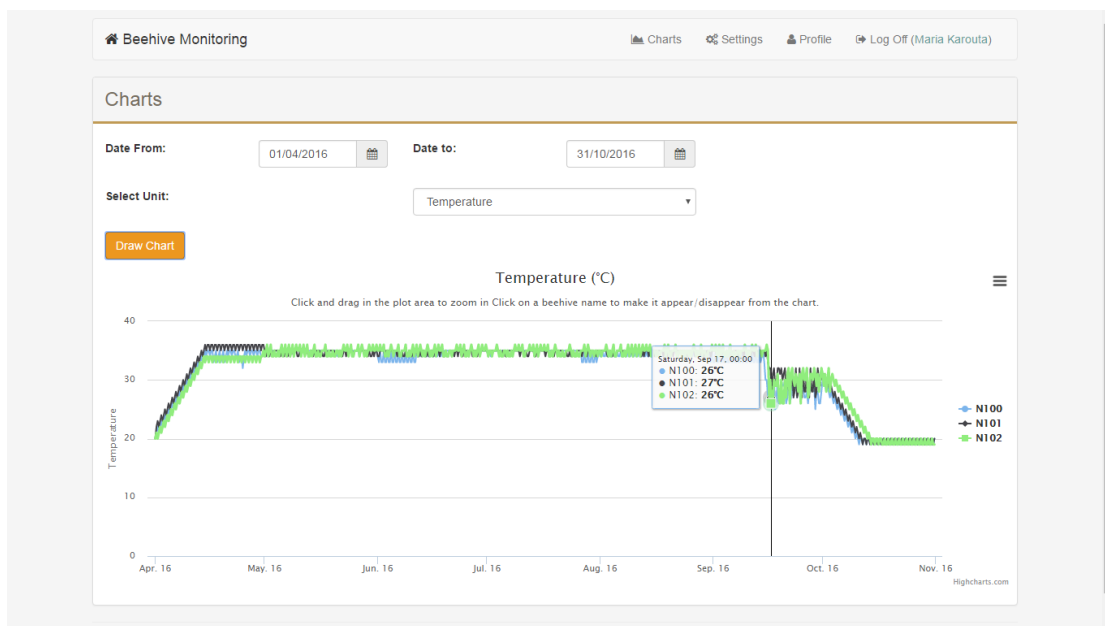
## ***5.5 Γραφική αναπαράσταση των μετρήσεων***

Ένα από τα σημαντικότερα γνωρίσματα της εφαρμογής αυτής είναι η αποθήκευση των μετρήσεων που λαμβάνονται από τους αισθητήρες των κυψελών σε βάση δεδομένων. Με αυτόν τον τρόπο συγκεντρώνεται ένα ιστορικό για τις συνθήκες που επικρατούν στο μελισσοκομείο του κάθε μελισσοκόμου, το οποίο θα μπορούσε να το αξιοποιήσει κάθε ένας χωριστά ώστε να βελτιώσει την παραγωγή του.

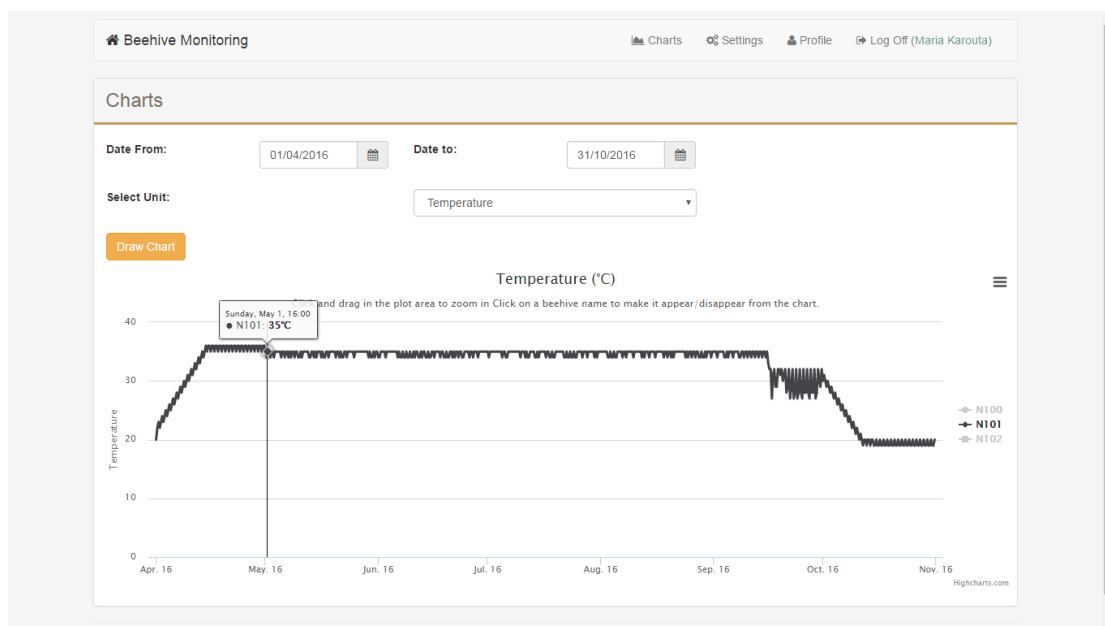
Οι μελισσοκόμοι – χρήστες της εφαρμογής, έχουν τη δυνατότητα να προβάλλουν και να εξετάσουν το ιστορικό των μετρήσεων που έχουν ληφθεί από τους αισθητήρες σε κάθε κυψέλη τους μέσα από τον ιστότοπο της εφαρμογής. Στη σελίδα «Charts» του ιστότοπου ο χρήστης έχει τη δυνατότητα να προβάλλει γραφικές παραστάσεις εξέλιξης της θερμοκρασίας, της υγρασίας και του βάρους στις κυψέλες του σε βάθος χρόνου. Όταν ο χρήστης επισκεφθεί τη σελίδα αυτή, αρχικά προβάλλεται ένα γράφημα με τόσες γραμμές όσες οι κυψέλες του χρήστη οι οποίες δείχνουν τις τιμές του αισθητήρα θερμοκρασίας για διάστημα ενός μήνα. Κατόπιν μπορεί να προσαρμόσει ο ίδιος τη γραφική παράσταση με τα κριτήρια που επιθυμεί. Θα πρέπει να προσδιορίσει λοιπόν το φίλτρο «Select Unit» για το μέγεθος που τον ενδιαφέρει να εξετάσει, επιλέγοντας μεταξύ Temperature, Humidity & Weight, καθώς και τα φίλτρα «Date from» και «Date To» για το χρονικό διάστημα μέσα στο οποίο επιθυμεί να εξετάσει την εξέλιξη αυτού του μεγέθους. Με κλικ στο «Draw Chart» κουμπί, πραγματοποιείται μία ajax κλήση προς το server η οποία επιστρέφει τα δεδομένα που ανταποκρίνονται στα κριτήρια που προσδιόρισε ο χρήστης και με βάση αυτά ζωγραφίζεται η γραφική παράσταση. Για κάθε μία από τις κυψέλες υπάρχει μία ξεχωριστή γραμμή στο γράφημα που δείχνει την εξέλιξη του εξεταζόμενου μεγέθους. Με πέρασμα του κέρσορα του ποντικιού πάνω από ένα σημείο της γραφικής παράστασης εμφανίζεται tooltip που δείχνει τις τιμές και την ακριβή ημερομηνία λήψης των εν λόγω μετρήσεων σε όλες τις κυψέλες. Επιπλέον, ο χρήστης μπορεί εύκολα να προσθέσει και να αφαιρέσει τις κυψέλες που επιθυμεί να φαίνονται στο γράφημα κάνοντας κλικ στο όνομά τους.

Οι γραφικές παραστάσεις δημιουργήθηκαν κάνοντας χρήση των βιβλιοθηκών του Highcharts. Το Highcharts (<http://www.highcharts.com/>) είναι ένα προϊόν ανεπτυγμένο σε κώδικα JavaScript. Αποτελεί εργαλείο που χρησιμοποιείται για την παρουσίαση διαφόρων αποτελεσμάτων σε μορφή διαδραστικών γραφικών παραστάσεων σε web ή mobile εφαρμογές και παρέχεται δωρεάν για μη κερδοσκοπικούς σκοπούς. [8] Παρέχει επιπλέον τη δυνατότητα απευθείας εκτύπωσης ενός διαγράμματος ή αποθήκευσής του σε μορφές PNG, JPEG, PDF και SVG.

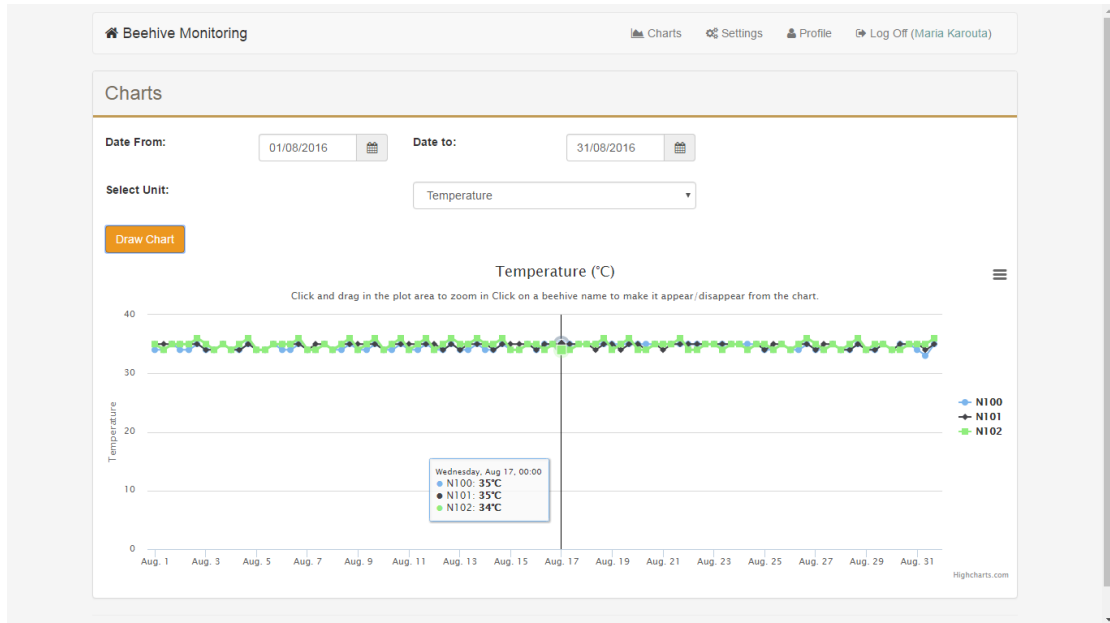
- ❖ Στιγμιότυπα από τη σελίδα «Charts» σε υπολογιστή για τη *θερμοκρασία* σε όλες τις κυψέλες ενός χρήστη για τους μήνες Απρίλιο έως Οκτώβριο 2016 (Εικόνα 17), για την κυψέλη N101 για τους ίδιους μήνες (Εικόνα 18), για όλες τις κυψέλες το μήνα Αύγουστο 2016 (Εικόνα 19) και για την κυψέλη N100 το μήνα Αύγουστο 2016 (Εικόνα 20). Ακολουθεί ένα στιγμιότυπο από κινητό τηλέφωνο για τη θερμοκρασία σε όλες τις κυψέλες το μήνα Σεπτέμβριο 2016 (Εικόνα 21).



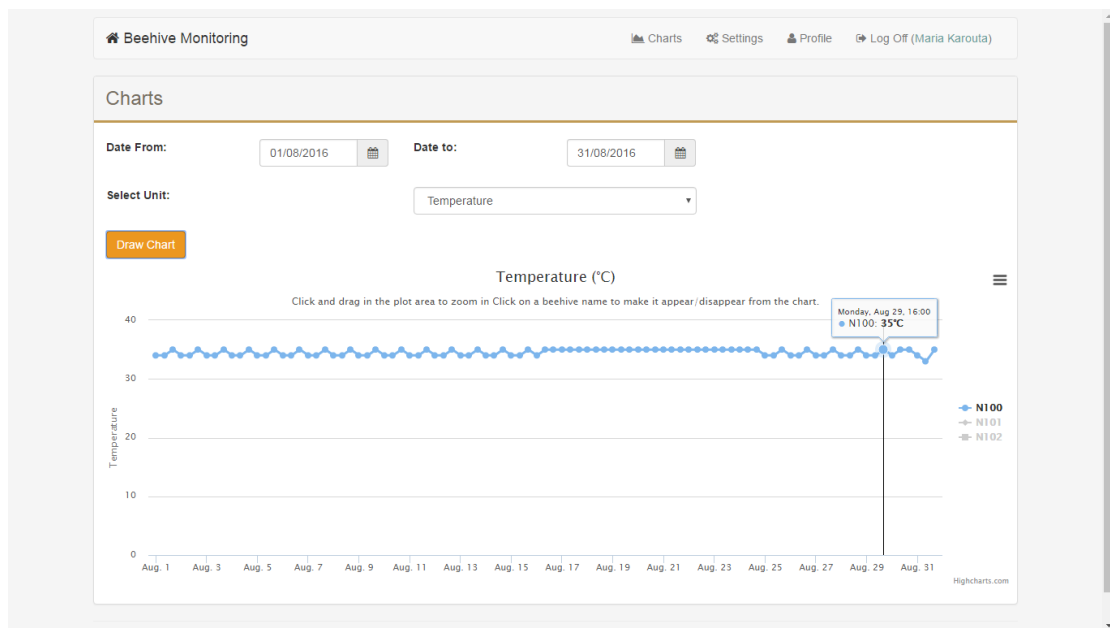
**Εικόνα 17: Θερμοκρασία των κυψελών για το διάστημα 04/2016-10/2016**



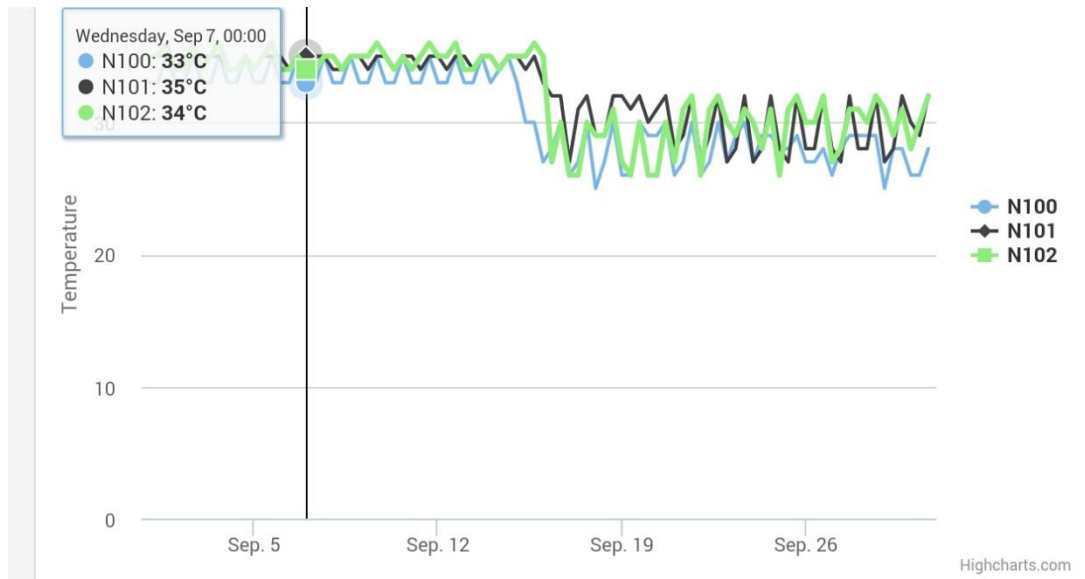
**Εικόνα 18: Θερμοκρασία μίας κυψέλης (N101) για το διάστημα 04/2016-10/2016**



**Εικόνα 19: Θερμοκρασία των κυψελών για το μήνα Αύγουστο 2016**

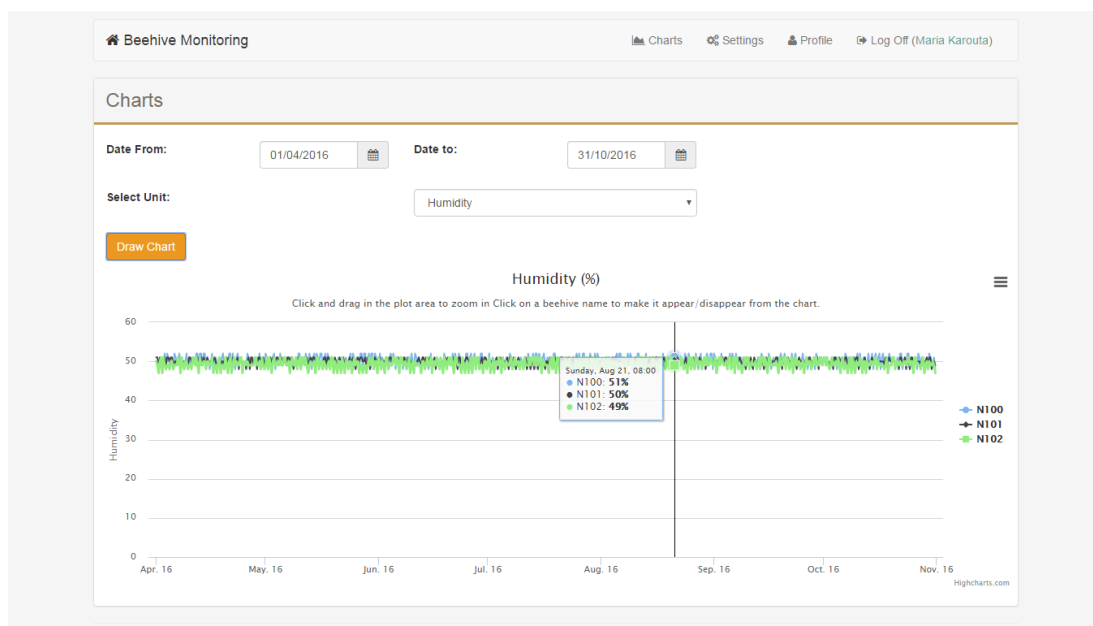


**Εικόνα 20: Θερμοκρασία μίας κυψέλης (N100) για το μήνα Αύγουστο 2016**

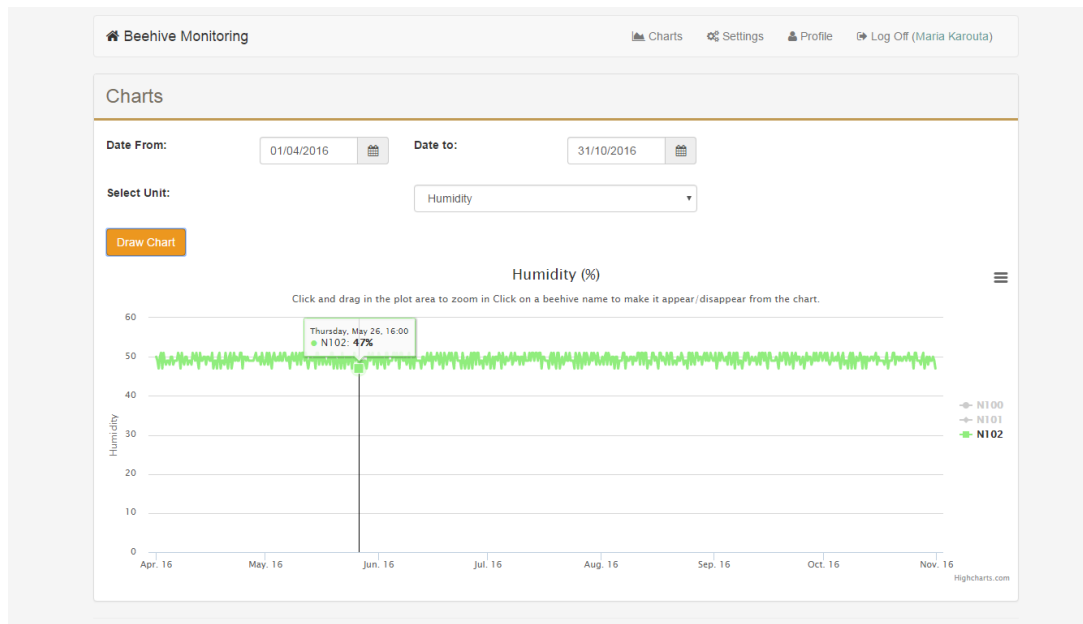


**Εικόνα 21: Θερμοκρασία των κυψελών για το μήνα Σεπτέμβριο 2016 (στιγμιότυπο από κινητό τηλέφωνο)**

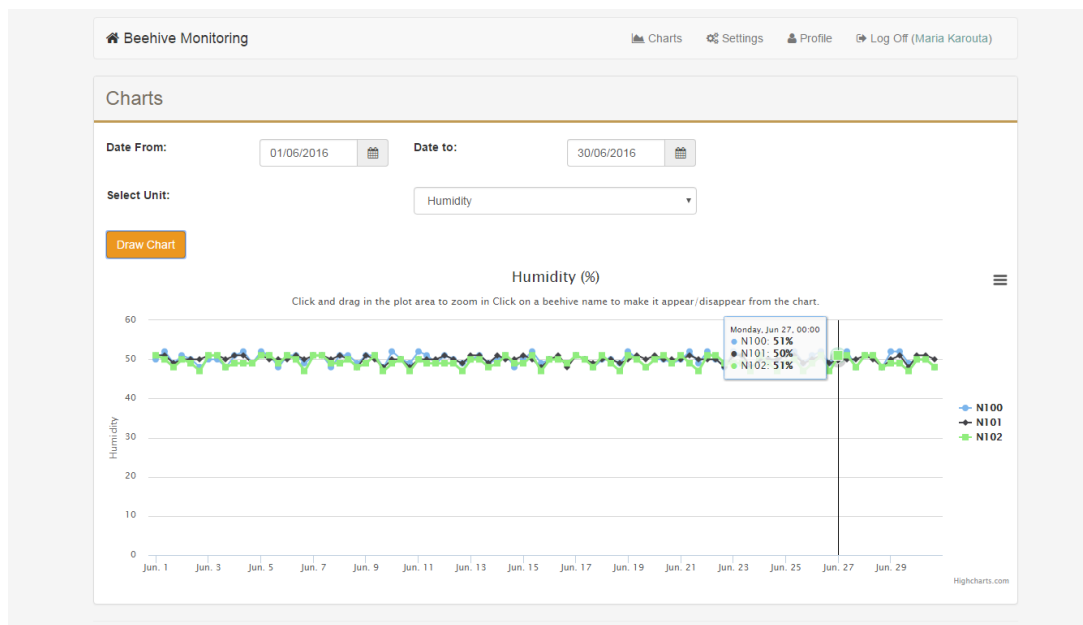
- ❖ Στιγμιότυπα από τη σελίδα «Charts» σε υπολογιστή για την υγρασία σε όλες τις κυψέλες ενός χρήστη για τους μήνες Απρίλιο έως Οκτώβριο 2016 (Εικόνα 22), για την κυψέλη N102 για τους ίδιους μήνες (Εικόνα 23), για όλες τις κυψέλες το μήνα Ιούνιο 2016 (Εικόνα 24) και για την κυψέλη N100 το μήνα Ιούνιο 2016 (Εικόνα 25). Ακολουθεί ένα στιγμιότυπο από κινητό τηλέφωνο για την υγρασία σε όλες τις κυψέλες το μήνα Οκτώβριο 2016 (Εικόνα 26).



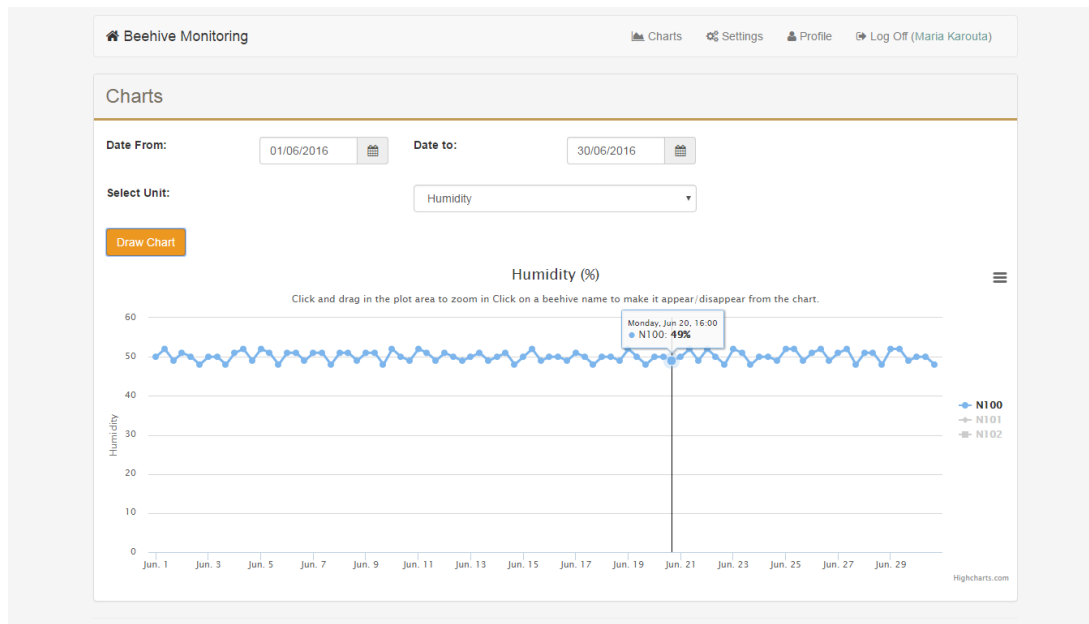
**Εικόνα 22: Υγρασία σε όλες τις κυψέλες για το διάστημα 04/2016-10/2016**



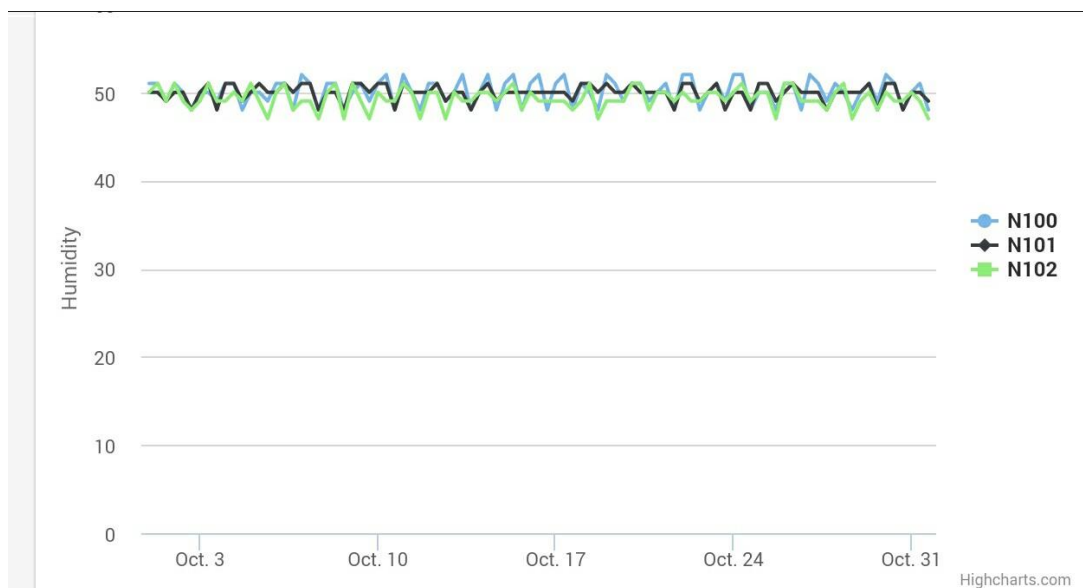
**Εικόνα 23: Υγρασία σε μία κυψέλη (N102) για το διάστημα 04/2016-10/2016**



**Εικόνα 24: Υγρασία σε όλες τις κυψέλες για το μήνα Ιούνιο 2016**



**Εικόνα 25: Υγρασία σε μία κυψέλη (N100) για το μήνα Ιούνιο 2016**

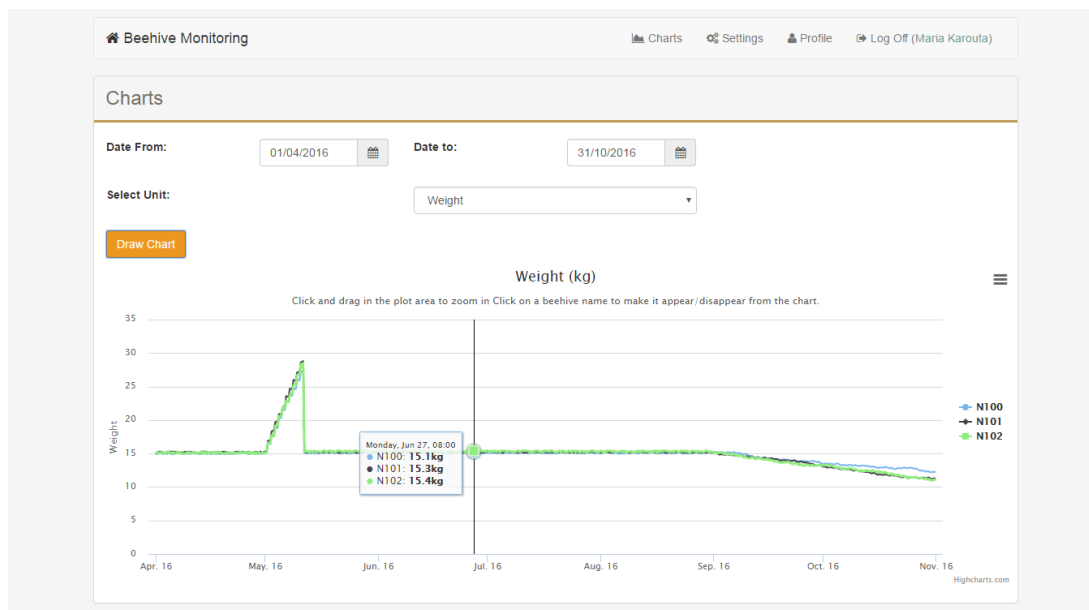


**Εικόνα 26: Υγρασία σε όλες τις κυψέλες για το μήνα Οκτώβριο 2016 (στιγμιότυπο από κινητό τηλέφωνο)**

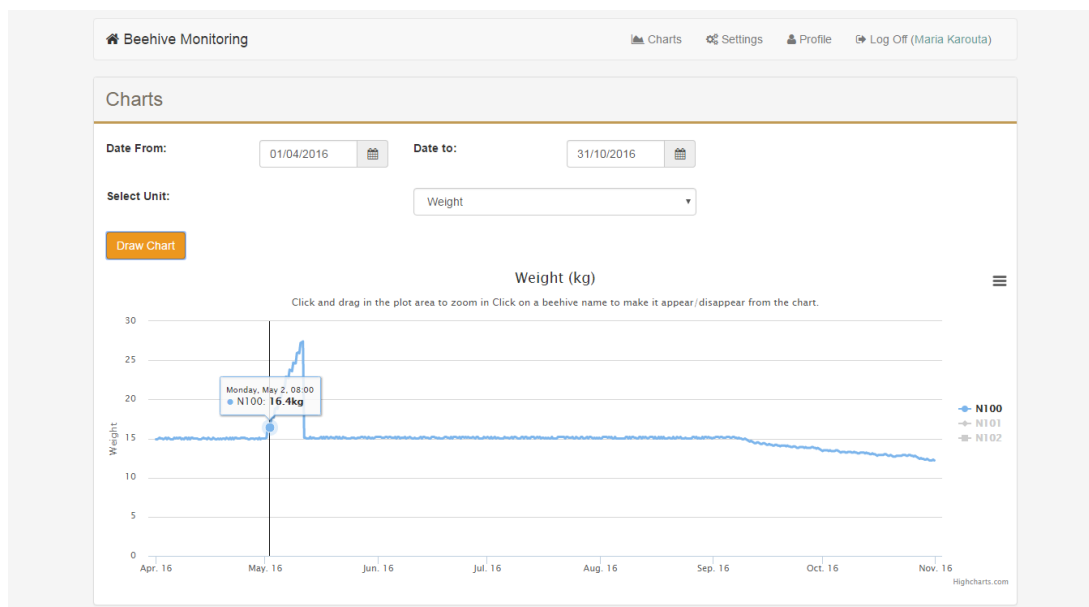
- ❖ Στιγμιότυπα από τη σελίδα «Charts» σε υπολογιστή για το βάρος σε όλες τις κυψέλες ενός χρήστη για τους μήνες Απρίλιο έως Οκτώβριο 2016 (Εικόνα 27), για την κυψέλη N100 για τους ίδιους μήνες (Εικόνα 28), για όλες τις κυψέλες το μήνα Μάιο 2016 (Εικόνα 29) και για την κυψέλη N101 το μήνα Μάιο 2016 (Εικόνα 30). Ακολουθεί ένα στιγμιότυπο από κινητό τηλέφωνο για το βάρος σε όλες τις



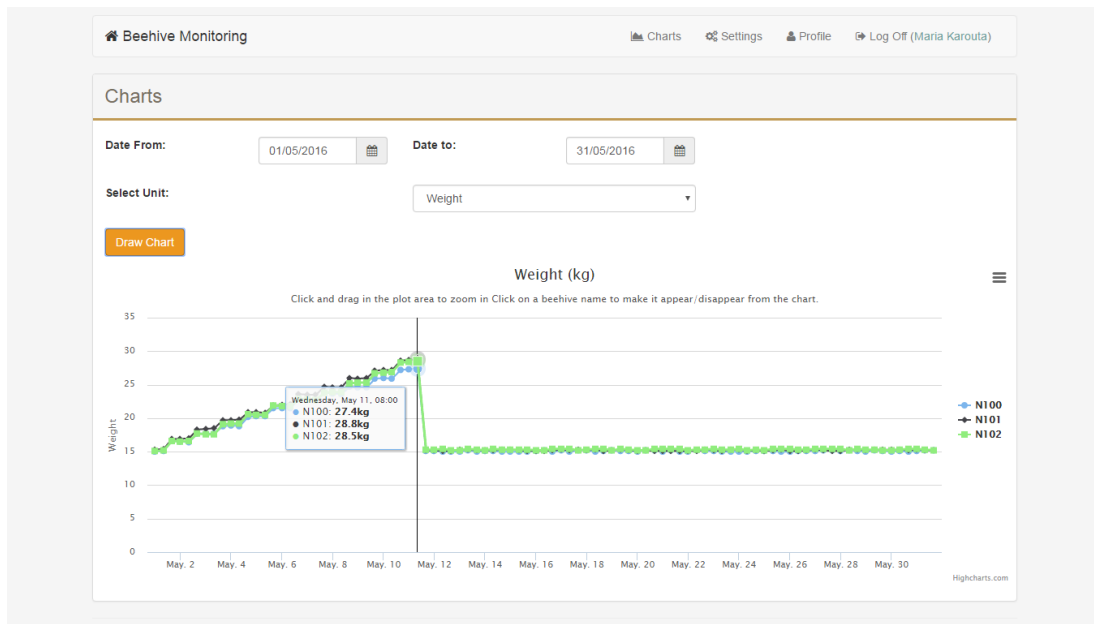
κυψέλες το μήνα Μάιο 2016 (Εικόνα 31). Αξίζει να σημειωθεί για το βάρος ότι θεωρήσαμε πως οι κυψέλες βρίσκονται τοποθετημένες σε πορτοκαλιές, συνεπώς η ανθοφορία τους ξεκινά τέλη Απριλίου με αρχές Μαΐου και διαρκεί για δέκα ημέρες. Το βάρος της κυψέλης κάθε μέρα κατά την περίοδο αυτή αυξάνεται περίπου 1-1.5 κιλά. Παρατηρούμε επίσης μία μικρή μείωση του βάρους των κυψελών το μήνα Οκτώβρη, διότι αρχίζει να μειώνεται ο πληθυσμός τους σε μέλισσες.



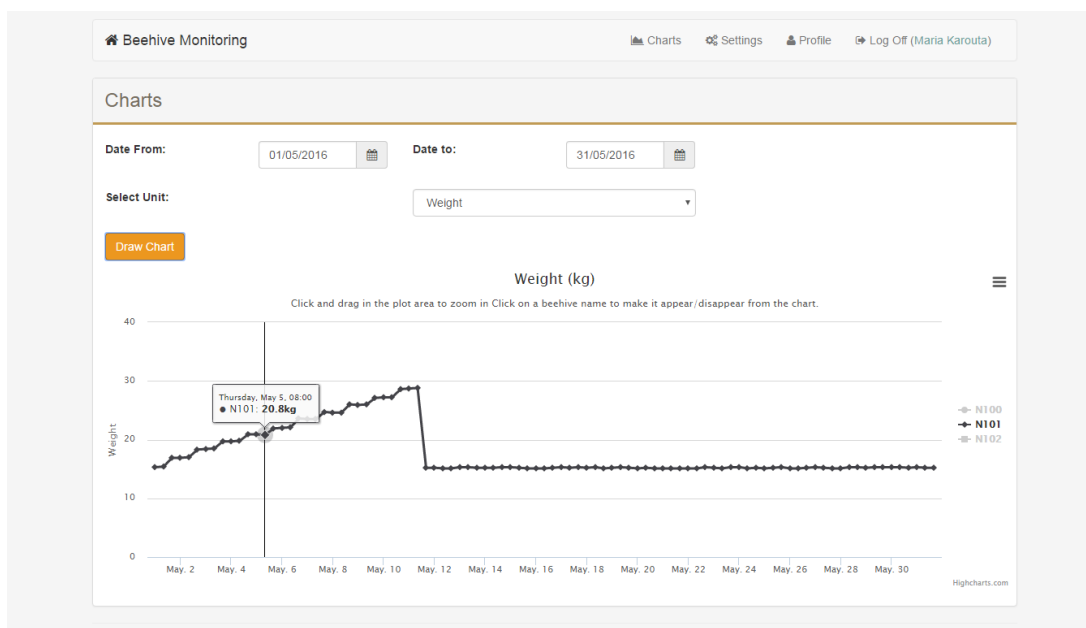
**Εικόνα 27: Το βάρος των κυψελών για το διάστημα 04/2016-10/2016**



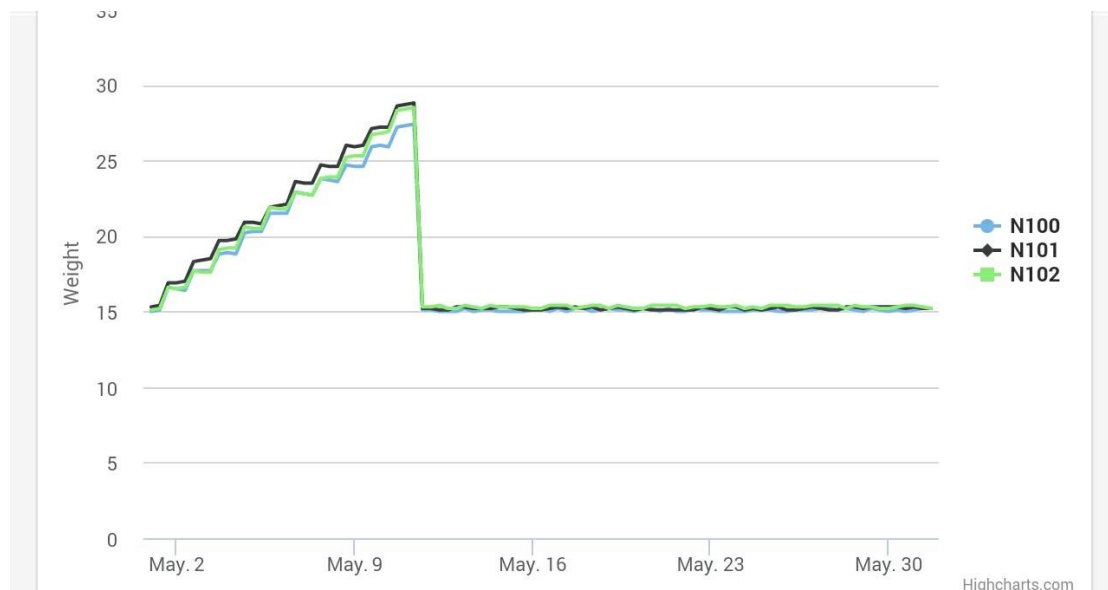
**Εικόνα 28: Το βάρος μιας κυψέλης (N100) για το διάστημα 04/2016-10/2016**



**Εικόνα 29: Το βάρος των κυψελών για το μήνα Μάιο 2016**



**Εικόνα 30: Το βάρος μιας κυψέλης (N101) για το μήνα Μάιο 2016**



**Εικόνα 31: Το βάρος των κυψελών για το μήνα Μάιο 2016 (στιγμιότυπο από κινητό τηλέφωνο)**

## 5.6 Οι ρυθμίσεις του χρήστη

Κάθε χρήστης της εφαρμογής έχει τη δυνατότητα να προβάλλει και να επεξεργαστεί τις ρυθμίσεις του σχετικά με την εφαρμογή. Οι ρυθμίσεις αυτές αφορούν την ενημέρωσή του σε περιπτώσεις ειδικών συνθηκών στις κυψέλες και τον χρονοπρογραμματισμό της λήψης μετρήσεων από τους αισθητήρες των κυψελών. Διακρίνονται λοιπόν δύο κατηγορίες ρυθμίσεων: Notification και Schedule Settings.

### 5.6.1 Notification Settings

Τα Notification Settings περιλαμβάνουν κανόνες που καθορίζουν τις συνθήκες στις κυψέλες κάτω από τις οποίες ο χρήστης θα ενημερωθεί με Email ή/και SMS όταν επικρατούν. Για κάθε ένα εκ των μεγεθών (υγρασία, θερμοκρασία και βάρος) ο χρήστης μπορεί να καθορίσει μία ελάχιστη τιμή κάτω από την οποία θέλει να ενημερωθεί, καθώς και μία μέγιστη τιμή που όταν ξεπεραστεί θέλει να το γνωρίζει. Ακολουθούν μερικά παραδείγματα που αποδεικνύουν τη χρησιμότητα αυτής της λειτουργίας:

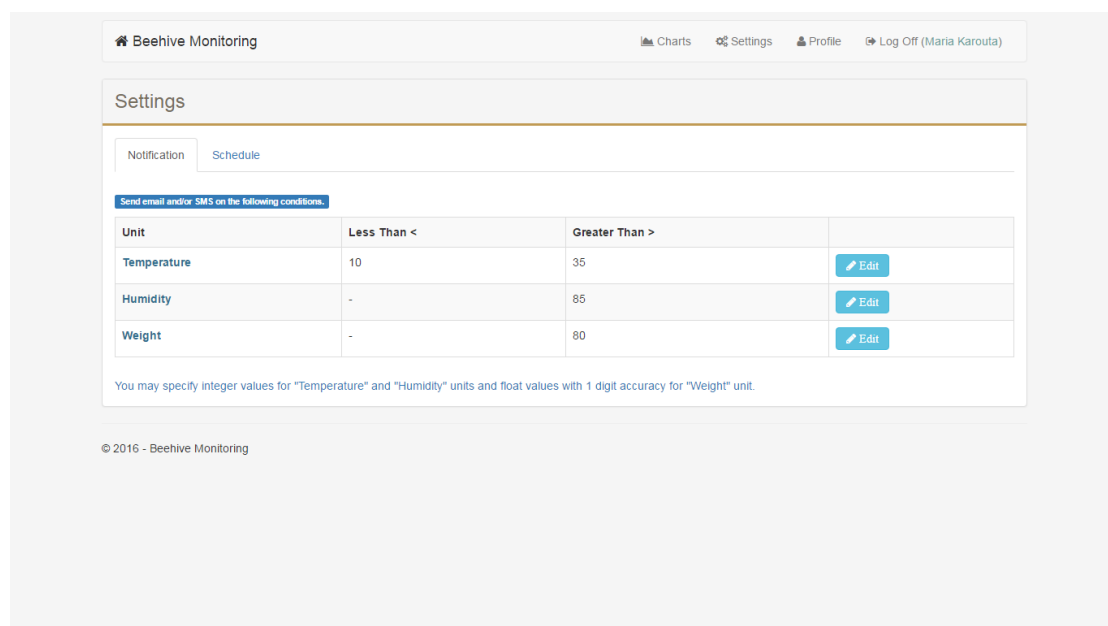
1. Ο μελισσοκόμος γνωρίζει το βάρος που έχει περίπου η κυψέλη όταν η ανθοφορία έχει φτάσει στο τέλος της. Μπορεί να καθορίσει λοιπόν ότι θέλει να ενημερωθεί μέσω SMS/Email όταν το βάρος μιας κυψέλης του ξεπεράσει τα X κιλά.

Με αυτόν τον τρόπο δεν χρειάζεται να παρακολουθεί καθημερινά μέσω της εφαρμογής την εξέλιξη του βάρους στις κυψέλες του και μπορεί να φροντίσει για την έγκαιρη συγκομιδή.

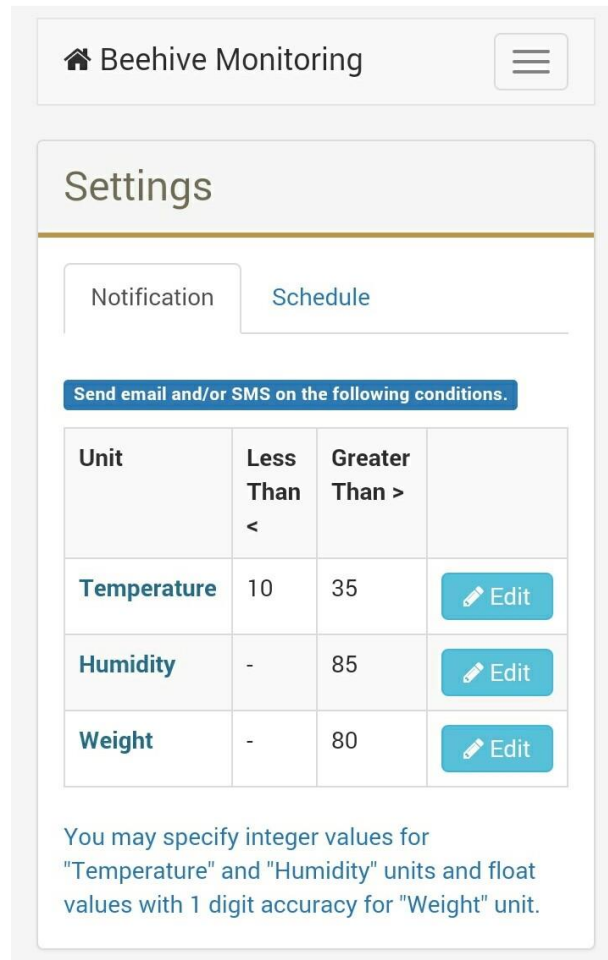
2. Ο μελισσοκόμος γνωρίζει ότι η περίοδος ανάπτυξης του μελισσιού ξεκινάει την άνοιξη ή και λίγο νωρίτερα, όταν παρατηρηθεί θερμοκρασία εντός του μελισσιού γύρω στους 35 βαθμούς Κελσίου. [1] Τότε θα πρέπει να ελέγξει τα αποθέματα τροφής στο μελίσσι. Συνεπώς, για διευκόλυνσή του, μπορεί να καθορίσει ότι θέλει να ενημερωθεί όταν η θερμοκρασία στην κυψέλη ξεπεράσει τους 35 βαθμούς.
3. Πολύ αυξημένα επίπεδα υγρασίας εντός της κυψέλης σε συνδυασμό με χαμηλές θερμοκρασίες μπορεί να αποτελέσουν κίνδυνο για το μελίσσι. [1] Καθορίζοντας κάποιες ενδεικτικές τιμές για ελάχιστη θερμοκρασία και μέγιστη υγρασία εντός της κυψέλης θα βοηθούσαν στην καλύτερη παρακολούθησή της και την πρόληψη των κινδύνων που ελλοχεύουν.

Αρχικά οι κανόνες για τα Notification Settings στην εφαρμογή δεν έχουν τιμές για τα εξεταζόμενα μεγέθη (υγρασία, θερμοκρασία, βάρος), και μόνο εάν ο χρήστης το επιθυμεί θα τους δώσει τιμές, ανάλογα με τις ανάγκες του.

Παρακάτω φαίνεται ένα στιγμιότυπο από την εφαρμογή σε υπολογιστή (Εικόνα 32) αλλά και κινητό τηλέφωνο (Εικόνα 33) για τα Notification Settings. Στη σελίδα αυτή ο χρήστης μπορεί να μεταβεί μέσα από τα «Settings» στο Navigation Bar.



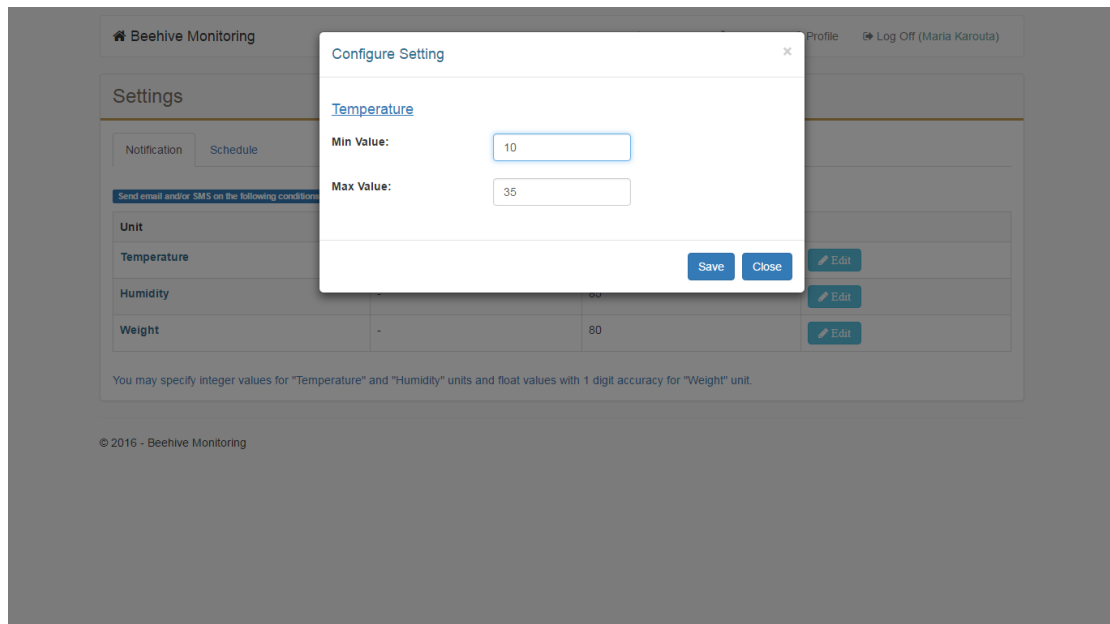
**Εικόνα 32: Στιγμιότυπο οθόνης για τα Notification Settings σε υπολογιστή**



**Εικόνα 33: Στιγμιότυπο οθόνης για τα Notification Settings σε κινητό τηλέφωνο**

Στη σελίδα αυτή παρουσιάζεται ένας πίνακας με τις τιμές που έχουν οριστεί από το χρήστη για τα τρία μεγέθη (θερμοκρασία, υγρασία και βάρος) ως οριακές καταστάσεις στις κυψέλες. Στην 1<sup>η</sup> κολόνα ορίζεται το μέγεθος που εξετάζεται, στη 2<sup>η</sup> κολόνα ορίζεται η ελάχιστη τιμή του εξεταζόμενου μεγέθους κάτω από την οποία ο χρήστης πρέπει να ειδοποιηθεί και στην 3<sup>η</sup> κολόνα ορίζεται η μέγιστη τιμή του μεγέθους πάνω από την οποία ο χρήστης πρέπει να ειδοποιηθεί.

Ο χρήστης μπορεί να επεξεργαστεί τις οριακές τιμές των τριών μεγεθών όπως επιθυμεί. Με κλικ στο «Edit» κουμπί ενός εκ των μεγεθών, ανοίγει ένα παράθυρο διαλόγου όπου ο χρήστης μπορεί να αλλάξει τις τιμές Min Value και Max Value για το αντίστοιχο μέγεθος, όπως φαίνεται στην Εικόνα 34. Με «Save» αποθηκεύονται οι αλλαγές του, ενώ με «Close» απορρίπτονται. Με το κλείσιμο του παραθύρου, ανανεώνονται οι τιμές στον πίνακα εμφανίζοντας τις αλλαγές που έκανε ο χρήστης.



**Εικόνα 34: Στιγμιότυπο οθόνης κατά την επεξεργασία μίας ρύθμισης**

Σε αυτό το σημείο πρέπει να επισημανθεί ότι γίνεται έλεγχος του input που δίνει ο χρήστης για τα Min και Max Values, πριν αποθηκευτούν οι αλλαγές στη βάση δεδομένων. Οι έλεγχοι που γίνονται είναι οι εξής:

1. Ελέγχεται ότι δόθηκαν ακέραιοι αριθμοί για τη θερμοκρασία και την υγρασία, κατ' αντιστοιχία με τις τιμές που λαμβάνονται ως μετρήσεις από τους αισθητήρες.
2. Ελέγχεται ότι δόθηκαν ακέραιοι ή δεκαδικοί αριθμοί για το βάρος, με ακρίβεια ενός δεκαδικού ψηφίου, κατ' αντιστοιχία με τις τιμές που λαμβάνονται ως μετρήσεις από τον αισθητήρα βάρους.
3. Για την υγρασία, ελέγχεται ότι οι τιμές που δόθηκαν από το χρήστη είναι μεταξύ 0 και 100, αφού η υγρασία μετριέται σε ποσοστό επί τοις εκατό.
4. Για το βάρος, ελέγχεται ότι δε δόθηκε αρνητική τιμή.

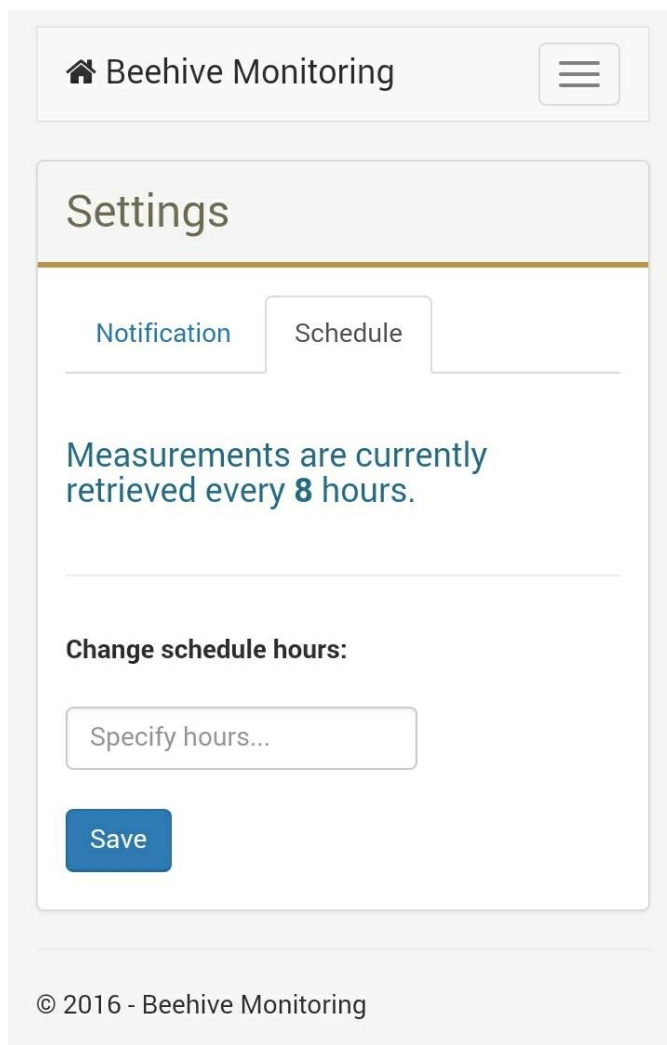
Σε περίπτωση που δεν ικανοποιείται κάποιος από τους παραπάνω περιορισμούς, το σύστημα εμφανίζει μήνυμα σφάλματος στο χρήστη και δεν προχωρά σε αποθήκευση των αλλαγών στη βάση δεδομένων.

Αντίθετα, εάν ικανοποιούνται όλοι οι περιορισμοί, τότε πραγματοποιείται μία ajax post κλήση η οποία αποστέλλει προς το server τις αλλαγές που πραγματοποιήσε ο χρήστης για το συγκεκριμένο setting. Κατόπιν, ο server προχωρά στην ενημέρωση της βάσης δεδομένων με τα νέα στοιχεία.

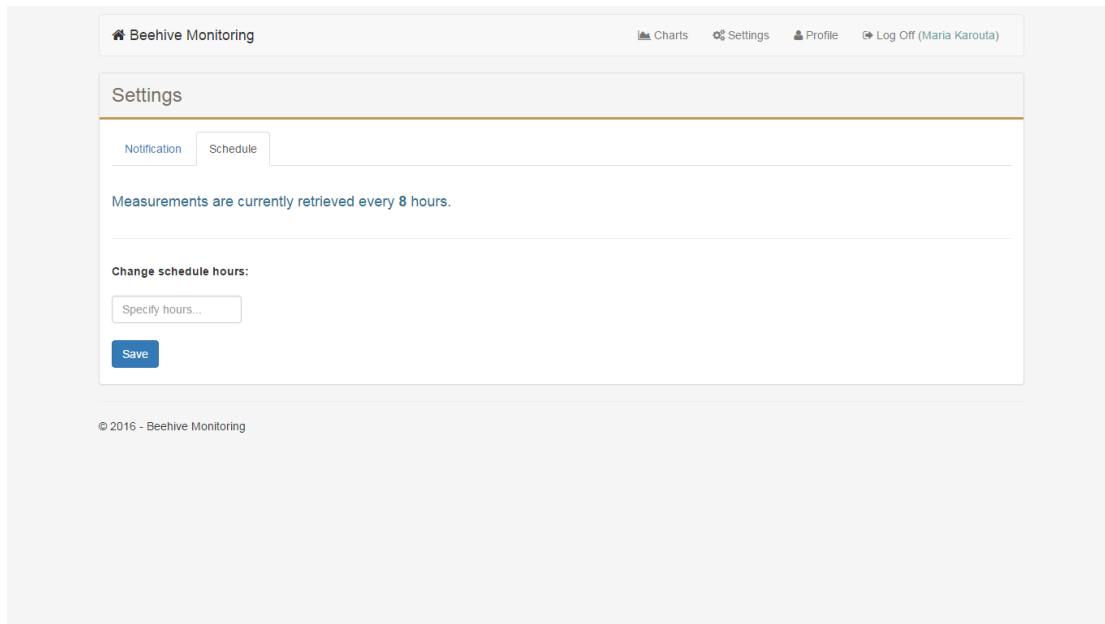
## 5.6.2 Schedule Settings

Τα Schedule Settings δίνουν τη δυνατότητα στο χρήστη να καθορίσει πόσο συχνά επιθυμεί να λαμβάνονται μετρήσεις από τους αισθητήρες στις κυψέλες του. Η προκαθορισμένη τιμή είναι κάθε οκτώ ώρες, αφού μέσα στην ημέρα δεν αλλάζουν ιδιαίτερα οι τιμές των αισθητήρων. Συνεπώς κρίθηκε σκόπιμο να λαμβάνονται μετρήσεις από τις κυψέλες κάθε χρήστη στις 00:00 το βράδυ, 08:00 το πρωί και 16:00 το απόγευμα, όπου μπορεί να παρατηρούνται διαφοροποιήσεις στις τιμές των αισθητήρων. Ο χρήστης μπορεί φυσικά να αλλάξει αυτήν τη ρύθμιση μέσα από αυτήν τη σελίδα ώστε να λαμβάνονται περισσότερο συχνά ή σπάνια μετρήσεις.

Παρακάτω παρατίθεται στιγμιότυπο σε κινητό τηλέφωνο (Εικόνα 35) και σε υπολογιστή (Εικόνα 36) από το 2<sup>ο</sup> tab της σελίδας «Settings» που αφορά τη ρύθμιση του Schedule του χρήστη για τη λήψη μετρήσεων από τις κυψέλες του.



Εικόνα 35: Στιγμιότυπο οθόνης για τα Schedule Settings σε κινητό τηλέφωνο



**Εικόνα 36: Στιγμιότυπο οθόνης για τα Schedule Settings σε υπολογιστή**

Στη σελίδα αυτή φαίνεται η τρέχουσα ρύθμιση του χρήστη για τη λήψη μετρήσεων από το μελίσσι (π.χ. κάθε 8 ώρες), και αμέσως από κάτω δίνεται η δυνατότητα στο χρήστη να αλλάξει την τιμή αυτή. Ο χρήστης προσδιορίζει μία νέα τιμή και με κλικ στο «Save» η τιμή αυτή αποθηκεύεται και αλλάζει στη βάση δεδομένων. Κατόπιν η σελίδα ανανεώνεται ώστε να δείχνει τη νέα τιμή.

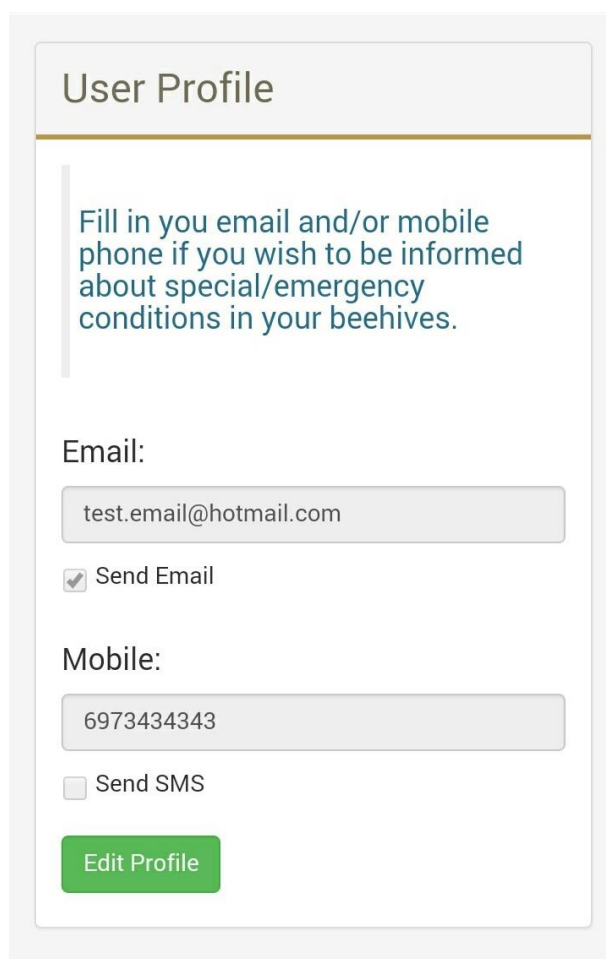
Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, κάθε μία ώρα τρέχει το πρόγραμμα του server μέσα από τον Task Scheduler το οποίο εξετάζει για ποιους χρήστες πρέπει να ληφθούν μετρήσεις, με βάση τις ρυθμίσεις τους. Συνεπώς, κρίθηκε σκόπιμο η ρύθμιση των χρηστών για το schedule να αφορά μόνο φυσικό αριθμό ωρών, διάφορο του μηδενός έτσι ώστε να ανταποκρίνεται με επιτυχία στη ρύθμιση του task του server. Έτσι, με το κλικ του χρήστη στο «Save» πυροδοτούνται οι κατάλληλοι έλεγχοι που αποτρέπουν την αποθήκευση αρνητικής τιμής, δεκαδικού αριθμού ή οτιδήποτε άλλο πέρα από φυσικό αριθμό διάφορο του μηδενός στη βάση δεδομένων. Σε περίπτωση σφάλματος στη ρύθμιση που εισήγαγε ο χρήστης, εμφανίζεται κατάλληλο μήνυμα στην οθόνη. Εάν το schedule που εισήγαγε ο χρήστης είναι αποδεκτή τιμή, τότε πραγματοποιείται μία ajax post κλήση η οποία αποστέλλει την τιμή αυτή στο server. Ο server στη συνέχεια ενημερώνει κατάλληλα τις πληροφορίες του χρήστη στη βάση δεδομένων.



## 5.7 Το προφίλ του χρήστη

Κάθε χρήστης της web εφαρμογής έχει τη δυνατότητα να ορίσει ρυθμίσεις οι οποίες αφορούν την άμεση ενημέρωσή του σε περιπτώσεις ειδικών συνθηκών στις κυψέλες του. Οι συνθήκες αυτές μπορεί να είναι για παράδειγμα υψηλή θερμοκρασία, ή υγρασία και άλλα, όπως περιγράφηκε αναλυτικά στην προηγούμενη ενότητα. Η ενημέρωση του χρήστη πραγματοποιείται είτε με email, είτε με SMS, είτε και με τους δύο τρόπους. Σε αυτήν τη σελίδα της εφαρμογής ο χρήστης μπορεί να ορίσει με ποια ή με ποιες μεθόδους επιθυμεί να ενημερώνεται, καθώς και τα αντίστοιχα στοιχεία του (email address, αριθμός κινητού τηλεφώνου).

Ακολουθούν στιγμιότυπα της σελίδας Profile τόσο σε κινητό τηλέφωνο (Εικόνα 37) όσο και σε υπολογιστή (Εικόνα 38).



User Profile

Fill in you email and/or mobile phone if you wish to be informed about special/emergency conditions in your beehives.

Email:  
test.email@hotmail.com

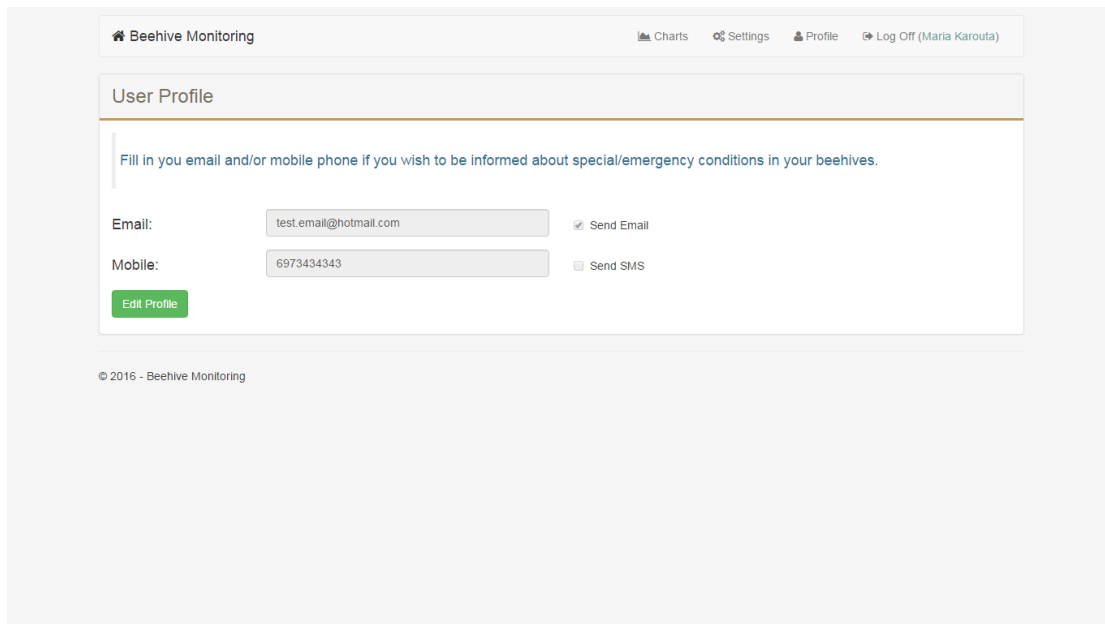
Send Email

Mobile:  
6973434343

Send SMS

Edit Profile

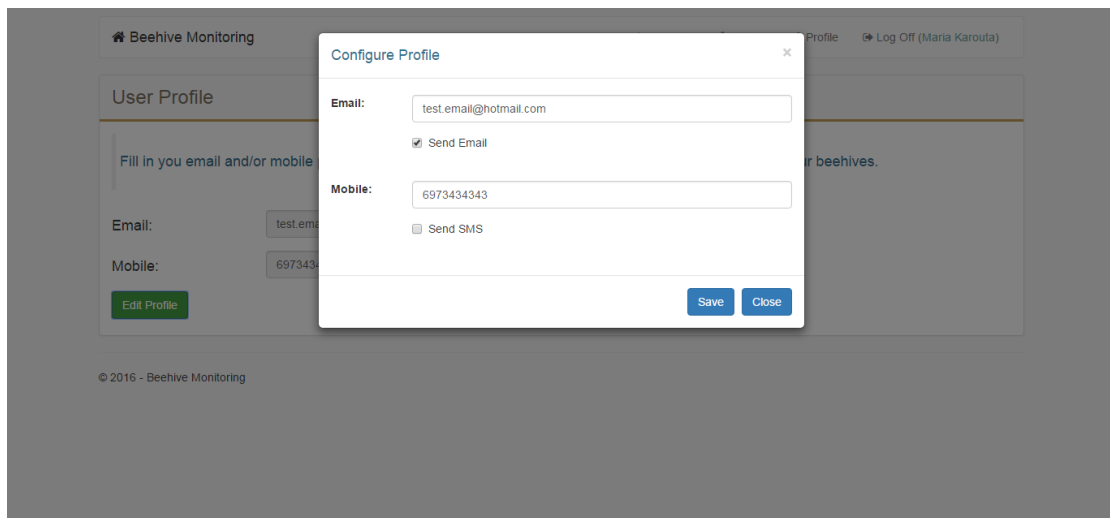
Εικόνα 37: Η οθόνη «Profile» της εφαρμογής σε κινητό τηλέφωνο



**Εικόνα 38: Η οθόνη «Profile» της εφαρμογής σε υπολογιστή**

Στη σελίδα αυτή προβάλλεται το email και το κινητό του χρήστη καθώς και η επιλογή του σχετικά με το εάν επιθυμεί να ενημερώνεται για ειδικές συνθήκες θερμοκρασίας, υγρασίας και βάρους στις κυψέλες του μέσω της μίας ή της άλλης μεθόδου ή και των δύο. Επιπλέον, ο χρήστης έχει τη δυνατότητα να επεξεργαστεί τις τρέχουσες ρυθμίσεις στο προφίλ του κάνοντας κλικ στο «Edit Profile». Με αυτόν τον τρόπο ανοίγει ένα νέο παράθυρο – modal στη σελίδα στο οποίο ο χρήστης μπορεί να επεξεργαστεί τα στοιχεία του, όπως φαίνεται στην Εικόνα 39. Επιλέγοντας αποθήκευση με το «Save» πυροδοτούνται κάποιοι κανόνες που ελέγχουν εάν ο χρήστης έδωσε έγκυρο email και έγκυρο αριθμό τηλεφώνου (δέκα ψηφία). Σε περίπτωση σφάλματος εμφανίζεται κατάλληλο μήνυμα στην οθόνη. Επιπρόσθετα, δεν επιτρέπεται η αποθήκευση αληθούς τιμής για το «Send Email» ή το «Send Mobile» χωρίς να δοθούν τα αντίστοιχα στοιχεία και εμφανίζεται μήνυμα στο χρήστη. Αντιθέτως, το email και το κινητό του χρήστη επιτρέπεται να είναι αποθηκευμένα στο προφίλ του είτε επιθυμεί να χρησιμοποιούνται για επικοινωνία είτε όχι. Εάν όλοι οι παραπάνω έλεγχοι στεφθούν με επιτυχία, τότε πραγματοποιείται μια ajax post κλήση η οποία αποστέλλει με ασφάλεια τις προσωπικές πληροφορίες του χρήστη προς το server. Ο server κατόπιν ενημερώνει τις πληροφορίες του χρήστη στη βάση δεδομένων με τις νέες τιμές.

Κλείνοντας το παράθυρο διαλόγου, η σελίδα ανανεώνεται και δείχνει τις ενημερωμένες ρυθμίσεις του χρήστη για την επικοινωνία με email και SMS.



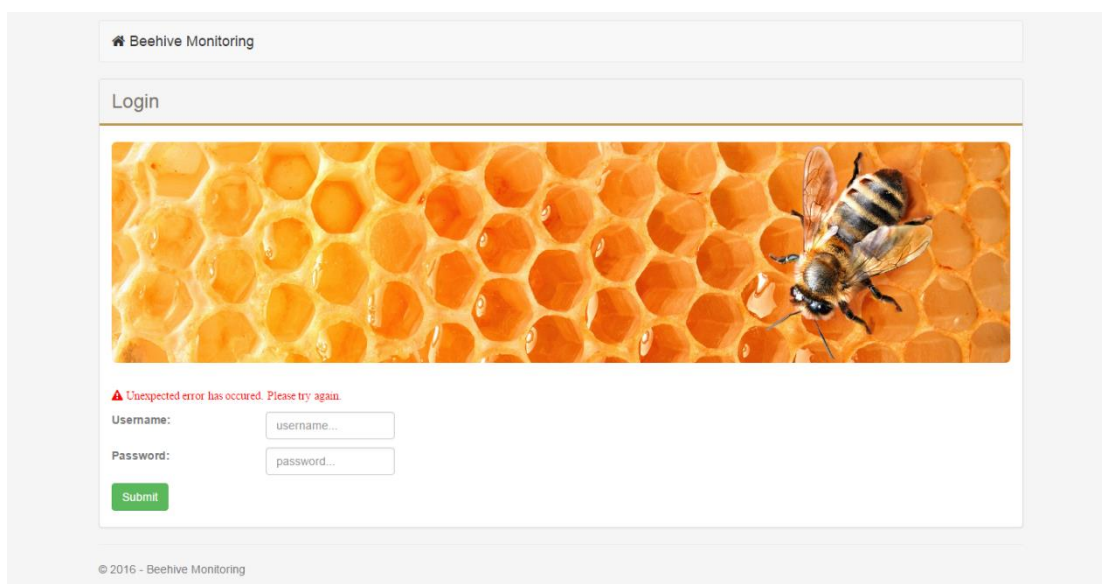
**Εικόνα 39: Η οθόνη επεξεργασίας στοιχείων του χρήστη**

## **5.8 Αποσύνδεση από την εφαρμογή**

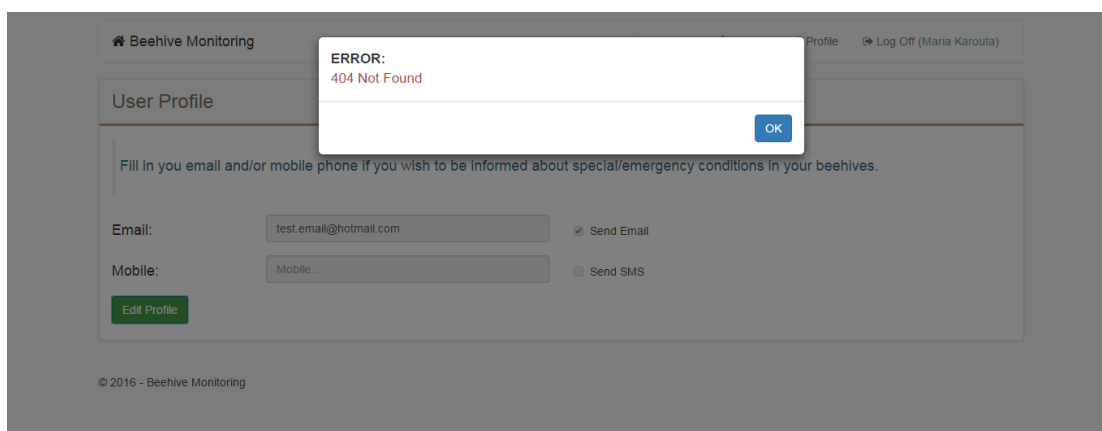
Ο χρήστης μπορεί να αποσυνδεθεί από την εφαρμογή κάνοντας κλικ στο link «Log Off (username)» στο navigation bar. Αυτομάτως, πραγματοποιείται μια ajax post κλήση προς το server η οποία μεταφέρει το αίτημα του χρήστη για αποσύνδεση. Όπως αναφέρθηκε σε προηγούμενη ενότητα, η πιστοποίηση του χρήστη στην εφαρμογή πραγματοποιείται χρησιμοποιώντας τη «Forms Authentication» μέθοδο που υποστηρίζει το .NET περιβάλλον. Σύμφωνα με αυτήν τη μέθοδο, ο χρήστης πιστοποιείται από ένα ticket για οποιαδήποτε ενέργεια πραγματοποιεί εντός του website. Αυτό το ticket αποστέλλεται στο server σε οποιοδήποτε request του client. Στη συγκεκριμένη εφαρμογή, κατά τη login διαδικασία δημιουργείται ένα cookie το οποίο αποθηκεύεται στον browser του client και χρησιμοποιείται ως authentication ticket. Έτσι, σε κάθε επόμενο request σε αυτήν την τοποθεσία, αποστέλλεται στο server μαζί με το request και το ticket-cookie για να πιστοποιηθεί ο χρήστης. [7] Κατά τη Log Off διαδικασία, το ticket-cookie διαγράφεται από τον browser. Με αυτόν τον τρόπο ο χρήστης παύει να είναι πιστοποιημένος στον ιστότοπο της εφαρμογής και τελικά ανακατευθύνεται στη Login σελίδα για να συνδεθεί ξανά.

## 5.9 Χειρισμός σφαλμάτων στην εφαρμογή

Σε όλες τις διαδικτυακές εφαρμογές υπάρχει πιθανότητα να παρουσιαστούν σφάλματα κατά την πλοήγηση των χρηστών σε αυτές. Τα πιο συνηθισμένα αφορούν κάποιο σφάλμα στο server (500 – Internal Server Error), επιχείρηση πρόσβασης σε σελίδα από μη πιστοποιημένο χρήστη (401 – Unauthorized), πρόσβαση σε σελίδα που δε βρέθηκε (404 – Not Found) και άλλα. Κρίνεται λοιπόν απαραίτητη η πρόβλεψη του χειρισμού τους κατά το σχεδιασμό της εφαρμογής, με την εμφάνιση κατάλληλου μηνύματος προς το χρήστη. Παρακάτω παρατίθενται δύο ενδεικτικά στιγμιότυπα από την εφαρμογή κατά την ανάκυψη σφάλματος (α) στην προσπάθεια σύνδεσης του χρήστη στην εφαρμογή (Εικόνα 40) και (β) κατά την κλήση σε μία μέθοδο που δεν υπάρχει στον controller (Εικόνα 41).



Εικόνα 40: Λανθασμένη εισαγωγή στοιχείων από το χρήστη



Εικόνα 41: Οθόνη εμφάνισης σφάλματος

# ΚΕΦΑΛΑΙΟ 6

## Η ΣΥΛΛΟΓΗ ΜΕΤΡΗΣΕΩΝ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΕΠΙΠΕΔΟ

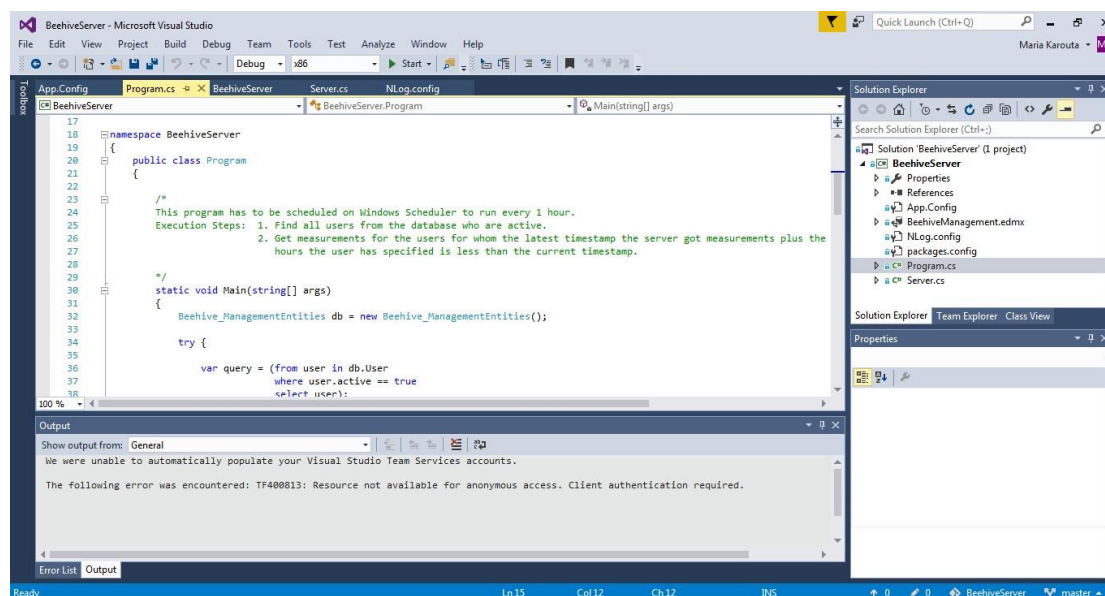
---

Στο κεφάλαιο αυτό πραγματοποιείται αναφορά στις γλώσσες προγραμματισμού και στα πλαίσια (frameworks) που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής που αφορά την αυτοματοποιημένη συλλογή των μετρήσεων από τις κυψέλες των μελισσοκόμων. Ακολουθεί μία συνοπτική περιγραφή της δομής του κώδικα και των συστατικών του στοιχείων. Τέλος, περιγράφεται ο μηχανισμός καταγραφής συμβάντων που χρησιμοποιείται κατά την εκτέλεση του προγράμματος αυτού.

### ***6.1 Γλώσσες προγραμματισμού και πλαίσια***

Για την ανάπτυξη του προγράμματος συλλογής μετρήσεων από τις κυψέλες χρησιμοποιήθηκε το προγραμματιστικό εργαλείο Microsoft Visual Studio 2015 Community Edition. Το Microsoft Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft. Χρησιμοποιείται για την ανάπτυξη προγραμμάτων ηλεκτρονικών υπολογιστών για τα Microsoft Windows, καθώς και ιστοσελίδων, διαδικτυακών εφαρμογών και υπηρεσιών Web. Το Visual Studio χρησιμοποιεί πλατφόρμες ανάπτυξης λογισμικού της Microsoft, όπως τα Windows API, τα Windows Forms, το Windows Presentation Foundation, το Windows Store και το Microsoft Silverlight. Μπορεί να παράγει τόσο εγγενή όσο και διαχειριζόμενο κώδικα. Το Visual Studio υποστηρίζει ποικίλες γλώσσες προγραμματισμού. Ανάμεσα στις ενσωματωμένες γλώσσες τις οποίες υποστηρίζει ανήκουν οι C, C++ και C++/CLI, VB.NET, C# και F#. Υποστήριξη για άλλες γλώσσες, όπως Python, Ruby, Node.js και M μεταξύ άλλων, είναι διαθέσιμη μέσω ξεχωριστής εγκατάστασης των αντίστοιχων language services. Υποστηρίζει, επίσης, XML / XSLT, HTML / XHTML, JavaScript και

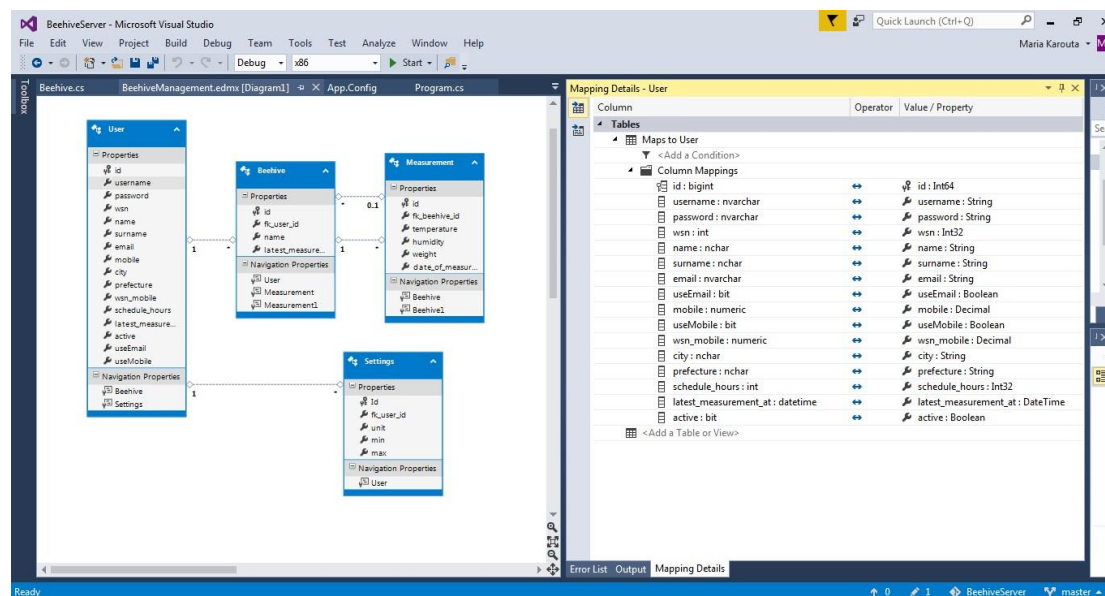
CSS. Java και J# υποστηρίζονταν στο παρελθόν. [9] Στην εν λόγω εφαρμογή χρησιμοποιήθηκε η γλώσσα C#.



**Εικόνα 42: Το περιβάλλον προγραμματισμού Microsoft Visual Studio 2015 Community Edition**

Κατά την ανάπτυξη του κώδικα του προγράμματος αυτού χρησιμοποιήθηκε το Entity Framework. Το Entity Framework είναι ένα Object/Relational Mapping (O/RM) πλαίσιο. Παρέχει στους προγραμματιστές έναν αυτοματοποιημένο μηχανισμό για την ανάκτηση και την αποθήκευση δεδομένων στη βάση δεδομένων καθώς και την εργασία με τα αποτελέσματα των ερωτημάτων. Το ORM είναι ένα εργαλείο που χρησιμοποιείται για αποθήκευση δεδομένων από domain αντικείμενα σε σχεσιακές βάσεις δεδομένων, όπως ο SQL Server της Microsoft, με έναν αυτοματοποιημένο τρόπο, χωρίς πολύ προγραμματισμό. Το ORM έχει τρία βασικά μέρη: τα domain class objects, τα relational database objects (τα αντικείμενα της σχεσιακής βάσης π.χ. tables, views, stored procedures) και mapping για τον τρόπο με τον οποίο τα domain objects αντιστοιχίζονται στα relational database objects. Το ORM συμβάλλει στο διαχωρισμό της σχεδίασης της βάσης δεδομένων από το σχεδιασμό και την ανάπτυξη του προγραμματιστικού μέρους. Με αυτόν τον τρόπο η εφαρμογή καθίσταται διατηρήσιμη και επεκτάσιμη. Επιπλέον, το ORM αυτοματοποιεί και τις τυπικές λειτουργίες CRUD (Create, Read, Update, Delete), συνεπώς ο προγραμματιστής δε χρειάζεται να τις υλοποιήσει χειροκίνητα. [10]

Στην Εικόνα 43 παρουσιάζεται το Entity Data Model της εφαρμογής. Αριστερά φαίνεται το διάγραμμα των domain κλάσεων και των μεταξύ τους σχέσεων, ενώ δεξιά το mapping πεδίο προς πεδίο από τον πίνακα User της βάσης δεδομένων προς την domain κλάση User.



Εικόνα 43: Entity Framework

## 6.2 Περιγραφή κλάσεων και μεθόδων

Στο υποκεφάλαιο αυτό αναλύονται οι κλάσεις και οι μέθοδοί τους που συγκροτούν το πρόγραμμα αυτοματοποιημένης συλλογής μετρήσεων από τις κυψέλες των μελισσοκόμων.

### 6.2.1 Program

Πρόκειται για την «κεντρική» κλάση του προγράμματος, μέσα από την οποία εκκινεί το πρόγραμμα.

#### Πεδία:

- public static Logger Logger

Το πεδίο αυτό τύπου Logger της βιβλιοθήκης NLog χρησιμοποιείται για την καταγραφή συμβάντων στην κονσόλα αλλά και σε αρχεία κατά την εκτέλεση του προγράμματος.

### Μέθοδοι:

➤ `static void Main (string[] args)`

Κάθε εφαρμογή σε C# περιέχει μία και μοναδική Main μέθοδο η οποία προσδιορίζει από πού θα ξεκινήσει η εκτέλεση του προγράμματος. Τα βήματα που εκτελεί η συνάρτηση Main στο συγκεκριμένο πρόγραμμα είναι:

- i. Ανάκτηση από τη βάση δεδομένων της εφαρμογής όλων των ενεργών χρηστών.
- ii. Λήψη μετρήσεων για όλους τους ενεργούς χρήστες για τους οποίους έχουν παρέλθει οι προγραμματισμένες ώρες από τη λήψη του τελευταίου στιγμιότυπου μετρήσεων.

### **6.2.2 Server**

Η κλάση Server συγκεντρώνει όλες τις μεθόδους που απαιτούνται προκειμένου να πραγματοποιηθεί η λήψη και επεξεργασία μετρήσεων από τις κυψέλες ενός μελισσοκόμου.

### Μέθοδοι:

➤ `public void GetUserMeasurements(decimal mobile)`

Η μέθοδος αυτή συγκεντρώνει όλα τα βήματα που πρέπει να γίνουν κατά τη λήψη και την επεξεργασία των μετρήσεων από τις κυψέλες ενός χρήστη:

- i. Κλήση στον αριθμό τηλεφώνου που αντιστοιχεί στο GSM modem με το οποίο συνδέεται ο Gateway του ασύρματου δικτύου αισθητήρων του μελισσοκόμου (παράμετρος mobile).
- ii. Δημιουργία socket σύνδεσης με τον Gateway.
- iii. Λήψη των μετρήσεων από τον Gateway μέσω της socket σύνδεσης και αποθήκευσή τους στη βάση δεδομένων.
- iv. Ενημέρωση του timestamp τελευταίας λήψης μετρήσεων στη βάση δεδομένων, ώστε στην επόμενη εκτέλεση του προγράμματος να μπορεί να ελεγχθεί εάν έχουν περάσει οι προγραμματισμένες ώρες και πρέπει να ληφθούν ξανά μετρήσεις για το χρήστη αυτό.



- `private void AddNewMeasurement(string wsn, DateTime time, string measurement)`  
 Η μέθοδος αυτή χρησιμοποιείται για την προσθήκη ενός στιγμιότυπου μετρήσεων μίας κυψέλης στη βάση δεδομένων. Αρχικά ελέγχεται αν το στιγμιότυπο αυτό υπάρχει ήδη, με βάση το timestamp που στέλνει ο Gateway, το οποίο αντιστοιχεί στη χρονική στιγμή κατά την οποία ο Gateway έλαβε μετρήσεις από την κυψέλη. Εάν δεν υπάρχει, οι μετρήσεις αποθηκεύονται στη βάση δεδομένων ως οι πιο πρόσφατες και κατόπιν ελέγχεται εάν οι τιμές αυτές βρίσκονται εκτός των ορίων που έχει θέσει ο χρήστης στα Settings του.
- `private void CheckNotificationSettings(Measurement measToAdd, long beehiveId, int wsn)`  
 Στη μέθοδο αυτή ελέγχεται εάν η μετρούμενη τιμή ενός μεγέθους (θερμοκρασίας, υγρασίας, βάρους) σε μία κυψέλη έχει ξεπεράσει ή πέσει κάτω από μια τιμή που έχει ορίσει ο χρήστης στις ρυθμίσεις του. Εάν ναι, ο χρήστης θα ειδοποιηθεί για το γεγονός αυτό.
- `private void ContactUser(long userId, long beehiveId, string value, string unit)`  
 Η μέθοδος αυτή χρησιμοποιείται για να ειδοποιήσει ένα χρήστη μέσω email ή/και SMS ότι η τιμή ενός μετρούμενου μεγέθους σε μία κυψέλη του έχει βγει εκτός των ορίων που έχει θέσει ο ίδιος. Ο χρήστης θα ειδοποιηθεί μόνο σε περίπτωση που έχει δηλώσει ότι επιθυμεί να λαμβάνει email ή/και SMS από την εφαρμογή.
- `private void SendSMS(decimal? mobile, string message)`  
 Η μέθοδος αυτή χρησιμοποιείται για την αποστολή SMS προς τον αριθμό κινητού τηλεφώνου ενός χρήστη.
- `private void SendEmail(string email, string message)`  
 Η μέθοδος αυτή χρησιμοποιείται για την αποστολή email προς έναν χρήστη της εφαρμογής.

### ***6.3 Μηχανισμός καταγραφής συμβάντων***

Σε εφαρμογές όπου συνδέονται συσκευές μέσω δικτύου είναι συχνό φαινόμενο η εμφάνιση σφαλμάτων. Στην εφαρμογή που εξετάζουμε τα σφάλματα που μπορεί να ανακύψουν και να εμποδίσουν την επικοινωνία Gateway-server ενδεικτικά είναι:

- Το GSM modem που συνδέεται στον Gateway δεν έχει σήμα, συνεπώς η αναπάντητη κλήση από το server προς το GSM θα αποτύχει και ο Gateway δε θα αφυπνιστεί προκειμένου να στείλει μετρήσεις. Ακόμη, εάν το GSM δεν έχει σύνδεση στο Διαδίκτυο, τότε δε θα μπορέσει ο Gateway να συνδεθεί μέσω socket με το server.
- Κατ' αντιστοιχία με πριν, το GSM modem που συνδέεται πάνω στο server μπορεί να μην έχει σήμα και να μην μπορεί να πραγματοποιήσει κλήση.

Λόγω των παραπάνω προβλημάτων καθίσταται αναγκαία η δυνατότητα παρακολούθησης της λειτουργίας του συστήματος λήψης και αποθήκευσης των μετρήσεων των αισθητήρων.

Ο μηχανισμός του συστήματος όπως περιγράφηκε σε προηγούμενο κεφάλαιο εκτελείται με αυτόματο τρόπο κάθε μία ώρα. Συνεπώς, είναι ανέφικτο να ελέγχει ο διαχειριστής του συστήματος μετά την εκτέλεσή του προγράμματος κάθε μία ώρα τη βάση δεδομένων ώστε να καταγράψει για ιστορικούς λόγους για ποιους χρήστες ελήφθησαν μετρήσεις και εάν η διαδικασία λήψης μετρήσεων ήταν ανεπιτυχής για κάποιο χρήστη. Επιπλέον, με αυτόν τον τρόπο σε περίπτωση κάποιας αποτυχίας δεν γνωρίζει ακριβώς τι πρόβλημα ανέκυψε. Κρίνεται λοιπόν αναγκαίος ένας μηχανισμός καταγραφής όλων των συμβάντων που λαμβάνουν χώρα κατά την εκτέλεση του προγράμματος συλλογής μετρήσεων ο οποίος θα ελέγχεται από το διαχειριστή του συστήματος ανά τακτά χρονικά διαστήματα.

Για όλους τους παραπάνω λόγους χρησιμοποιήθηκε το NLog για την καταγραφή των συμβάντων. Το NLog είναι μια δωρεάν πλατφόρμα καταγραφής για .NET, Xamarin, Silverlight και Windows Phone με πλούσιες δυνατότητες δρομολόγησης και διαχείρισης καταγραφής. Το NLog καθιστά εύκολη την παραγωγή και τη διαχείριση υψηλής ποιότητας αρχείων καταγραφής για κάθε εφαρμογή, ανεξάρτητα από το μέγεθος ή την πολυπλοκότητά της. Το NLog μπορεί να επεξεργαστεί τα διαγνωστικά μηνύματα που εκπέμπονται από οποιαδήποτε γλώσσα .NET (C#, VB.NET κλπ.), εμπλουτίζοντάς τα με συναφείς πληροφορίες (την ημερομηνία και ώρα, τη σοβαρότητα σφάλματος, το νήμα, τη διαδικασία, το περιβάλλον), μορφοποιώντας τα ανάλογα με τις προτιμήσεις του χρήστη και να τα στείλει σε έναν ή περισσότερους στόχους. Το NLog μπορεί να καταγράψει κάθε ένα από τα μηνύματα αυτά σε έναν ή και περισσότερους στόχους.

Μερικοί από τους στόχους που μπορούν να χρησιμοποιηθούν ενδεικτικά είναι:

- Αρχεία: καταγραφή των log μηνυμάτων σε ένα ή περισσότερα αρχεία τα οποία ονοματίζονται και τοποθετούνται σε φακέλους του συστήματος αυτόματα.
- Βάση δεδομένων: καταγραφή των μηνυμάτων σε βάσεις δεδομένων που υποστηρίζονται από το .NET.
- Console: καταγραφή των μηνυμάτων στην κονσόλα κατά την εκτέλεση του προγράμματος.
- Email: αποστολή email σε περίπτωση σφάλματος.
- Event Log: καταγραφή μηνυμάτων στα event logs τοπικού ή απομακρυσμένου μηχανήματος. [11]

Στο NLog.config αρχείο της λύσης περιγράφονται οι στόχοι και οι κανόνες καταγραφής. Έχουν χρησιμοποιηθεί τρεις στόχοι στην παρούσα εφαρμογή: δύο για αρχεία και η κονσόλα.

1. Ο πρώτος στόχος καταγράφει τα μηνύματα όλων των τύπων συμβάντων σε ένα αρχείο με όνομα την ημερομηνία εκτέλεσης του προγράμματος στη μορφή “yyyy-MM-dd.log”. Τα αρχεία αυτά αποθηκεύονται στο δίσκο στο path “./logs/{year}/{month}”, ανάλογα με το έτος και το μήνα.
2. Ο δεύτερος στόχος καταγράφει τα μηνύματα τύπου Error σε ένα αρχείο με όνομα της μορφής “yyyy-MM-dd\_error.log”. Τα αρχεία αυτά αποθηκεύονται στο δίσκο στο path “./logs/{year}/{month}”, ανάλογα με το έτος και το μήνα.
3. Ο τρίτος στόχος αφορά τη real-time εκτέλεση του προγράμματος και καταγράφει στην κονσόλα όλων των τύπων τα συμβάντα.

Στην επόμενη σελίδα παρουσιάζεται ένα δείγμα αρχείου καταγραφής της εκτέλεσης του προγράμματος.

2016-08-28 11:51:12.7671 INFO Process start at 28/8/2016 11:51:12 pm  
2016-08-28 11:51:12.7941 INFO Total users to process: 1  
2016-08-28 11:51:13.5381 INFO Modem is connected.  
2016-08-28 11:51:13.5431 INFO Calling 69XXXXXXXXX...  
2016-08-28 11:51:24.0487 INFO Successful phone call.  
2016-08-28 11:51:34.0543 INFO Waiting for socket connection...  
2016-08-28 11:51:40.8237 INFO New socket connection.  
2016-08-28 11:51:40.8337 INFO Starting to process user...  
2016-08-28 11:51:40.8337 INFO 7001  
2016-08-28 11:51:40.8507 INFO timestamp 16/08/28,11:51:40  
2016-08-28 11:51:44.7239 INFO N100 H=5% T=38C W=60,1KG  
2016-08-28 11:51:45.1409 INFO Checked notification settings for  
beehive N100  
2016-08-28 11:51:57.1426 INFO N101 H=53% T=35C W=43,9KG  
2016-08-28 11:51:57.1706 INFO Checked notification settings for  
beehive N101  
2016-08-28 11:52:07.5792 INFO N102 H=15% T=26C W=42,9KG  
2016-08-28 11:52:07.6472 INFO Checked notification settings for  
beehive N102  
2016-08-28 11:52:15.6947 INFO N103 H=39% T=19C W=46KG  
2016-08-28 11:52:15.7397 INFO Checked notification settings for  
beehive N103  
2016-08-28 11:52:19.5979 INFO N104 H=12% T=32C W=13,6KG  
2016-08-28 11:52:19.6779 INFO Contact user id 2 because of emergency  
weight conditions in beehive N104.  
2016-08-28 11:52:21.3840 INFO Email successfully sent.  
2016-08-28 11:52:21.8940 INFO Modem is connected. Ready to send SMS  
2016-08-28 11:52:28.8984 INFO SMS successfully sent.  
2016-08-28 11:52:28.8984 INFO Checked notification settings for  
beehive N104  
2016-08-28 11:52:28.9104 INFO N105 H=51% T=24C W=12,5KG  
2016-08-28 11:52:28.9274 INFO Contact user id 2 because of emergency  
weight conditions in beehive N105.  
2016-08-28 11:52:30.4425 INFO Email successfully sent.  
2016-08-28 11:52:30.9535 INFO Modem is connected. Ready to send SMS  
2016-08-28 11:52:37.9589 INFO SMS successfully sent.  
2016-08-28 11:52:37.9589 INFO Checked notification settings for  
beehive N105  
2016-08-28 11:52:38.0319 INFO Successfully processed user with wsn  
7001.  
2016-08-28 11:52:38.0369 INFO End of socket connection.  
2016-08-28 11:52:38.0669 INFO End of process at 28/8/2016 11:52:38 pm

Κάθε συμβάν καταγράφεται με την εξής μορφή: timestamp – type of the event – description of the event. Στη συγκεκριμένη εκτέλεση του προγράμματος βλέπουμε τη λήψη μετρήσεων για ένα χρήστη. Στη συγκεκριμένη περίπτωση βλέπουμε μάλιστα ότι οι τιμές του βάρους για τις κυψέλες N104 και N105 βγήκαν εκτός των ορίων που είχε θέσει ο χρήστης, συνεπώς ειδοποιήθηκε μέσα από το πρόγραμμα με SMS και email για αυτά τα δύο συμβάντα.



# ΚΕΦΑΛΑΙΟ 7

## ΣΧΕΔΙΑΣΗ ΤΗΣ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ

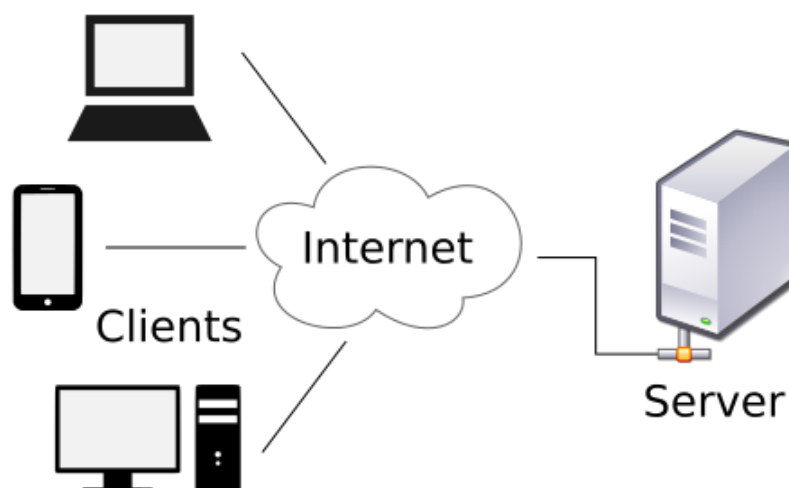
---

Στο κεφάλαιο αυτό πραγματοποιείται μια αναφορά στην αρχιτεκτονική, στις γλώσσες προγραμματισμού και στα πλαίσια (frameworks) που χρησιμοποιήθηκαν για την υλοποίηση της διαδικτυακής εφαρμογής παρουσίασης των μετρήσεων στο χρήστη – μελισσοκόμο. Ακολουθεί μία περιγραφή της δομής του κώδικα και των συστατικών του στοιχείων.

### 7.1 Αρχιτεκτονική εφαρμογής

#### 7.1.1 Μοντέλο Client-Server

Το μοντέλο αρχιτεκτονικής που χρησιμοποιήθηκε για την υλοποίηση της διαδικτυακής εφαρμογής είναι αυτό του πελάτη – εξυπηρετητή (client – server).

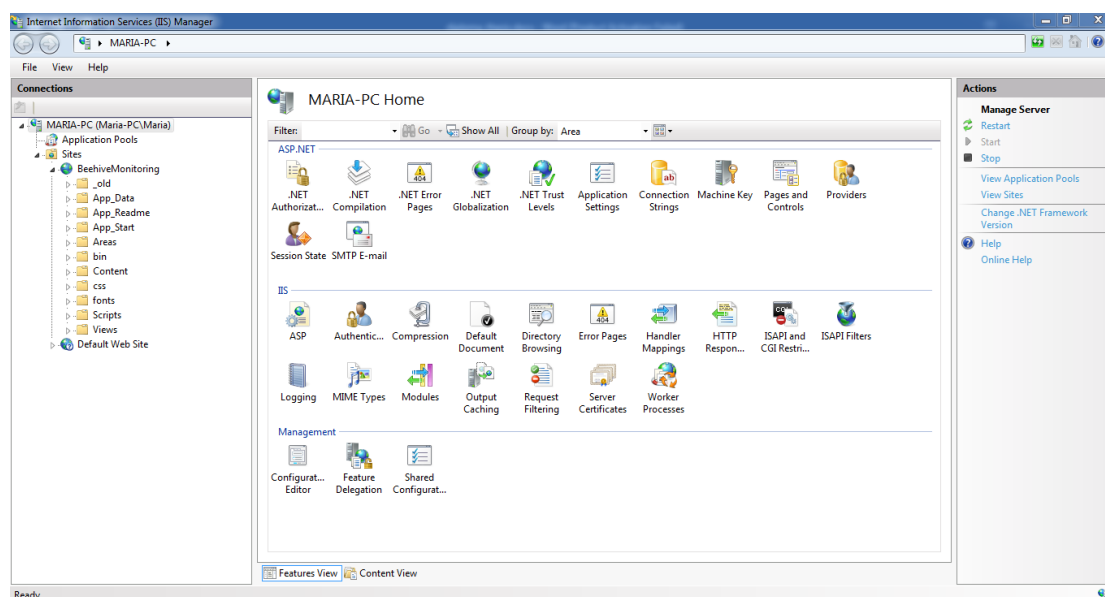


Εικόνα 44: Μοντέλο Client – Server

Σύμφωνα με το μοντέλο αυτό, υπάρχουν δύο διαφορετικές πλευρές που αλληλεπιδρούν μεταξύ τους. Ο εξυπηρετητής (ή αλλιώς back-end) οργανώνει, διαχειρίζεται το αρχείο δεδομένων, δέχεται ερωτήματα και απαντά στο πρόγραμμα του πελάτη. Από την άλλη πλευρά, το πρόγραμμα του πελάτη (ή αλλιώς front-end) θέτει ερωτήματα στον εξυπηρετητή και μπορεί να αποκωδικοποιεί τις απαντήσεις του εξυπηρετητή. [12]

### 7.1.2 Web Server

Ο Web Server που χρησιμοποιήθηκε για να φιλοξενήσει την εφαρμογή είναι ο Internet Information Services (IIS). Πρόκειται για έναν εξυπηρετητή που δημιουργήθηκε από τη Microsoft για να χρησιμοποιείται στα μηχανήματα που διαθέτουν λειτουργικό σύστημα Windows. Ο IIS υποστηρίζει τα πρωτόκολλα HTTP (Hypertext Transfer Protocol), HTTPS (HTTP Secure), FTP (File Transfer Protocol), FTPS (FTP Secure) , SMTP (Simple Mail Transfer Protocol) και NNTP (Network News Transfer Protocol). [13]



Εικόνα 45: Internet Information Services (IIS) Manager



## **7.2 Γλώσσες προγραμματισμού και πλαίσια**

Η διαδικτυακή εφαρμογή παρουσίασης των μετρήσεων που συλλέγονται από τις κυψέλες των μελισσοκόμων αναπτύχθηκε χρησιμοποιώντας το εργαλείο Microsoft Visual Studio 2015 Community Edition. Όπως αναφέρθηκε στην προηγούμενη υποενότητα, το μοντέλο αρχιτεκτονικής που χρησιμοποιήθηκε είναι το client – server, συνεπώς θα αναλυθούν ξεχωριστά οι γλώσσες και τα πλαίσια που χρησιμοποιήθηκαν σε κάθε ένα από τα δύο μέρη.

### **7.2.1 Ανάπτυξη πλευράς πελάτη (front-end development)**

Η ανάπτυξη πλευράς πελάτη είναι η πρακτική παραγωγής HTML (HyperText Markup Language), CSS (Cascading Style Sheets) και JavaScript για μία ιστοσελίδα ή μια διαδικτυακή εφαρμογή, έτσι ώστε ο χρήστης να μπορεί να τη βλέπει και να αλληλεπιδρά μαζί της άμεσα. [14]

Για την ανάπτυξη της πλευράς πελάτη χρησιμοποιήθηκε κυρίως το Bootstrap πλαίσιο. Προσφέρει ένα σύνολο αντικειμένων υλοποιημένων με HTML και CSS, όπως φόρμες, κουμπιά, μπάρες πλοήγησης, καθώς και πιθανές επεκτάσεις JavaScript. [15] Παράλληλα, χρησιμοποιήθηκαν και κανόνες CSS ώστε να διαμορφωθεί η εμφάνιση ακριβώς στις ανάγκες της εφαρμογής. Για την υλοποίηση των δυναμικών και των επιπρόσθετων γραφικών τμημάτων έγινε χρήση της JavaScript. Χρησιμοποιήθηκε ακόμη η JQuery βιβλιοθήκη η οποία είναι μια βιβλιοθήκη JavaScript σχεδιασμένη για να απλοποιήσει την υλοποίηση σεναρίων (scripting) στη πλευρά του πελάτη (client-side) της HTML και η οποία υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού. [16] Η επικοινωνία μεταξύ client και server έγινε ασύγχρονα με JavaScript μέσω AJAX κλήσεων.

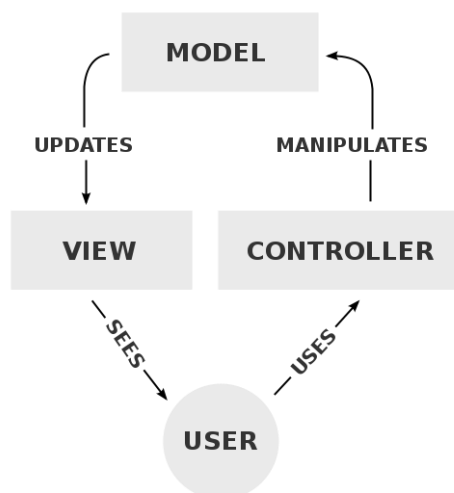
### **7.2.2 Ανάπτυξη πλευράς εξυπηρετητή (back-end development)**

Η πλευρά εξυπηρετητή αναλαμβάνει να εξυπηρετήσει τις front-end υπηρεσίες, εφόσον έχει πρόσβαση στους απαιτούμενους πόρους, όπως για παράδειγμα η βάση δεδομένων.

Η γλώσσα που χρησιμοποιήθηκε για την ανάπτυξη της πλευράς εξυπηρετητή είναι η C#. Επιπλέον, χρησιμοποιήθηκε το Entity Framework το οποίο παρέχει έναν

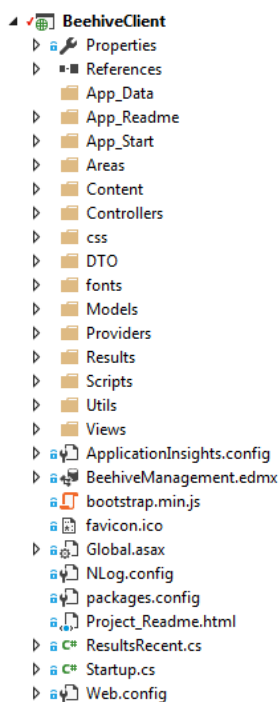
αυτοματοποιημένο μηχανισμό για την ανάκτηση και την αποθήκευση δεδομένων στη βάση δεδομένων, όπως περιγράφηκε αναλυτικότερα σε προηγούμενη ενότητα. Τέλος, έγινε χρήση του αρχιτεκτονικού μοντέλου MVC (Model – View – Controller). Πρόκειται για ένα μοντέλο αρχιτεκτονικής λογισμικού το οποίο χρησιμοποιείται για τη δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη. Στο μοντέλο αυτό η εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη ούτως ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στο χρήστη από τη μορφή που έχει αποθηκευτεί στο σύστημα. Το κύριο μέρος του μοντέλου είναι το αντικείμενο Model το οποίο διαχειρίζεται την ανάκτηση/αποθήκευση των δεδομένων στο σύστημα. Το αντικείμενο View χρησιμοποιείται μόνο για να παρουσιάζεται η πληροφορία στον χρήστη (π.χ. με γραφικό τρόπο). Το τρίτο μέρος είναι ο Controller ο οποίος δέχεται την είσοδο και στέλνει εντολές στο αντικείμενο Model και στο View. Ειδικότερα, οι αλληλεπιδράσεις μεταξύ των τριών συστατικών μερών του MVC μοντέλου είναι οι ακόλουθες:

- Ο Controller μπορεί να στέλνει εντολές στο Model και να ενημερώνει την κατάστασή του. Μπορεί επίσης να στέλνει εντολές ώστε να γίνει η αντίστοιχη αναπαράσταση των δεδομένων του Model μέσω του View.
- Το Model ενημερώνει τις αντίστοιχες αναπαραστάσεις Views και τους Controllers όταν υπάρχει αλλαγή στα δεδομένα. Αυτή η ενημέρωση επιτρέπει στα Views να ενημερώνουν τη γραφική απεικόνιση.
- Το View αναπαριστά με γραφικό τρόπο την πληροφορία που περιέχει το Model δημιουργώντας γραφική παρουσίαση στο χρήστη. [17]



Εικόνα 46: Μοντέλο Model – View – Controller

## 7.3 Αρχιτεκτονική σε επίπεδο κώδικα και κλάσεων



Σε αυτήν την ενότητα περιγράφονται οι βασικές συνιστώσες που συγκροτούν τη διαδικτυακή εφαρμογή παρουσίασης των μετρήσεων από προγραμματιστική άποψη. Όπως αναφέρθηκε στην προηγούμενη ενότητα, κατά την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε το αρχιτεκτονικό μοντέλο Model – View – Controller. Κρίθηκε λοιπόν απαραίτητο να αναφερθούν ποια είναι τα Models, τα Views και οι Controllers της εφαρμογής και πώς επικοινωνούν μεταξύ τους. Πριν από αυτό όμως, θα γίνει αναφορά στις Data Transfer Object κλάσεις και τη χρησιμότητά τους.

**Εικόνα 47: Οργάνωση του project σε φακέλους στο Visual Studio**

### 7.3.1 Data Transfer Objects (DTO)

Ένα Data Transfer Object είναι ένα αντικείμενο που μεταφέρει δεδομένα μεταξύ διεργασιών. Το κίνητρο για τη χρήση των DTO είναι ότι η επικοινωνία μεταξύ των διεργασιών γίνεται συνήθως καταφεύγοντας σε απομακρυσμένες διεπαφές (π.χ. web services), όπου κάθε κλήση είναι μία δαπανηρή λειτουργία. Επειδή το μεγαλύτερο μέρος του κόστους κάθε κλήσης αφορά το χρόνο μετ' επιστροφής μεταξύ του πελάτη και του εξυπηρετητή, ένας τρόπος για τη μείωση του πλήθους των κλήσεων είναι η χρήση ενός αντικειμένου που συγκεντρώνει τα δεδομένα που θα είχαν μεταφερθεί από πολλές κλήσεις, το οποίο όμως μεταφέρεται σε μία μόνο κλήση. Η διαφορά μεταξύ των business objects ή data access objects και των data transfer objects είναι ότι ένα DTO δεν έχει κάποια άλλη λειτουργικότητα πέρα από την αποθήκευση και την ανάκτηση των δικών του δεδομένων. [18]

Οι DTO κλάσεις που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής είναι οι ακόλουθες:

### 1. Beehive

Αυτή η DTO κλάση περιγράφει ένα στιγμιότυπο μετρήσεων σε μία κυψέλη. Χρησιμοποιείται στα πλαίσια της εφαρμογής όταν ανακτώνται από τη βάση δεδομένων οι τελευταίες μετρήσεις για τις κυψέλες. Τα πεδία της κλάσης αυτής είναι:

- Name – Το όνομα της κυψέλης (π.χ. N100).
- Humidity – Υγρασία.
- Temperature – Θερμοκρασία.
- Weight – Βάρος.
- DateOfMeasurement – Ημερομηνία λήψης του στιγμιότυπου μετρήσεων.

### 2. Profile

Η κλάση αυτή περιλαμβάνει τα στοιχεία επικοινωνίας ενός χρήστη καθώς και την προτίμησή του για μέθοδο επικοινωνίας (email, SMS). Τα πεδία της κλάσης αυτής είναι:

- Email – Το email του χρήστη.
- UseEmail – Αληθής/Ψευδής τιμή, ανάλογα εάν ο χρήστης επιθυμεί να λαμβάνει email από την εφαρμογή ή όχι.
- Mobile – Το κινητό τηλέφωνο του χρήστη.
- UseMobile – Αληθής/Ψευδής τιμή, ανάλογα εάν ο χρήστης επιθυμεί να λαμβάνει SMS από την εφαρμογή ή όχι.

### 3. Setting

Τα αντικείμενα της κλάσης αυτής χρησιμοποιούνται για να περιγράψουν μία ρύθμιση του χρήστη. Η ρύθμιση αυτή μπορεί να αφορά την ειδοποίησή του σε περίπτωση ειδικών συνθηκών στις κυψέλες ή το χρονοπρογραμματισμό της λήψης μετρήσεων. Τα πεδία της κλάσης αυτής είναι:

- Name – Το όνομα της ρύθμισης. Πιθανές τιμές: Schedule, Temperature, Humidity, Weight.
- Mode – Το είδος της ρύθμισης. Τα είδη των ρυθμίσεων είναι δύο: Notifications και Schedules.
- MaxValue – Αφορά μόνο τα settings τύπου Notifications. Είναι η μέγιστη τιμή του μεγέθους που όταν ξεπεραστεί ο χρήστης θα πρέπει να ειδοποιηθεί.

- MinValue – Αφορά μόνο τα settings τύπου Notifications. Όταν η τιμή του μεγέθους πέσει κάτω από αυτήν την ελάχιστη τιμή, ο χρήστης θα πρέπει να ειδοποιηθεί.
- Interval – Αφορά μόνο τα settings τύπου Schedules. Περιγράφει κάθε πόσες ώρες θα λαμβάνονται μετρήσεις από τις κυψέλες του μελισσοκόμου.

### 7.3.2 Models

Τα Models της εφαρμογής διαχειρίζονται την ανάκτηση και την αποθήκευση δεδομένων στο σύστημα – βάση δεδομένων χρησιμοποιώντας τα Data Transfer Objects. Τα Models που χρησιμοποιήθηκαν στην εφαρμογή είναι τα ακόλουθα:

#### 1. Beehives

Το μοντέλο αυτό διαχειρίζεται τις τελευταίες μετρήσεις που ελήφθησαν από τις κυψέλες ενός μελισσοκόμου. Η κλάση αυτή περιλαμβάνει τα εξής πεδία/μεθόδους:

- `List<DTO.Beehive> BeehivesList`  
Πρόκειται για μία λίστα από DTO αντικείμενα Beehive. Περιέχει τα στιγμιότυπα των τελευταίων μετρήσεων που ελήφθησαν από όλες τις κυψέλες του μελισσοκόμου.
- `DateTime DateOfMeasurement`  
Η χρονική στιγμή κατά την οποία ελήφθησαν οι τελευταίες μετρήσεις.
- `public static Beehives GetLatestMeasurements(long userId)`  
Η μέθοδος αυτή κατασκευάζει και επιστρέφει το μοντέλο Beehives για ένα χρήστη.

#### 2. Profile

Το μοντέλο αυτό διαχειρίζεται προσωπικές πληροφορίες του χρήστη οι οποίες περιλαμβάνουν τα στοιχεία επικοινωνίας, καθώς και τις προτιμώμενες μεθόδους επικοινωνίας μαζί του (email ή/και SMS). Η κλάση αυτή περιλαμβάνει τα εξής πεδία/μεθόδους:

- `public DTO.Profile ConfProfile`  
Ένα DTO αντικείμενο τύπου Profile που χρησιμοποιείται για την ανάκτηση και την αποθήκευση πληροφοριών σχετικών με το προφίλ του χρήστη στη βάση δεδομένων.
- `public static Profile GetProfile(long userId)`  
Η μέθοδος αυτή κατασκευάζει και επιστρέφει το μοντέλο Profile για ένα χρήστη. Το DTO αντικείμενο ConfProfile του μοντέλου περιέχει όλες τις απαραίτητες πληροφορίες που ανακτήθηκαν από τη βάση δεδομένων.
- `public static Profile SaveProfile(DTO.Profile tmpProfile, long userId)`  
Η μέθοδος αυτή χρησιμοποιείται για να ενημερώσει το προφίλ ενός χρήστη στη βάση δεδομένων. Το DTO αντικείμενο tmpProfile περιέχει τις πληροφορίες που θα αποθηκευτούν στη βάση δεδομένων.

### 3. Settings

Το μοντέλο αυτό χρησιμοποιείται για τη διαχείριση όλων των Settings ενός χρήστη. Τα πεδία/μέθοδοι αυτής της κλάσης είναι:

- `public List<DTO.Setting> ConfSettings`  
Πρόκειται για μία λίστα από DTO αντικείμενα Setting. Περιέχει όλα τα Settings ενός χρήστη (Notification & Schedule).
- `public static Settings GetSettings(long userId)`  
Η μέθοδος αυτή κατασκευάζει και επιστρέφει το μοντέλο Settings για ένα χρήστη. Η λίστα από τα DTO αντικείμενα ConfSettings του μοντέλου περιέχει τις πληροφορίες για όλες τις ρυθμίσεις, όπως ανακτήθηκαν από τη βάση δεδομένων.

### 4. Setting

Η κλάση αυτή διαχειρίζεται μία μόνο ρύθμιση ενός χρήστη. Τα πεδία/μέθοδοι αυτής της κλάσης είναι:

- `public DTO.Setting ConfSetting`  
Πρόκειται για ένα DTO αντικείμενο Setting το οποίο χρησιμοποιείται για την ανάκτηση και την αποθήκευση πληροφοριών στη βάση δεδομένων σχετικών με μία ρύθμιση του χρήστη.

- `public static Setting GetSettingByName(long userId, string settingName, string settingMode)`  
 Η μέθοδος αυτή κατασκευάζει και επιστρέφει το μοντέλο `Setting` μίας συγκεκριμένης ρύθμισης ενός χρήστη. Το DTO αντικείμενο `ConfSetting` του μοντέλου περιέχει τις πληροφορίες της ρύθμισης, όπως ανακτήθηκαν από τη βάση δεδομένων.
- `public static Setting SaveSetting(DTO.Setting setting, long userId)`  
 Η μέθοδος αυτή χρησιμοποιείται για να ενημερώσει μία ρύθμιση ενός χρήστη στη βάση δεδομένων. Το DTO αντικείμενο `setting` περιέχει τις πληροφορίες που θα αποθηκευτούν στη βάση δεδομένων.
- `private static void SaveSchedule(DTO.Setting setting, long userId)`  
 Η μέθοδος αυτή χρησιμοποιείται για την αποθήκευση αλλαγών στη βάση δεδομένων για ρύθμιση τύπου «Schedules».
- `private static void SaveNotification(DTO.Setting setting, long userId)`  
 Η μέθοδος αυτή χρησιμοποιείται για την αποθήκευση αλλαγών στη βάση δεδομένων για ρύθμιση τύπου «Notifications».

### 7.3.3 Views

Στην ενότητα αυτή θα αναφερθούν συνοπτικά τα Views που κατασκευάστηκαν στην εφαρμογή σε συνδυασμό με τα Models τα οποία χρησιμοποιούν (εάν χρησιμοποιούν κάποιο).

1. Για τη Login σελίδα: `Views/Home/Login.cshtml`
2. Για τη Home σελίδα: `Views/Home/Index.cshtml` – Χρησιμοποιεί το μοντέλο `Beehives`.
3. Για την Charts σελίδα: `Views/Charts/Data.cshtml`
4. Για τη Settings σελίδα:
  - i. `Views/Settings/Settings.cshtml` – Χρησιμοποιεί το μοντέλο `Settings`.
  - ii. `Views/Settings/ScheduleSettings.cshtml` – Partial View που ενσωματώνεται στη `Settings.cshtml` για να δείξει το tab «Schedule». Χρησιμοποιεί το μοντέλο `Settings`.

- iii. Views/Settings/UnitSettings.cshtml – Partial View που ενσωματώνεται στη Settings.cshtml για να δείξει το tab «Notification». Χρησιμοποιεί το μοντέλο Settings.
  - iv. Views/Settings/\_dlgSetting.cshtml – Παράθυρο διαλόγου που εμφανίζεται κατά το «Edit» ενός Notification Setting. Χρησιμοποιεί ως μοντέλο το DTO Setting.
5. Για την Profile σελίδα:
- i. Views/Profile/Profile.cshtml – Χρησιμοποιεί το μοντέλο Profile.
  - ii. Views/Profile/\_dlgProfile.cshtml – Παράθυρο διαλόγου που εμφανίζεται κατά το «Edit Profile». Χρησιμοποιεί ως μοντέλο το DTO Profile.

Τέλος, τα κοινά Views για όλες τις σελίδες είναι τα ακόλουθα:

- i. Views/Shared/\_error.cshtml – Παράθυρο διαλόγου που εμφανίζεται όταν προκύψει κάποιο σφάλμα κατά την εκτέλεση της εφαρμογής.
- ii. Views/Shared/\_Layout.cshtml – Αυτό το View ενσωματώνεται σε όλες τις βασικές σελίδες (Views) της εφαρμογής, διότι σε αυτό ορίζονται τα στοιχεία τα οποία είναι κοινά σε όλες, όπως οι JavaScript βιβλιοθήκες και τα CSS αρχεία που χρησιμοποιήθηκαν, η μπάρα πλοήγησης και άλλα, έτσι ώστε να μην επαναλαμβάνονται στον κώδικα κάθε σελίδας.
- iii. Views/Shared/\_WaitDialog.cshtml – Παράθυρο διαλόγου που εμφανίζεται κατά την αναμονή απάντησης από τον εξυπηρετητή σε μία αίτηση.



### 7.3.4 *Controllers*

Οι *Controllers* είναι υπεύθυνοι για να απαντήσουν στα αιτήματα που τίθενται στον ιστότοπο της εφαρμογής. Κάθε αίτημα στον browser αντιστοιχεί σε μία μέθοδο ενός controller. [19] Πιο αναλυτικά, οι *Controllers* της εφαρμογής, μαζί με τις μεθόδους τους, είναι:

1. HomeController: Διαχειρίζεται τα αιτήματα που αφορούν τη σύνδεση και την αποσύνδεση του χρήστη από την εφαρμογή. Ακόμη, αναλαμβάνει την αναπαράσταση των σελίδων Login και Home.

Μέθοδοι:

- `public ActionResult Index()`  
Η μέθοδος αυτή μπορεί να κληθεί μόνο από πιστοποιημένους χρήστες. Κατασκευάζει το Model *Beehives* με τις τελευταίες μετρήσεις που ελήφθησαν από τις κυψέλες του χρήστη και το επιστρέφει μαζί με το View *Index.cshtml*. Σαν αποτέλεσμα αυτής της μεθόδου, εμφανίζεται η Home σελίδα της εφαρμογής.
- `public ActionResult Login()`  
Η μέθοδος αυτή επιστρέφει τη Login σελίδα της εφαρμογής (δηλαδή το View *Login.cshtml*).
- `public void AuthenticateUser(string username, string password)`  
Πρόκειται για μία μέθοδο που καλείται μέσω μίας ajax post κλήσης από τον client μέσω JavaScript. Δέχεται ως ορίσματα το `username` και το `password` του χρήστη και τον πιστοποιεί χρησιμοποιώντας *Forms Authentication*. Εάν τα `credentials` του χρήστη είναι σωστά τον ανακατευθύνει στην Home σελίδα, αλλιώς εμφανίζει κατάλληλο μήνυμα.
- `public void SignOut()`  
Η μέθοδος αυτή μπορεί να κληθεί από πιστοποιημένους χρήστες μόνο μέσω μίας ajax post κλήσης. Αναλαμβάνει την αποσύνδεση του χρήστη από την εφαρμογή.

2. ModemController: Διαχειρίζεται τα αιτήματα που αφορούν την άμεση ενημέρωση του χρήστη σχετικά με την κατάσταση στις κυψέλες του. Ο χρήστης πρέπει να είναι πιστοποιημένος στην εφαρμογή ώστε να έχει πρόσβαση στις μεθόδους αυτού του Controller.

Μέθοδοι:

- `public string UpdateMeasurements()`

Πρόκειται για μία μέθοδο που καλείται μέσω μίας ajax get κλήσης από τον client μέσω JavaScript. Αναλαμβάνει την επικοινωνία με τον Gateway του ασύρματου δικτύου αισθητήρων του χρήστη για τη λήψη μετρήσεων. Επιστρέφει κατάλληλο μήνυμα στο χρήστη εάν ήταν επιτυχής ή όχι η διαδικασία.

- `private static void AddNewMeasurement(string wsn, DateTime time, string measurement)`

Προσθήκη μίας ληφθείσας μέτρησης στη βάση δεδομένων.

3. ChartController: Διαχειρίζεται τα αιτήματα που αφορούν το σχεδιασμό των γραφημάτων στην εφαρμογή. Ο χρήστης πρέπει να είναι πιστοποιημένος στην εφαρμογή ώστε να έχει πρόσβαση στις μεθόδους αυτού του Controller.

Μέθοδοι:

- `public ActionResult Data()`

Η μέθοδος αυτή επιστρέφει την Charts σελίδα της εφαρμογής, δηλαδή το View Data.cshtml.

- `public string GetData(string unit, string dateFrom, string dateTo)`

Πρόκειται για μία μέθοδο που καλείται μέσω μίας ajax get κλήσης από τον client μέσω JavaScript. Χρησιμοποιείται ώστε να ανακτήσει από τη βάση δεδομένων όλες τις μετρήσεις του client για το μέγεθος unit σε ένα συγκεκριμένο χρονικό διάστημα (από dateFrom έως dateTo). Οι μετρήσεις επιστρέφονται ως ένα JSON αντικείμενο το οποίο διαχειρίζεται ύστερα η JavaScript ώστε να σχηματιστεί το γράφημα.

4. SettingsController: Χρησιμοποιείται για τη διαχείριση όλων των αιτημάτων που αφορούν τις ρυθμίσεις του χρήστη στην εφαρμογή. Ο χρήστης πρέπει να είναι πιστοποιημένος στην εφαρμογή ώστε να έχει πρόσβαση στις μεθόδους αυτού του Controller.

Μέθοδοι:

- `public ActionResult Settings()`

Η μέθοδος αυτή κατασκευάζει το Model Settings με όλες τις ρυθμίσεις του χρήστη και το επιστρέφει μαζί με το View Settings.cshtml. Σαν αποτέλεσμα αυτής της μεθόδου, εμφανίζεται η Settings σελίδα της εφαρμογής.

- `public ActionResult GetSetting(string name, string mode)`

Πρόκειται για μία μέθοδο που καλείται μέσω μιας ajax post κλήσης από τον client μέσω JavaScript, όταν κάνει κλικ στο «Edit» ενός Setting. Η μέθοδος αυτή κατασκευάζει το Model Setting της συγκεκριμένης ρύθμισης (με βάση το name και το mode) και επιστρέφει το αντίστοιχο DTO αντικείμενο μαζί με το Partial View \_dlgSetting.cshtml. Ως αποτέλεσμα αυτής της μεθόδου, εμφανίζεται το παράθυρο διαλόγου στο οποίο ο χρήστης μπορεί να επεξεργαστεί μία ρύθμιση.

- `public ActionResult SaveSetting(string name, string mode, string minVal, string maxVal, int? interval)`

Πρόκειται για μία μέθοδο που καλείται μέσω μιας ajax post κλήσης από τον client μέσω JavaScript και αφορά την αποθήκευση των αλλαγών μιας ρύθμισης στη βάση δεδομένων. Το Model Setting αναλαμβάνει τη διεκπεραίωση της αποθήκευσης των αλλαγών.

5. ProfileController: Διαχειρίζεται τα αιτήματα που αφορούν προσωπικές πληροφορίες του χρήστη και ειδικότερα τα στοιχεία επικοινωνίας του. Ο χρήστης πρέπει να είναι πιστοποιημένος στην εφαρμογή ώστε να έχει πρόσβαση στις μεθόδους αυτού του Controller.

Μέθοδοι:

- `public ActionResult Profile()`

Η μέθοδος αυτή κατασκευάζει το Model Profile με τις προσωπικές πληροφορίες του χρήστη και το επιστρέφει μαζί με το View Profile.cshtml. Σαν αποτέλεσμα αυτής της μεθόδου, εμφανίζεται η Profile σελίδα της εφαρμογής.

➤ `public ActionResult GetProfile()`

Πρόκειται για μία μέθοδο που καλείται μέσω μίας ajax post κλήσης από τον client μέσω JavaScript, όταν κάνει κλικ στο «Edit Profile». Η μέθοδος αυτή κατασκευάζει το Model Profile του χρήστη και επιστρέφει το αντίστοιχο DTO αντικείμενο μαζί με το Partial View `_dlgProfile.cshtml`. Ως αποτέλεσμα αυτής της μεθόδου, εμφανίζεται το παράθυρο διαλόγου στο οποίο ο χρήστης μπορεί να επεξεργαστεί τις προσωπικές του πληροφορίες.

➤ `public ActionResult SaveProfile(string email, bool useEmail, decimal? mobile, bool useMobile)`

Πρόκειται για μία μέθοδο που καλείται μέσω μίας ajax post κλήσης από τον client μέσω JavaScript και αφορά την αποθήκευση των αλλαγών στις προσωπικές πληροφορίες του χρήστη στη βάση δεδομένων. Το Model Profile αναλαμβάνει τη διεκπεραίωση της αποθήκευσης των αλλαγών.

# ΚΕΦΑΛΑΙΟ 8

## ΕΠΙΛΟΓΟΣ

---

Τα ασύρματα δίκτυα αισθητήρων, και κατ' επέκταση η απομακρυσμένη παρακολούθησή τους μέσω του Διαδικτύου, βρίσκουν πληθώρα εφαρμογών στην εποχή μας. Ενδεικτικά παρατίθενται ορισμένες εφαρμογές στις οποίες θα μπορούσε να αξιοποιηθεί το λογισμικό που κατασκευάστηκε σε αυτήν την εργασία για την απομακρυσμένη παρακολούθηση των ασύρματων δικτύων αισθητήρων μέσω του Διαδικτύου:

- Παρακολούθηση της ποιότητας ή της ρύπανσης του αέρα.
- Ανίχνευση δασικών πυρκαγιών και παρακολούθησή τους με τη χρήση κόμβων εξοπλισμένων με αισθητήρες για τη μέτρηση της θερμοκρασίας, της υγρασίας και των αερίων που παράγονται από τη φωτιά.
- Ανίχνευση κατολισθήσεων πολύ πριν συμβούν μέσω της παρακολούθησης διαφόρων παραμέτρων του εδάφους.
- Παρακολούθηση της ποιότητας των υδάτων.
- Πρόληψη φυσικών καταστροφών (π.χ. παρακολούθηση της στάθμης του νερού για πρόληψη πλημμύρας).
- Στον κλάδο της γεωργίας για την ακριβή παρακολούθηση της καλλιέργειας κατά την ανάπτυξή της.
- Στα θερμοκήπια για την παρακολούθηση και τη διατήρηση των απαιτούμενων επιπέδων θερμοκρασίας και υγρασίας στο εσωτερικό τους. [20]
- Παρακολούθηση των συνθηκών στο εσωτερικό των κτιρίων όσον αφορά την υγρασία, τον εξαερισμό και τον κλιματισμό.
- Σε βιομηχανικές εφαρμογές που σχετίζονται με τον έλεγχο διαφόρων δυσπρόσιτων παραγωγικών περιοχών, όπως ο έλεγχος στο εσωτερικό μηχανών και σε υπόγειες παραγωγικές διαδικασίες, οι οποίες για προφανείς λόγους είναι αρκετά επικίνδυνες και δύσκολες στον χειρισμό.

- Στα διυλιστήρια για τη μέτρηση και την καταγραφή της θερμοκρασίας και της πίεσης κατά την εκτέλεση διαφόρων διεργασιών.
- Στον τομέα της υγείας για το διαρκή εξ αποστάσεως έλεγχο της κατάστασης της υγείας ενός ασθενούς μέσω της συλλογής, επεξεργασίας, αξιολόγησης, αξιοποίησης και αποθήκευσης της κατάλληλης ιατρικής πληροφορίας. [1]

# ΒΙΒΛΙΟΓΡΑΦΙΑ

---

- [1] Α. Γ. Ζωγόπουλος, Δίκτυο ασύρματων αισθητήρων για απομακρυσμένη παρακολούθηση (Με μια εφαρμογή στο κλάδο της μελισσοκομίας), Αθήνα: Εθνικό Μετσόβιο Πολυτεχνείο, 2016.
- [2] «Διαδικτυακή εφαρμογή» Βικιπαίδεια, [Ηλεκτρονικό]. Available: [https://el.wikipedia.org/wiki/Διαδικτυακή\\_εφαρμογή](https://el.wikipedia.org/wiki/Διαδικτυακή_εφαρμογή). [Πρόσβαση 13/10/2016].
- [3] «Microsoft SQL Server» Wikipedia, the free encyclopedia, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server). [Πρόσβαση 13/10/2016].
- [4] A. Silberschatz, H. Korth και Sudarshan, Συστήματα Βάσεων Δεδομένων, Αθήνα: Μ. Γκιούρδας, 2012.
- [5] «Regular expression» Wikipedia, the free encyclopedia, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression). [Πρόσβαση 13/10/2016].
- [6] Υ. Tzitzikas, «Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών: Καθορισμός των Απαιτήσεων» [Ηλεκτρονικό]. Available: [http://www.csd.uoc.gr/~hy351/2007/downloads/Lectures/IS\\_06a\\_Requirements\\_Determination.pdf](http://www.csd.uoc.gr/~hy351/2007/downloads/Lectures/IS_06a_Requirements_Determination.pdf). [Πρόσβαση 13/10/2016].
- [7] S. Mitchell, «An Overview of Forms Authentication (C#)» Microsoft, 14/01/2008. [Ηλεκτρονικό]. Available: <https://www.asp.net/web-forms/overview/older-versions-security/introduction/an-overview-of-forms-authentication-cs>. [Πρόσβαση 13/10/2016].
- [8] «Highsoft» Wikipedia, the free encyclopedia, [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/Highsoft>. [Πρόσβαση 13/10/2016].
- [9] «Microsoft Visual Studio» Wikipedia, the free encyclopedia, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio). [Πρόσβαση 14/10/2016].
- [10] «What is Entity Framework» Entity Framework Tutorial, [Ηλεκτρονικό]. Available: <http://www.entityframeworktutorial.net/what-is-entityframework.aspx>. [Πρόσβαση 14/10/2016].
- [11] «NLog» NLog, [Ηλεκτρονικό]. Available: <http://nlog-project.org/>. [Πρόσβαση 14/10/2016].

- [12] Κ. Μαχαίρας, «Ψυχαγωγία & ενημέρωση με τη χρήση νέων τεχνολογιών» [Ηλεκτρονικό]. Available: <http://repository.edulll.gr/edulll/retrieve/3473/1037.pdf>. [Πρόσβαση 14/10/2016].
- [13] «Internet Information Services» Wikipedia, the free encyclopedia, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Internet\\_Information\\_Services](https://en.wikipedia.org/wiki/Internet_Information_Services). [Πρόσβαση 14/10/2016].
- [14] «Front-end web development» Wikipedia, the free encyclopedia, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Front-end\\_web\\_development](https://en.wikipedia.org/wiki/Front-end_web_development). [Πρόσβαση 14/10/2016].
- [15] «Bootstrap (front-end framework)» Wikipedia, the free encyclopedia, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)). [Πρόσβαση 14/10/2016].
- [16] «jQuery» Βικιπαίδεια, [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/JQuery>. [Πρόσβαση 14/10/2016].
- [17] «Model-view-controller» Βικιπαίδεια, [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/Model-view-controller>. [Πρόσβαση 14/10/2016].
- [18] «Data transfer object» Wikipedia, the free encyclopedia, [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Data\\_transfer\\_object](https://en.wikipedia.org/wiki/Data_transfer_object). [Πρόσβαση 14/10/2016].
- [19] S. Walther, «ASP.NET MVC Controller Overview (C#)» Microsoft, 16/02/2008. [Ηλεκτρονικό]. Available: <https://www.asp.net/mvc/overview/older-versions-1/controllers-and-routing/aspnet-mvc-controllers-overview-cs>. [Πρόσβαση 14/10/2016].
- [20] «Ασύρματο δίκτυο αισθητήρων» Βικιπαίδεια, [Ηλεκτρονικό]. Available: [https://el.wikipedia.org/wiki/Ασύρματο\\_δίκτυο\\_αισθητήρων](https://el.wikipedia.org/wiki/Ασύρματο_δίκτυο_αισθητήρων). [Πρόσβαση 14/10/2016].



# ΠΑΡΑΡΤΗΜΑ

## Οδηγίες εγκατάστασης της εφαρμογής

---

Στις παρακάτω σελίδες περιγράφεται αναλυτικά τι κινήσεις πρέπει να γίνουν για την εγκατάσταση του συστήματος που κατασκευάστηκε σε αυτήν τη διπλωματική εργασία σε κάποιον υπολογιστή χρησιμοποιώντας το CD με τα αρχεία της εργασίας.

Τα κοινά βήματα που πρέπει να γίνουν για την εγκατάσταση των δύο επιμέρους εφαρμογών του συστήματος είναι:

- A. Εγκατάσταση SQL Server 2014 Express – Αφού εγκαταστήσουμε το πρόγραμμα αυτό στο μηχάνημά μας, ανοίγουμε το SQL Server Management Studio και συνδεόμαστε στο τοπικό SQL Server Instance του μηχανήματός μας με όνομα `.\SQLEXPRESS`. Δημιουργούμε μία νέα βάση δεδομένων με όνομα «Beehive\_Management» και collation `Latin1_General_CI_AS`. Στη συνέχεια εκτελούμε το script `database.sql` που περιέχεται στο CD ώστε να δημιουργηθούν οι πίνακες και οι procedures που χρησιμοποιούνται στις δύο εφαρμογές.
- B. Εγκατάσταση Microsoft Visual Studio 2015 Community Edition – Χρησιμοποιείται για την προβολή και επεξεργασία του κώδικα των επιμέρους project (open solution `BeehiveServer/BeehiveServer.sln` για το πρόγραμμα αυτόματης συλλογής μετρήσεων και `BeehiveClient/BeehiveClient.sln` για τη διαδικτυακή εφαρμογή).
- C. Δημιουργία Dynamic DNS ονόματος – Εγγραφή στο site `no-ip.com` και δημιουργία ενός τουλάχιστον ονόματος. Εγκατάσταση και εκτέλεση του client DUC.
- D. SIM 900 – Για την αναγνώριση της USB To Serial Port συσκευής απαιτείται η εγκατάσταση του Prolific driver. Αφού συνδέσουμε τη συσκευή σε μία USB θύρα του υπολογιστή, μεταβαίνουμε στον Device Manager των Windows και εντοπίζουμε τη συσκευή. Κάνουμε διπλό κλικ στο όνομα της συσκευής και στο tab «Driver» επιλέγουμε «Update Driver...». Κατόπιν προσδιορίζουμε τη

θέση του driver στο σύστημα φακέλων του υπολογιστή, εάν τον έχουμε κατεβάσει, αλλιώς επιλέγουμε να βρουν αυτόματα τα Windows το software online ή στον υπολογιστή.

- E. Static IP – Θα πρέπει να δοθεί στατική IP στο μηχάνημα που εγκαθίστανται οι δύο εφαρμογές, ώστε να μπορούν να δημιουργηθούν ύστερα οι port forwarding rules στο router. Αυτό γίνεται μέσα από τα Properties της σύνδεσης που χρησιμοποιεί ο υπολογιστής στο Internet, και ειδικότερα μέσα από τα IPv4 Properties.

Για την εφαρμογή αυτοματοποιημένης συλλογής μετρήσεων:

- I. Απαιτείται ρύθμιση στο router για port – forwarding από την πόρτα 5000, προς τη στατική IP του υπολογιστή όπου θα τρέξει η εφαρμογή στην πόρτα 5000.
- II. Απαιτείται η δημιουργία ενός Inbound Rule στο firewall του server μηχανήματος που θα επιτρέπει την εισερχόμενη TCP σύνδεση στην πόρτα 5000.
- III. Το executable αρχείο της εφαρμογής που θα πρέπει να εκτελεστεί είναι το BeehiveServer\BeehiveServer\bin\Debug\SocketServer.exe. Επιπλέον θα πρέπει να ρυθμιστούν κατάλληλα τα settings «password», «email» και «smtp\_host» στο αρχείο SocketServer.exe.config στο φάκελο BeehiveServer\BeehiveServer\bin\Debug\ τα οποία αφορούν το λογαριασμό και τον smtp host που θα χρησιμοποιηθούν για την αποστολή email από την εφαρμογή.
- IV. Για τη ρύθμιση της αυτόματης εκτέλεσης του προγράμματος καθημερινά κάθε μία ώρα μέσα από τον Task Scheduler των Windows απαιτούνται τα παρακάτω βήματα:
  - a. Start – Accessories – System Tools – Task Scheduler
  - b. Μέσα στον Task Scheduler: Action – Create Task
  - c. Συμπληρώνουμε το Name και το Description του task.
  - d. Ορίζουμε το trigger που θα προκαλεί την εκτέλεση του task κάθε μέρα κάθε μία ώρα, ξεκινώντας από τις 12:00:00 π.μ. και το καθιστούμε ενεργό (Enabled).

- e. Ορίζουμε ένα καινούριο Action το οποίο θα τρέχει το executable αρχείο που έχει παραχθεί από το πρόγραμμά μας, προσδιορίζοντας κατάλληλα το path του exe μέσα στο δίσκο (όπως ορίζεται στο III).
- f. Στα Conditions άρουμε τον περιορισμό «Start the task only if the computer is on AC power» και στα Settings αφήνουμε τις προεπιλεγμένες ρυθμίσεις.

Για τη διαδικτυακή εφαρμογή:

- I. Απαιτείται ρύθμιση στο router για port – forwarding από την πόρτα 8080, προς τη στατική IP του υπολογιστή όπου θα τρέξει η εφαρμογή στην πόρτα 8080.
- II. Απαιτείται η δημιουργία ενός Inbound Rule στο firewall του server μηχανήματος που θα επιτρέπει την εισερχόμενη TCP σύνδεση στην πόρτα 8080.
- III. Σε όλους τους υπολογιστές που έχουν εγκατεστημένο το λειτουργικό σύστημα Windows, υπάρχει προεγκατεστημένος ο Web Server «Internet Information Services IIS». Ενεργοποιείται μέσα από τον Πίνακα Ελέγχου – Turn Windows Features on or off επιλέγοντας το στοιχείο «Internet Information Services». Επιπλέον, πρέπει να ενεργοποιηθούν και οι επιλογές ASP και ASP.NET που βρίσκονται στο path Internet Information Services-World Wide Web Services-Application Development Features των Windows Features.
- IV. Για την εγκατάσταση της εφαρμογής στον IIS:
  - a. Αντιγράφουμε τα αρχεία που βρίσκονται στο φάκελο WebApp\_Publish στον υπολογιστή σε ένα νέο φάκελο στο path C:\inetpub\wwwroot\.
  - b. Μέσα στο φάκελο που δημιουργήθηκε στο βήμα a, δημιουργούμε φάκελο «App\_Data» και μέσα σε αυτόν φάκελο «logs». Μέσα σε αυτόν θα δημιουργούνται τα αρχεία καταγραφής της εφαρμογής. Ελέγχουμε ώστε ο χρήστης IIS\_IUSRS να έχει δικαίωμα write στο φάκελο αυτό.
  - c. Στον IIS προσθέτουμε ένα καινούριο Web Site (δεξί κλικ στα Sites – Add Web Site). Δίνουμε όνομα στο Web Site, ορίζουμε ως physical path το φάκελο που δημιουργήσαμε στο a και ορίζουμε στο binding τη θύρα 8080 όπου θα ακούει το web site.

- d. Στα Application Pools έχει δημιουργηθεί ένα καινούριο με όνομα ίδιο με του Web Site του βήματος c. Με διπλό κλικ βλέπουμε τα Properties του Application Pool και ελέγχουμε ότι η έκδοση του .NET Framework είναι 4 και το Managed Pipelined Mode «Integrated».