



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

Διπλωματική Εργασία

Σχεδιασμός και υλοποίηση ενσωματωμένου
συστήματος καταγραφής περιβαλλοντικών
συνθηκών και ανάπτυξη υποστηρικτικής
εφαρμογής ιστοχώρου

Embedded system design and implementation for recording
environmental conditions and supporting web application
development



Βάλλας Βασίλειος

Επιβλέπων Καθηγητής: Δρ. Μηνάς Δασυγένης
Εργαστήριο Ψηφιακών Συστημάτων
και Αρχιτεκτονικής Υπολογιστών

Κοζάνη, Μάρτιος 2019

Περιεχόμενα

Κατάλογος πινάκων.....	6
Κατάλογος εικόνων.....	7
Περίληψη	10
Abstract.....	12
Δήλωση Πνευματικών Δικαιωμάτων.....	15
Ευχαριστίες	17
Κεφάλαιο 1ο – Εισαγωγή.....	19
1.1 Περιγραφή σημερινής κατάστασης	19
1.2 Ιδέα και σκοπός υλοποίησης.....	20
1.3 Αντίστοιχα συστήματα και συμπέρασμα υλοποίησης.....	21
1.3.1 Real time standalone data acquisition system for environmental data	21
1.3.2 Weather Station Design Using IoT Platform Based On Arduino Mega.....	22
1.3.3 Weather Station for educational purposes based on Atmega8	23
1.3.4 WiFi ESP8266 and DHT22 sensor.....	24
1.3.5 Arduino Weather Station Project	25
1.3.6 MKRZero Weather Data Logger	27
1.3.7 Measuring water and air temperature with Arduino GSM shield.	29
1.4 Ανάλυση και συμπεράσματα αντίστοιχων συστημάτων	30
1.5 Σύνοψη διπλωματικής εργασίας.....	32
Κεφάλαιο 2ο – Θεωρητικό υπόβαθρο	34
2.1 Προγραμματισμός διαδικτύου	34
2.1.1 HTML.....	35
2.1.2 CSS.....	36
2.1.3 Javascript.....	37
2.1.4 MYSQL.....	39

2.1.5 PHP.....	40
2.2 Δομές (frameworks), βιβλιοθήκες (libraries), μεταγλωτιστές (compilers), εργαλεία.....	41
2.2.1 Laravel και αρχιτεκτονική MVC.....	42
2.2.2 Bootstrap και responsive design.....	44
2.2.3 jQuery.....	44
2.2.4 React.....	45
2.2.5 Axios.....	46
2.2.6 Redux.....	47
2.2.7 Webpack.....	47
2.2.8 Babel.....	48
2.2.8 Pusher API.....	48
2.2.9 Sass.....	49
2.3 Αρχιτεκτονική Arduino.....	49
2.3.1 Είδη Arduino.....	50
2.3.2 Arduino Shields.....	51
2.4 Πλατφόρμες Ανάπτυξης Λογισμικού.....	52
2.4.1 PHPStorm.....	53
2.4.2 Visual Studio Code.....	53
2.4.3 Arduino IDE.....	53
2.5 Εργαλεία Virtualisation – Έτοιμα κουτιά ανάπτυξης εφαρμογών....	54
2.6 Λογισμικά σχεδίασης.....	54
2.6.1 Fritzing.....	54
2.7 Σύνοψη κεφαλαίου.....	55
Κεφάλαιο 3ο – Δομικά μέρη συστήματος.....	57
3.1 Απαιτήσεις συστήματος.....	57
3.2 Ανάλυση βάσης δεδομένων.....	59
3.2.1 Σχεσιακό διάγραμμα βάσης δεδομένων.....	60
3.2.2 Δομή πίνακα “users”.....	61
3.2.3 Δομή πίνακα “roles”.....	64
3.2.4 Δομή πίνακα “stations”.....	65
3.2.5 Δομή πίνακα “categories”.....	67

3.2.6 Δομή πίνακα “category_station”	69
3.2.7 Δομή πίνακα “collections”	69
3.2.8 Δομή πίνακα “logs”	71
3.2.9 Δομή πίνακα “measures”	72
3.2.10 Δομή πίνακα “migrations”	73
3.2.11 Δομή πίνακα “password_resets”	74
3.3 Ασφάλεια συστήματος	74
3.4 Σύνοψη κεφαλαίου	77
Κεφάλαιο 4ο – Υλικό μέρος συστήματος	79
4.1 Arduino Mega 2560 Rev3	80
4.2 Ethernet Shield	82
4.3 ESP8266 WiFi Module και βοηθητικός αντάπτορας	83
4.4 Αισθητήρας θερμοκρασίας, υγρασίας και ατμοσφαιρικής πίεσης ονομασίας GY-BME280	88
4.5 Αισθητήρας καταγραφής συγκέντρωσης σκόνης και μικροσωματιδίων ονομασίας DSM501A	90
4.6 Αισθητήρας αίσθησης βροχής ονομασίας Raindrop Sensor	92
4.7 Αισθητήρας υπεριώδους ακτινοβολίας ονομασίας UV Sensor	93
4.8 Εξάρτημα καταγραφής χρόνου ονομασίας DS3231 RTC	95
4.9 RGB Led	97
4.10 Τρανζίστορ	99
4.11 Τελικό κύκλωμα	99
4.12 Στοιχεία κατανάλωσης ενέργειας	102
4.13 Υπολογισμός κόστους εξαρτημάτων	103
4.14 Σύνοψη κεφαλαίου	105
Κεφάλαιο 5ο – Λογισμικό μέρος συστήματος	107
5.1 Ανάλυση λειτουργιών διαδικτυακής εφαρμογής και διεπαφές χρήστη.	107
5.1.1 Αρχική σελίδα	108
5.1.2 Εγγραφή, σύνδεση, αποσύνδεση, διαδικασία επιβεβαίωσης χρήστη και λειτουργία ανάκτησης κωδικού πρόσβασης	109

5.1.3 Κεντρικό περιβάλλον εξουσιοδοτημένου χρήστη (διαχειριστής - απλός χρήστης).....	114
5.1.4 Σελίδα κατηγοριών (διαχειριστής - απλός χρήστης).....	116
5.1.5 Σελίδα σταθμών (διαχειριστής - απλός χρήστης).....	120
5.1.6 Σελίδα χρηστών (διαχειριστής).....	125
5.1.7 Σελίδα εμφάνισης αποτελεσμάτων συλλογών μετρήσεων (διαχειριστής - απλός χρήστης).....	129
5.1.8 Σελίδα σύγκρισης αποτελεσμάτων συλλογών μετρήσεων (διαχειριστής - απλός χρήστης).....	131
5.1.9 Σελίδα προβολής και επεξεργασίας λογαριασμού χρήστη (διαχειριστής - απλός χρήστης).....	132
5.2 Ανάλυση λογισμικού Arduino	134
5.2.1 Παρουσίαση προγράμματος λειτουργίας	134
5.2.2 Ροή προγράμματος και επεξήγηση λειτουργιών	136
5.2.3 Παρουσίαση βιβλιοθηκών και συναρτήσεων.....	140
5.3 Μετρηκές κώδικα.....	145
5.3.1 Διαδικτυακή εφαρμογή	145
5.3.2 Arduino.....	145
Κεφάλαιο 6ο – Επίλογος	148
6.1 Σύνοψη και συμπεράσματα	148
6.2 Προβλήματα που προέκυψαν και η αντιμετώπιση τους	149
6.3 Μελλοντικές επεκτάσεις	150
Βιβλιογραφία	152
Παράρτημα.....	160
Οδηγίες εγκατάστασης σε localhost.....	160
Οδηγίες εγκατάστασης σε shared hosting provider	161

Κατάλογος πινάκων

ΠΙΝΑΚΑΣ 1: ΠΑΡΟΥΣΙΑΣΗ ΒΑΣΙΚΩΝ ΣΤΟΙΧΕΙΩΝ HTML	36
ΠΙΝΑΚΑΣ 2: ΕΙΔΗ ARDUINO	51
ΠΙΝΑΚΑΣ 3: ΠΙΝΑΚΑΣ "USERS"	62
ΠΙΝΑΚΑΣ 4: ΠΙΝΑΚΑΣ "ROLES"	64
ΠΙΝΑΚΑΣ 5: ΠΙΝΑΚΑΣ "STATIONS"	66
ΠΙΝΑΚΑΣ 6: ΠΙΝΑΚΑΣ "CATEGORIES"	68
ΠΙΝΑΚΑΣ 7: ΠΙΝΑΚΑΣ "CATEGORY_STATION"	69
ΠΙΝΑΚΑΣ 8: ΠΙΝΑΚΑΣ "COLLECTIONS"	70
ΠΙΝΑΚΑΣ 9: ΠΙΝΑΚΑΣ "LOGS"	71
ΠΙΝΑΚΑΣ 10: ΠΙΝΑΚΑΣ "MEASURES"	73
ΠΙΝΑΚΑΣ 11: ΠΙΝΑΚΑΣ "MIGRATIONS"	74
ΠΙΝΑΚΑΣ 12: ΠΙΝΑΚΑΣ "PASSWORD_RESETS"	74
ΠΙΝΑΚΑΣ 13: ΛΙΣΤΑ ΚΑΙ ΚΟΣΤΟΛΟΓΙΟ ΥΛΙΚΩΝ	104
ΠΙΝΑΚΑΣ 14: ΜΕΤΡΙΚΕΣ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ	145
ΠΙΝΑΚΑΣ 15: ΜΕΤΡΙΚΕΣ ARDUINO SKETCH	146

Κατάλογος εικόνων

ΕΙΚΟΝΑ 1: ΕΝΔΕΙΚΤΙΚΗ ΕΙΚΟΝΑ ΠΡΩΤΟΥ ΑΝΤΙΣΤΟΙΧΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	21
ΕΙΚΟΝΑ 2: ΕΝΔΕΙΚΤΙΚΗ ΕΙΚΟΝΑ ΤΡΙΤΟΥ ΑΝΤΙΣΤΟΙΧΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	23
ΕΙΚΟΝΑ 3: ΕΝΔΕΙΚΤΙΚΗ ΕΙΚΟΝΑ ΤΕΤΑΡΤΟΥ ΑΝΤΙΣΤΟΙΧΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	24
ΕΙΚΟΝΑ 4: ΣΧΕΔΙΑΣΗ ΚΥΚΛΩΜΑΤΟΣ ΤΕΤΑΡΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	25
ΕΙΚΟΝΑ 5: ΕΝΔΕΙΚΤΙΚΗ ΕΙΚΟΝΑ ΠΕΜΠΤΟΥ ΑΝΤΙΣΤΟΙΧΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	25
ΕΙΚΟΝΑ 6: ΣΧΕΔΙΑΣΗ ΚΥΚΛΩΜΑΤΟΣ ΠΕΜΠΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	27
ΕΙΚΟΝΑ 7: ΕΝΔΕΙΚΤΙΚΗ ΕΙΚΟΝΑ ΕΚΤΟΥ ΑΝΤΙΣΤΟΙΧΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	27
ΕΙΚΟΝΑ 8: ΣΧΕΔΙΑΣΗ ΚΥΚΛΩΜΑΤΟΣ ΈΚΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	28
ΕΙΚΟΝΑ 9: ΕΝΔΕΙΚΤΙΚΗ ΕΙΚΟΝΑ ΕΒΔΟΜΟΥ ΑΝΤΙΣΤΟΙΧΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	29
ΕΙΚΟΝΑ 10: ΣΧΕΔΙΑΣΗ ΚΥΚΛΩΜΑΤΟΣ ΕΒΔΟΜΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	30
ΕΙΚΟΝΑ 11: ΠΑΡΑΔΕΙΓΜΑ ΣΥΝΤΑΞΗΣ CSS.....	37
ΕΙΚΟΝΑ 12: ΠΑΡΑΔΕΙΓΜΑ ΣΥΝΤΑΞΗΣ JS.....	39
ΕΙΚΟΝΑ 13: ΠΑΡΑΔΕΙΓΜΑ SQL ΕΝΤΟΛΗΣ.....	40
ΕΙΚΟΝΑ 14: ΠΑΡΑΔΕΙΓΜΑ ΣΥΝΤΑΞΗΣ PHP.....	41
ΕΙΚΟΝΑ 15: ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ MVC ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ.....	43
ΕΙΚΟΝΑ 16: ΠΑΡΑΔΕΙΓΜΑ ΕΝΤΟΛΗΣ JQUERY.....	45
ΕΙΚΟΝΑ 17: ΠΑΡΑΔΕΙΓΜΑ REACT COMPONENT.....	46
ΕΙΚΟΝΑ 18: ΠΑΡΑΔΕΙΓΜΑ ΕΝΤΟΛΗΣ AXIOS.....	47
ΕΙΚΟΝΑ 19: ΠΑΡΑΔΕΙΓΜΑ ΚΩΔΙΚΑ SASS.....	49
ΕΙΚΟΝΑ 20: ETHERNET SHIELD.....	52
ΕΙΚΟΝΑ 21: ΣΧΕΣΙΑΚΟ ΔΙΑΓΡΑΜΜΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ SMARTWEATHERAPP.....	61
ΕΙΚΟΝΑ 22: ΠΛΑΚΕΤΑ ARDUINO MEGA 2560 REV3.....	82
ΕΙΚΟΝΑ 23: ETHERNET SHIELD R3.....	83
ΕΙΚΟΝΑ 24: ESP8266 WIFI MODULE.....	85
ΕΙΚΟΝΑ 25: ΑΚΡΟΔΕΚΤΕΣ ESP8266.....	85
ΕΙΚΟΝΑ 26: ESP8266 ΜΕ ADAPTER.....	86
ΕΙΚΟΝΑ 27: ΣΥΝΔΕΣΜΟΛΟΓΙΑ ESP8266 ΜΕ ARDUINO MEGA.....	87
ΕΙΚΟΝΑ 28: GY-BME280.....	88
ΕΙΚΟΝΑ 29: ΣΥΝΔΕΣΜΟΛΟΓΙΑ GY-BME280 ΜΕ ARDUINO MEGA.....	89
ΕΙΚΟΝΑ 30: DSM501A.....	90
ΕΙΚΟΝΑ 31: ΣΥΝΔΕΣΜΟΛΟΓΙΑ DSM501A ΜΕ ARDUINO MEGA.....	91
ΕΙΚΟΝΑ 32: RAINDROP SENSOR.....	92
ΕΙΚΟΝΑ 33: ΣΥΝΔΕΣΜΟΛΟΓΙΑ RAINDROP SENSOR ΜΕ ARDUINO MEGA.....	93
ΕΙΚΟΝΑ 34: UV SENSOR.....	94
ΕΙΚΟΝΑ 35: ΣΥΝΔΕΣΜΟΛΟΓΙΑ UV SENSOR ΜΕ ARDUINO MEGA.....	94
ΕΙΚΟΝΑ 36: DS3231 RTC.....	95
ΕΙΚΟΝΑ 37: ΣΥΝΔΕΣΜΟΛΟΓΙΑ DS3231 RTC ΜΕ ARDUINO MEGA.....	96
ΕΙΚΟΝΑ 38: RGB LED.....	97
ΕΙΚΟΝΑ 39: ΣΥΝΔΕΣΜΟΛΟΓΙΑ RGB LED ΜΕ ARDUINO MEGA.....	98
ΕΙΚΟΝΑ 40: IRF520 MOSFET.....	99
ΕΙΚΟΝΑ 41: ΣΧΕΔΙΟ ΣΥΝΔΕΣΜΟΛΟΓΙΑΣ.....	100
ΕΙΚΟΝΑ 42: ΤΕΛΙΚΟ ΚΥΚΛΩΜΑ.....	101
ΕΙΚΟΝΑ 43: ΜΕΤΡΗΣΗ ΚΑΤΑΝΑΛΩΣΗΣ ΕΝΕΡΓΕΙΑΣ ΣΥΣΚΕΥΗΣ ΣΕ ΕΝΕΡΓΗ ΚΑΤΑΣΤΑΣΗ.....	103
ΕΙΚΟΝΑ 44: ΜΕΤΡΗΣΗ ΚΑΤΑΝΑΛΩΣΗΣ ΕΝΕΡΓΕΙΑΣ ΣΥΣΚΕΥΗΣ ΣΕ ΚΑΤΑΣΤΑΣΗ ΥΠΝΟΥ.....	103
ΕΙΚΟΝΑ 45: ΤΕΛΙΚΗ ΣΥΣΚΕΥΗ.....	104
ΕΙΚΟΝΑ 46: ΑΡΧΙΚΗ ΣΕΛΙΔΑ.....	109
ΕΙΚΟΝΑ 47: ΣΕΛΙΔΑ ΕΓΓΡΑΦΗΣ.....	109
ΕΙΚΟΝΑ 48: ΜΗΝΥΜΑΤΑ ΣΦΑΛΜΑΤΟΣ ΑΠΟΤΥΧΗΜΕΝΗΣ ΠΡΟΣΠΑΘΕΙΑΣ ΕΓΓΡΑΦΗΣ.....	110
ΕΙΚΟΝΑ 49: ΜΗΝΥΜΑ ΕΠΙΤΥΧΗΜΕΝΗΣ ΕΓΓΡΑΦΗΣ.....	111
ΕΙΚΟΝΑ 50: EMAIL ΕΠΙΒΕΒΑΙΩΣΗΣ.....	111

ΕΙΚΟΝΑ 51: ΣΕΛΙΔΑ ΣΥΝΔΕΣΗΣ ΧΡΗΣΤΗ.....	112
ΕΙΚΟΝΑ 52: ΣΕΛΙΔΑ ΑΝΑΚΤΗΣΗΣ ΚΩΔΙΚΟΥ ΜΑΖΙ ΜΕ ΕΠΙΤΥΧΗΜΕΝΗ ΠΡΟΣΠΑΘΕΙΑ.....	112
ΕΙΚΟΝΑ 53: EMAIL ΣΥΝΔΕΣΜΟΥ ΕΠΑΝΑΦΟΡΑΣ ΚΩΔΙΚΟΥ	113
ΕΙΚΟΝΑ 54: ΠΕΡΙΒΑΛΛΟΝ ΔΗΜΙΟΥΡΓΙΑΣ ΚΑΙΝΟΥΡΓΙΟΥ ΚΩΔΙΚΟΥ	113
ΕΙΚΟΝΑ 55: ΣΥΝΔΕΣΜΟΣ ΑΠΟΣΥΝΔΕΣΗΣ ΧΡΗΣΤΗ	114
ΕΙΚΟΝΑ 56: ΠΕΡΙΒΑΛΛΟΝ ΔΙΑΧΕΙΡΙΣΤΗ	114
ΕΙΚΟΝΑ 57: ΠΕΡΙΒΑΛΛΟΝ ΑΠΛΟΥ ΧΡΗΣΤΗ.....	115
ΕΙΚΟΝΑ 58: ΣΕΛΙΔΑ ΚΑΤΗΓΟΡΙΩΝ ΔΙΑΧΕΙΡΙΣΤΗ	117
ΕΙΚΟΝΑ 59: ΣΕΛΙΔΑ ΚΑΤΗΓΟΡΙΩΝ ΑΠΛΟΥ ΧΡΗΣΤΗ.....	117
ΕΙΚΟΝΑ 60: ΔΗΜΙΟΥΡΓΙΑ ΝΕΑΣ ΚΑΤΗΓΟΡΙΑΣ	118
ΕΙΚΟΝΑ 61: ΠΑΝΕΛ ΑΝΑΝΕΩΣΗΣ ΚΑΤΗΓΟΡΙΑΣ.....	119
ΕΙΚΟΝΑ 62: MODAL ΔΙΑΓΡΑΦΗΣ ΚΑΤΗΓΟΡΙΑΣ	119
ΕΙΚΟΝΑ 63: ΣΕΛΙΔΑ ΔΗΜΙΟΥΡΓΙΑΣ ΣΤΑΘΜΟΥ	120
ΕΙΚΟΝΑ 64: ΣΕΛΙΔΑ ΠΡΟΒΟΛΗΣ ΠΡΟΣΩΠΙΚΩΝ ΣΤΑΘΜΩΝ ΧΡΗΣΤΗ	122
ΕΙΚΟΝΑ 65: ΣΕΛΙΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΤΑΘΜΟΥ	122
ΕΙΚΟΝΑ 66: MODAL ΔΙΑΓΡΑΦΗΣ ΣΤΑΘΜΟΥ.....	123
ΕΙΚΟΝΑ 67: ΣΕΛΙΔΑ ΠΡΟΒΟΛΗΣ ΟΛΩΝ ΤΩΝ ΣΤΑΘΜΩΝ	124
ΕΙΚΟΝΑ 68: ΠΑΝΕΛ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΤΑΘΜΟΥ	125
ΕΙΚΟΝΑ 69: ΣΕΛΙΔΑ ΠΡΟΒΟΛΗΣ ΧΡΗΣΤΩΝ	126
ΕΙΚΟΝΑ 70: ΣΕΛΙΔΑ ΠΡΟΒΟΛΗΣ ΛΟΓΑΡΙΑΣΜΟΥ ΧΡΗΣΤΗ	127
ΕΙΚΟΝΑ 71: ΣΕΛΙΔΑ ΔΗΜΙΟΥΡΓΙΑΣ ΧΡΗΣΤΗ	127
ΕΙΚΟΝΑ 72: ΠΑΝΕΛ ΕΠΕΞΕΡΓΑΣΙΑΣ ΧΡΗΣΤΗ	128
ΕΙΚΟΝΑ 73: ΣΕΛΙΔΑ ΕΜΦΑΝΙΣΗΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ.....	130
ΕΙΚΟΝΑ 74: ΣΕΛΙΔΑ ΣΥΓΚΡΙΣΗΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	132
ΕΙΚΟΝΑ 75: ΣΕΛΙΔΑ ΠΡΟΒΟΛΗΣ ΛΟΓΑΡΙΑΣΜΟΥ.....	133
ΕΙΚΟΝΑ 76: ΥΠΟΣΕΛΙΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΔΕΔΟΜΕΝΩΝ ΧΡΗΣΤΗ	133
ΕΙΚΟΝΑ 77: ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ	137

Περίληψη

Λόγω της ραγδαίας εξέλιξης της τεχνολογίας και παράλληλα, την εισχώρησή της σε κομβικούς τομείς για τον άνθρωπο και την καθημερινότητα του, έχει δημιουργηθεί πλέον η ανάγκη για δημιουργία ολοένα και περισσότερων “έξυπνων” συσκευών, οι οποίες θα προσφέρουν στον άνθρωπο λειτουργίες και δυνατότητες, θα τον διευκολύνουν στις καθημερινές του ασχολίες, καθώς και θα του εξυπηρετούν διάφορες ανάγκες. Η βασική ιδέα πάνω στην οποία στηρίζεται αυτή η διπλωματική εργασία, είναι η δημιουργία μίας τέτοιας “έξυπνης” συσκευής, η οποία θα καταγράφει και θα ενημερώνει τον χρήστη για τις περιβαλλοντικές συνθήκες, που επικρατούν στο κοντινό του περιβάλλον.

Πιο συγκεκριμένα, στην παρούσα διπλωματική εργασία σχεδιάστηκε και υλοποιήθηκε, ένα ενσωματωμένο σύστημα το οποίο βασίζεται σε έναν μικροεπεξεργαστή. Αυτός είναι συνδεδεμένος με έναν αριθμό από αισθητήρες, οι οποίοι πραγματοποιούν μετρήσεις στο περιβάλλον, τις επεξεργάζονται και τις αποστέλλουν μέσω διαδικτύου, σε εξυπηρετητή. Η διαδικασία αυτή γίνεται εφικτή με την βοήθεια εξαρτημάτων, συνδεδεμένων στον μικροεπεξεργαστή, που λειτουργούν ως κάρτες δικτύου. Παράλληλα με το ενσωματωμένο σύστημα, σχεδιάστηκε και υλοποιήθηκε μία υποστηρικτική διαδικτυακή εφαρμογή (Web Application), την οποία σερβίρει ο εξυπηρετητής και έχει σαν αποστολή, να λαμβάνει μέσω διαδικτύου, τις μετρήσεις του ενσωματωμένου συστήματος, να τις επεξεργάζεται και αφού τις αποθηκεύσει σε βάση δεδομένων, στην συνέχεια να τις προβάλλει.

Ο χρήστης, αφού δημιουργήσει προσωπικό λογαριασμό στην διαδικτυακή εφαρμογή, θα μπορεί να παρακολουθεί σε πραγματικό χρόνο (Real Time), τις μετρήσεις που θα λαμβάνει η εφαρμογή από το ενσωματωμένο σύστημα, θα έχει την δυνατότητα να τις προβάλλει ατομικά ή μέσω γραφημάτων, να τις επεξεργάζεται, καθώς και να τις συγκρίνει με άλλα συστήματα, άλλων χρηστών.

Οι διαχειριστές θα είναι υπεύθυνοι για την ομαλή λειτουργία της διαδικτυακής εφαρμογής, θα έχουν σαφώς αυξημένες δυνατότητες έναντι των απλών χρηστών, για προβολή και επεξεργασία μετρήσεων και συστημάτων, καθώς και δυνατότητα επεξεργασίας των χρηστών.

Ο δημιουργός της διαδικτυακής εφαρμογής, είναι μοναδικός και του παρέχονται όλες οι λειτουργίες και δυνατότητες της εφαρμογής, παράλληλα με τον έλεγχο και την επεξεργασία των χρηστών και διαχειριστών.

Γενικότερα η ιδέα της διαδικτυακής εφαρμογής παράλληλα με το ενσωματωμένο σύστημα, βασίζεται πάνω σε ένα μοτίβο: Την ελευθερία που δίνεται στον χρήστη, να δημιουργήσει όσα ενσωματωμένα συστήματα επιθυμεί και να παρακολουθεί τις μετρήσεις που αυτά συλλέγουν, μέσω της διαδικτυακής εφαρμογής.

Περιβάλλον Ανάπτυξης Ενσωματωμένου Συστήματος: Το ενσωματωμένο σύστημα υλοποιήθηκε, με βάση τον μικροεπεξεργαστή/μικροελεγκτή Arduino Mega και προγραμματίστηκε, μέσω του λογισμικού ανοικτού κώδικα Arduino IDE.

Περιβάλλον Ανάπτυξης Διαδικτυακής Εφαρμογής: Η διαδικτυακή εφαρμογή υλοποιήθηκε, με χρήση τεχνολογιών λογισμικού και προγραμματισμού διαδικτύου ανοικτού κώδικα (html5, css3, sass, php, Laravel, Bootstrap, JavaScript (ES2016, ES2017, ES2018), jQuery, Axios, Reactjs, React-router, Redux, Redux-thunk, Pusher API, Mysql, Webpack, Babel), αναπτύχθηκε μέσω των παρακάτω κειμενογράφων (PhpStorm, Visual Studio Code) καθώς και με παράλληλη χρήση λογισμικού-υπερόπτη ανοικτού κώδικα (Virtual Box) μαζί με έτοιμο πακέτο προ εγκατεστημένων τεχνολογιών για ανάπτυξη της εφαρμογής (Vagrant Box - Laravel Homestead).

Λέξεις κλειδιά: Ενσωματωμένο Σύστημα, Διαδικτυακή Εφαρμογή, Διεπαφή προγραμματισμού εφαρμογών (API), Διαδίκτυο των Πραγμάτων (IoT), Arduino, “Έξυπνη” Συσκευή, Πραγματικός Χρόνος, Σταθμός.

Abstract

Due to the rapid evolution of technology and its intrusion into key areas for human and his daily routine, the need for more "smart" devices has been created, that will provide people with operations and capabilities, facilitate their day-to-day activities, as well as serve their different needs. The basic idea on which this diploma thesis is based on the creation of such a "smart" device, which will record and inform the user of the environmental conditions prevailing in the surrounding environment.

More specifically, this diploma thesis has been designed and implemented, an embedded system based on a microprocessor. It is connected to a number of sensors, which perform measurements in the environment, process them and send them over the Internet, on a server. This process becomes possible with the help of some modules, connected to the microcontroller, which act as network cards. Along with the embedded system, a Web Application was designed and implemented, which is served from the server and its mission is, to receive the measurements of the embedded system via the Internet, process them and since storing them in a database, finally show them.

After creating a personal account on the web application, the user will be able to monitor in Real Time the measurements the application will receive from the embedded system. Furthermore, he will be able to project, process, and compare them with other systems, which belong to other users.

Administrators will be responsible for the smooth operation of the web application, will have much greater capabilities than simple users, for viewing and editing metrics and systems, and moreover the capability of editing all users.

The creator of the web application is unique and is provided with all the features and capabilities of the application, along with the control and editing of users and administrators.

In general, the concept of the web-based application alongside the embedded system is based on a pattern: The freedom given to the user, to create as many as embedded systems he wants and monitor, the measurements they collect through the web application.

Embedded System Development Environment: The embedded system was implemented, based on the Arduino Mega microprocessor/microcontroller, and was programmed through the open source software, Arduino IDE.

Web Application Development Environment: The web application was implemented using open source web and software technologies (html5, css3, sass, php, Laravel, Bootstrap, JavaScript (ES2016, ES2017, ES2018), jQuery, Axios, Reactjs, React-router, Redux, Redux-thunk, Pusher API, Mysql, Webpack, Babel), developed through the following code editors (PhpStorm, Visual Studio Code) as well as the simultaneous use of virtualization software (Virtual Box), together with a ready-made pre-installed technologies package for application development (Vagrant Box - Laravel Homestead).

Keywords: Embedded System, Web Application, Application Programming Interface (API), Internet of Things (IoT), Arduino, “Smart” Device, Real Time, Station

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο

“Σχεδιασμός και υλοποίηση ενσωματωμένου συστήματος καταγραφής περιβαλλοντολογικών συνθηκών και ανάπτυξη υποστηρικτικής εφαρμογής ιστοχώρου”

καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Μηνά Δασυγένη, αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

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 μόνο.

Ευχαριστίες

Με την ευκαιρία αυτής της διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω την οικογένειά μου, για τη διαχρονική συμπαράστασή τους και την υλική και ηθική στήριξη των επιλογών μου.

Ακόμη θα ήθελα να ευχαριστήσω θερμά όλους τους φίλους και συμφοιτητές μου, που πίστεψαν σε μένα και με ενθάρρυναν καθ' όλη την διάρκεια των σπουδών μου.

Τέλος θα ήθελα να ευχαριστήσω ιδιαίτερα τον κύριο Μηνά Δασυγένη, επιβλέπων της εργασίας και καθηγητή μου, για την εμπιστοσύνη που έδειξε στο πρόσωπο μου με την ανάθεση της παρούσας διπλωματικής εργασίας καθώς και για τη βοήθεια και καθοδήγηση που πρόσφερε όπου και όταν αυτή χρειάστηκε.

Κεφάλαιο 1ο – Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζεται αναλυτικά το αντικείμενο της διπλωματικής εργασίας. Αρχικά γίνεται μία ανάλυση της σημερινής κατάστασης και των συνθηκών που προκύπτουν, με την χρήση της τεχνητής νοημοσύνης, αναφέρονται επίσης η ιδέα παράλληλα με τον σκοπό, που οδήγησαν στην ανάληψη και υλοποίηση της παρούσης διπλωματικής εργασίας. Έπειτα αναφέρονται, αντίστοιχα συστήματα από άλλους ερευνητές, μαζί με το τελικό συμπέρασμα που προσκομίσθηκε, αναλύοντας και συγκρίνοντας ξεχωριστά κάθε σύστημα και τέλος, γίνεται μία συνοπτική παρουσίαση της δομής της εργασίας.

1.1 Περιγραφή σημερινής κατάστασης

Στον, εν εξελίξει, 21ο αιώνα η ανθρωπότητα βιώνει μία τεράστια μετάλλαξη της τεχνολογίας, η οποία τείνει να διεισδύσει σε κάθε κλάδο που συσχετίζεται με τον άνθρωπο. Από την υγεία, την εργασία και την μετακίνηση, μέχρι την ψυχαγωγία, την επικοινωνία και την εκπαίδευση, αναπτύσσονται καθημερινά, με την βοήθεια της τεχνολογίας, ολοένα και περισσότερα εργαλεία και υπηρεσίες, που αλληλεπιδρούν με τον άνθρωπο και την καθημερινότητα του, συμβάλλουν στην βελτίωση της ποιότητας ζωής και γενικότερα διευκολύνουν, καταστάσεις και υποχρεώσεις που βιώνει καθημερινά. Ένας από τους κλάδους που η τεχνολογία έχει διεισδύσει αρκετά, ειδικότερα τα τελευταία χρόνια με συνεχή μελέτη και έρευνα, είναι ο συναρπαστικός και παράλληλα απρόβλεπτος τομέας της τεχνητής νοημοσύνης. Λόγω των αυξανόμενων υποχρεώσεων (επαγγελματικών και μη) του ανθρώπου, υφίσταται πλέον η ανάγκη, για δημιουργία συσκευών και εξαρτημάτων, τα οποία θα είναι δομημένα και προγραμματισμένα με “κάποιου είδους” νοημοσύνη, ικανά να αλληλεπιδρούν με τον άνθρωπο, να επικοινωνούν μαζί του με την βοήθεια του διαδικτύου, εξυπηρετώντας του καθημερινά, διάφορες ανάγκες και υποχρεώσεις, συμβάλλοντας έτσι στην βελτίωση της ποιότητας ζωής του, καθώς και στην αποφόρτιση του προγράμματός του, δίνοντας του πολύτιμο χρόνο για διάφορες άλλες ασχολίες.

Ειδικότερα τα τελευταία χρόνια έχει δημιουργηθεί ο εξής όρος: “Διαδίκτυο των πραγμάτων” (Internet of Things ή IoT) που αναφέρεται για την περιγραφή τέτοιων

συσκευών, με παρόμοιες ή και ίδιες λειτουργίες με αυτές που προαναφέρθηκαν. Πιο συγκεκριμένα, ο όρος χαρακτηρίζει ουσιαστικά, τις συσκευές που έχουν την δυνατότητα να συνδέονται στο διαδίκτυο και παράλληλα, να είναι προγραμματισμένες σε τέτοιο βαθμό, που να εκτελούν διάφορες λειτουργίες, σύμφωνες με τον λόγο ύπαρξής τους, χωρίς την παρέμβαση του ανθρώπου. Συνεπώς, δημιουργείται το συμπέρασμα, ότι μπορεί οποιοσδήποτε να ελέγχει τις συσκευές απομακρυσμένα και παράλληλα, να λαμβάνει τις πληροφορίες που η κάθε συσκευή αποστέλλει, ως ανάδραση (feedback) ή ακόμα και να παρακολουθεί σε πραγματικό χρόνο (Real Time) την λειτουργία της κάθε συσκευής.

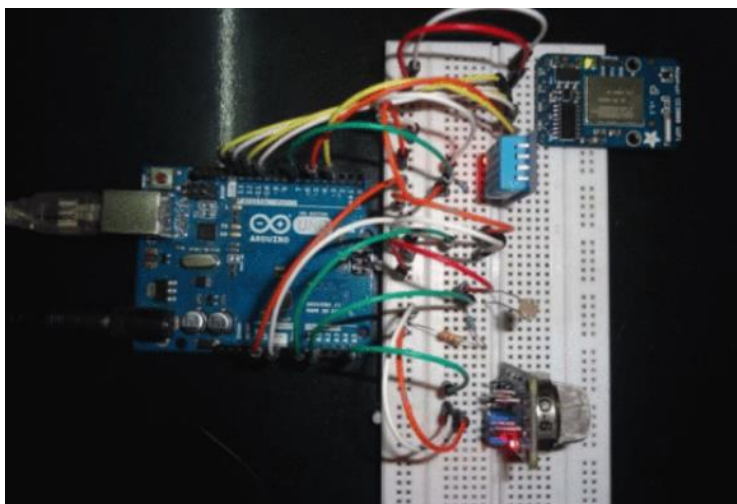
1.2 Ιδέα και σκοπός υλοποίησης

Ο καιρός και τα περιβατολογικά φαινόμενα, αποτελούσαν εξαρχής αντικείμενο καταγραφής και μελέτης από τον άνθρωπο. Λαμβάνοντας υπόψη, όλα τα παραπάνω και συνδυάζοντας τα σύγχρονα τεχνολογικά μέσα που έχουν ήδη καταγραφεί, προέκυψε η ιδέα για την δημιουργία μίας “έξυπνης” συσκευής με δυνατότητα καταγραφής των περιβατολογικών συνθηκών καθώς και δυνατότητα σύνδεσης με το διαδίκτυο (Internet), οικονομική στην υλοποίηση και κατασκευή, μικρή σε μέγεθος για δυνατότητα μεταφοράς και σχεδιασμένη, για ελάχιστη κατανάλωση ενέργειας, η οποία θα είναι τοποθετημένη σε σημεία επιλογής του χρήστη είτε σε κλειστό εσωτερικό χώρο (σπίτι), είτε σε εξωτερικό, προγραμματισμένη ώστε να συλλέγει μετρήσεις, σε τακτά χρονικά διαστήματα επιλεγμένα από τον χρήστη της συσκευής. Παράλληλα, υπάρχει η ανάγκη υλοποίησης μίας υποστηρικτικής διαδικτυακής εφαρμογής, σκοπός της οποίας είναι να συλλέγει τις μετρήσεις που αποστέλλει η συσκευή, παρέχοντας την δυνατότητα στον εκάστοτε χρήστη, να παρακολουθεί τις μετρήσεις που λαμβάνει η συσκευή του. Ο σκοπός της ιδέας όλου αυτού του εγχειρήματος και της υλοποίησής του, που είναι παράλληλα και το αντικείμενο της παρούσης διπλωματικής εργασίας, είναι η δημιουργία ενός ευφυούς συστήματος, υπεύθυνου για την συνεχή ενημέρωση του χρήστη, για τις περιβατολογικές συνθήκες που επικρατούν στο περιβάλλον όπου τοποθετήθηκε, αυτόνομου στην λειτουργία και εκτέλεση και τεχνολογικά εξελιγμένο.

1.3 Αντίστοιχα συστήματα και συμπέρασμα υλοποίησης

Ύστερα από ενδελεχή αναζήτηση σε διαδικτυακούς τόπους εκπαιδευτικού και τεχνολογικού ενδιαφέροντος, παρακάτω, παρουσιάζεται μία λίστα από πανομοιότυπα συστήματα, τα οποία εξυπηρετούν τον σκοπό της παρούσης διπλωματικής εργασίας. Παρουσιάζονται διαφορετικοί τύποι υλοποίησης με διαφορετικές τεχνολογίες και υλικά. Τα τρία πρώτα συστήματα αποτελούν έργα, τα οποία δημοσιεύτηκαν σε ιστοχώρο ερευνητικού και επιστημονικού τομέα και τα ακόλουθα τέσσερα αποτελούν έργα, που παρουσιάζονται σε ιστοχώρους εκπαιδευτικού σκοπού και καθοδήγησης υλοποίησης παρόμοιων συστημάτων. Αφού γίνει μία σύντομη περιγραφή όλων, έπειτα θα συγκριθούν.

1.3.1 Real time standalone data acquisition system for environmental data



Εικόνα 1: Ενδεικτική εικόνα πρώτου αντίστοιχου συστήματος

- Περιγραφή: Η συγκεκριμένη κατασκευή [1], αποτελεί ένα υλοποιημένο ενσωματωμένο σύστημα το οποίο, παίρνει μετρήσεις από αισθητήρες, υπεύθυνους για καταγραφή θερμοκρασίας, υγρασίας, φωτεινότητας, υγραερίου και άλλων τύπων αερίων και μέσω Wi-Fi επικοινωνίας, που επιτυγχάνεται με την βοήθεια εξαρτήματος (module), που λειτουργεί ως κάρτα δικτύου, στέλνει τα δεδομένα σε

μία διαδικτυακή πλατφόρμα υπολογιστικού νέφους (cloud), σχεδιασμένη για εισαγωγή, ανάλυση και εξαγωγή μετρήσεων και δεδομένων, για IoT συσκευές. Ο χρήστης για να δει τις μετρήσεις αυτές διαδικτυακά μέσω της πλατφόρμας, θα πρέπει να κάνει εγγραφή στην υπηρεσία, ώστε να πάρει το μοναδικό για αυτόν κλειδί API (API key), το οποίο είναι μία μέθοδος που συναντάται συχνά πλέον, σε ολοένα και περισσότερους ιστοχώρους και χρησιμεύει, για την αυθεντικοποίηση και αναγνώριση της συσκευής και εν συνεχεία του χρήστη, από το σύστημα. Το απαιτητικό κομμάτι του διαδικασίας περιορίζεται στον προγραμματισμό της συσκευής, ώστε να στοχεύει σε συγκεκριμένο τελικό σημείο, προορισμού ιστοχώρου που δίνεται από την υπηρεσία (API endpoint) μαζί με τις μετρήσεις καθώς και με το κλειδί API. Η υπηρεσία αυτή προσφέρεται δωρεάν και επί πληρωμή ωστόσο, για τους δωρεάν χρήστες διαθέτει περιορισμένες παροχές, σε συχνότητα ανανέωσης των δεδομένων, στον αριθμό των συνδεδεμένων συσκευών κ.α. Τέλος, να αναφερθεί ότι δεν δίνεται κάποιο σχηματικό διάγραμμα για την συνδεσμολογία του συστήματος.

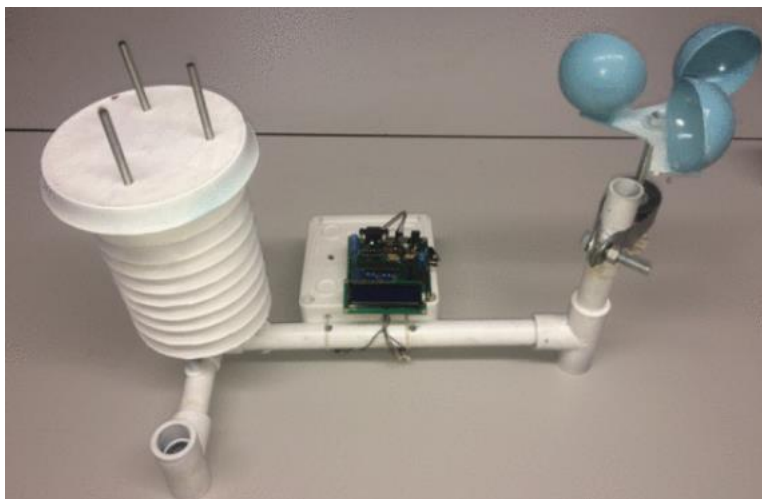
- Υλικά hardware: Το σύστημα αποτελείται από έναν μικροεπεξεργαστή τύπου Arduino Uno. Εξάρτημα για σύνδεση στο διαδίκτυο μέσω Wi-Fi ονομασίας CC3000 WiFi Chip. Αισθητήρα για μέτρηση θερμοκρασίας ονομασίας LM35DZ temperature sensor. Αισθητήρα για μέτρηση υγρασίας ονομασίας DHT11 humidity sensor. Αισθητήρα για μέτρηση ποσότητας φωτεινότητας ονομασίας NORP 12 LDR light sensor. Αισθητήρα για ανίχνευση αερίων (υγραέριο, βουτάνιο, προπάνιο) ονομασίας MQ6 gas sensor. Μία βάση συναρμολόγησης (breadboard) και καλώδια για την σύνδεση όλων των εξαρτημάτων μεταξύ τους.

1.3.2 Weather Station Design Using IoT Platform Based On Arduino Mega

- Περιγραφή: Η συγκεκριμένη κατασκευή [2], αποτελεί ένα υλοποιημένο ενσωματωμένο σύστημα το οποίο, πραγματοποιεί μετρήσεις για θερμοκρασία, υγρασία, ατμοσφαιρική πίεση και παρουσία βροχής. Με την βοήθεια εξαρτήματος, που λειτουργεί ως κάρτα δικτύου, στέλνει τα δεδομένα μέσω Wi-Fi επικοινωνίας

- σε μία διαδικτυακή πλατφόρμα υπολογιστικού νέφους και παράλληλα με την βοήθεια μίας μικρού μεγέθους TFT LCD οθόνης 2,2 ιντσών, παρουσιάζει και τοπικά στο σύστημα τις μετρήσεις. Δεν δίνονται εικόνες από την υλοποίηση, ούτε την συνδεσμολογία του συστήματος.
- Υλικά hardware: Το σύστημα αποτελείται από έναν μικροεπεξεργαστή τύπου Arduino Mega 2560. Εξάρτημα για σύνδεση στο διαδίκτυο μέσω Wi-Fi ονομασίας ESP 8266 Module. Αισθητήρα για μέτρηση θερμοκρασίας και υγρασίας ονομασίας DHT22 temperature and humidity sensor. Αισθητήρα για μέτρηση παρουσίας βροχής, ωστόσο δεν αναφέρεται κάποια περιγραφή του. Αισθητήρα για μέτρηση ατμοσφαιρικής πίεσης ονομασίας BMP180 pressure sensor.

1.3.3 Weather Station for educational purposes based on Atmega8L



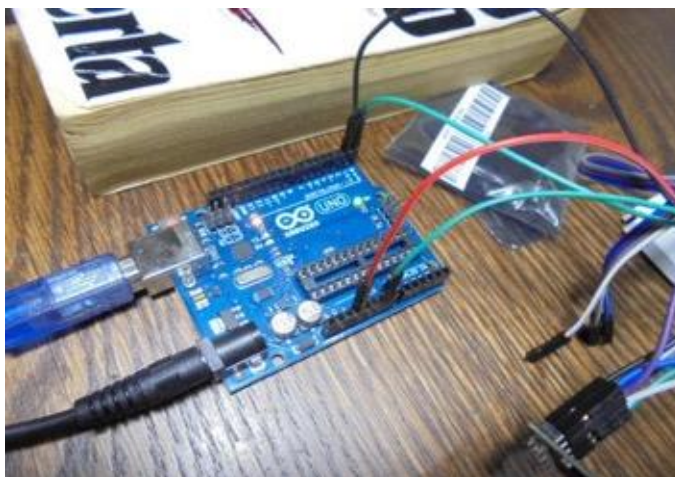
Εικόνα 2: Ενδεικτική εικόνα τρίτου αντίστοιχου συστήματος

- Περιγραφή: Η συγκεκριμένη κατασκευή [3], αποτελείται από δύο ενσωματωμένα συστήματα τα οποία λειτουργούν ως πομπός/δέκτης. Το πρώτο, πραγματοποιεί μετρήσεις για θερμοκρασία, υγρασία, παρουσία βροχής, ταχύτητα ανέμου και υπεριώδους ακτινοβολίας. Με την βοήθεια εξαρτημάτων (που βρίσκονται εγκατεστημένα και στα δύο συστήματα) τα οποία λειτουργούν με τεχνολογία ραδιοκυμάτων, επιτυγχάνεται ασύρματη αποστολή και λήψη δεδομένων. Το

δεύτερο σύστημα το οποίο λειτουργεί ως δέκτης, με τη βοήθεια μίας μικρού μεγέθους TFT LCD οθόνης 2,2 ιντσών, παρουσιάζει τις μετρήσεις. Δεν δίνονται εικόνες από την συνδεσμολογία των συστημάτων.

- Υλικά hardware: Το σύστημα αποτελείται από δύο μικροεπεξεργαστές τύπου ATmega8L (ίδιου τύπου και δυνατοτήτων με το Arduino Uno αλλά πιο οικονομικός). Δύο εξαρτήματα για την δυνατότητα αποστολής δεδομένων μέσω ραδιοκυμάτων ονομασίας nRF2410. Αισθητήρα για μέτρηση θερμοκρασίας και υγρασίας ονομασίας DHT11 temperature and humidity sensor. Αισθητήρα για μέτρηση παρουσίας βροχής, ονομασίας FC-37 και αισθητήρα για μέτρηση ταχύτητας ανέμου ωστόσο δεν αναφέρεται κάποια περιγραφή του.

1.3.4 WiFi ESP8266 and DHT22 sensor

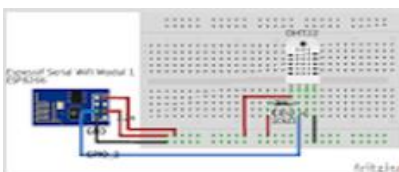


Εικόνα 3: Ενδεικτική εικόνα τέταρτου αντίστοιχου συστήματος

- Περιγραφή: Η συγκεκριμένη κατασκευή [4], αποτελεί ένα υλοποιημένο ενσωματωμένο σύστημα το οποίο, παίρνει μετρήσεις από έναν αισθητήρα θερμοκρασίας-υγρασίας και μέσω Wi-Fi επικοινωνίας, που επιτυγχάνεται με την βοήθεια εξαρτήματος, που λειτουργεί ως κάρτα δικτύου, στέλνει τα δεδομένα σε μία διαδικτυακή πλατφόρμα υπολογιστικού νέφους (cloud), σχεδιασμένη για εισαγωγή, ανάλυση και εξαγωγή μετρήσεων και δεδομένων, για IoT συσκευές ονομασίας: ThingSpeak (ονομασία ιστοχώρου: <https://thingspeak.com/>).

- Υλικά hardware: Το σύστημα αποτελείται από έναν μικροεπεξεργαστή τύπου Arduino Uno. Εξάρτημα για σύνδεση στο διαδίκτυο μέσω Wi-Fi ονομασίας Everything ESP8266 τύπου ESP-01. Έναν αισθητήρα για μέτρηση θερμοκρασίας και υγρασίας ονομασίας DHT22 temperature and humidity sensor. Μία βάση συναρμολόγησης (breadboard) και καλώδια για την σύνδεση όλων των εξαρτημάτων μεταξύ τους.

Παρακάτω ακολουθεί η σχεδίαση του κυκλώματος.



Εικόνα 4: Σχεδίαση κυκλώματος τέταρτου συστήματος

1.3.5 Arduino Weather Station Project

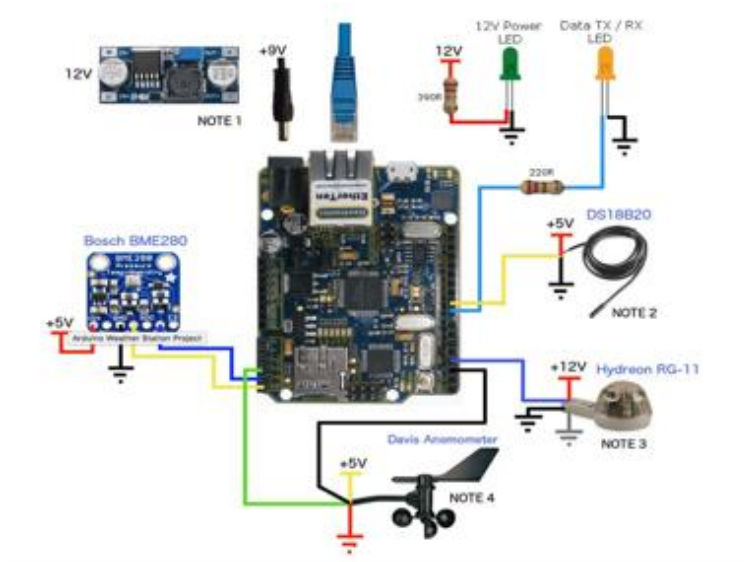


Εικόνα 5: Ενδεικτική εικόνα πέμπτου αντίστοιχου συστήματος

- Περιγραφή: Η συγκεκριμένη κατασκευή [5], αποτελεί έναν τελειοποιημένο περιβαλλοντικό σταθμό, ο οποίος, παίρνει αρκετές μετρήσεις για τις περιβαλλοντικές συνθήκες που επικρατούν (εσωτερική και εξωτερική θερμοκρασία της κατασκευής, υγρασία, πίεση, παρουσία βροχής, ταχύτητα και κατεύθυνση ανέμου) και μέσω ενσωματωμένης Ethernet θύρας που υπάρχει στον συγκεκριμένο τύπο μικροεπεξεργαστή Arduino, αποστέλλει τα δεδομένα

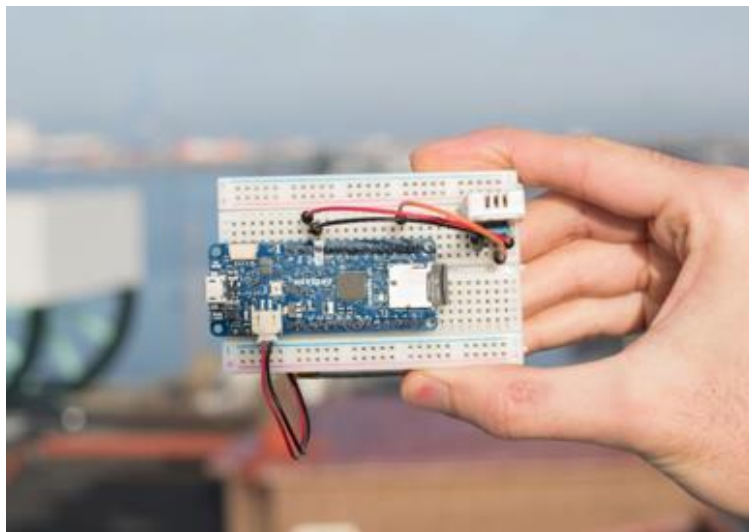
- ενσύρματα μέσω διαδικτύου σε εξυπηρετητή (server). Στην συνέχεια τα δεδομένα αποθηκεύονται σε βάση δεδομένων και προβάλλονται σε διαδικτυακή εφαρμογή (Web Application) με πολύ βασική διεπαφή χρήστη (UI), η οποία έχει υλοποιηθεί και την σερβίρει ο εξυπηρετητής. Η υλοποίηση του συγκεκριμένου συστήματος είναι αρκετά δαπανηρή, εξαιτίας της αξίας κάποιων αισθητήρων και εξαρτημάτων.
- Υλικά hardware: Το σύστημα αποτελείται από έναν μικροεπεξεργαστή τύπου Arduino (Freertronics EtherTen) με την μόνη διαφορά ότι έχει ενσωματωμένη μία θύρα Ethernet στην πλακέτα, με την οποία έχει την δυνατότητα να συνδέεται ενσύρματα στο διαδίκτυο. Έναν αισθητήρα για μέτρηση θερμοκρασίας ονομασίας DS18B20 και έναν αισθητήρα για μέτρηση θερμοκρασίας, υγρασίας και πίεσης ονομασίας BME280. Έναν αισθητήρα για την παρουσία βροχής ονομασίας Hydreon RG-II. Αισθητήρα ανέμου ονομασίας Davis Wind Anemometer, μία βάση συναρμολόγησης (breadboard) και καλώδια για την σύνδεση όλων των εξαρτημάτων μεταξύ τους. Η ανάγκη για παροχή ηλεκτρικής ενέργειας έχει καθοριστεί στα 12V DC λόγω των αισθητήρων ανέμου και παρουσίας βροχής και των απαιτήσεων τους για την ορθή λειτουργία τους, που επιτυγχάνονται με την συγκεκριμένη τιμή τάσης. Ωστόσο τα 12V είναι η μέγιστη συνιστώμενη τάση εισόδου από τον κατασκευαστή και παρόλο, που τα μετατρέπει ο μικροεπεξεργαστής μέσω του ενσωματωμένου ρυθμιστή τάσης, σε τιμή των 5V για την ασφαλή λειτουργία των υπόλοιπων εξαρτημάτων, ο ρυθμιστής τάσης υπερθερμαίνεται και μπορεί να συμβούν μοιραία αποτελέσματα με την πάροδο του χρόνου στον μικροεπεξεργαστή.

Παρακάτω ακολουθεί η σχεδίαση του κυκλώματος.



Εικόνα 6: Σχεδίαση κυκλώματος πέμπτου συστήματος

1.3.6 MKRZero Weather Data Logger



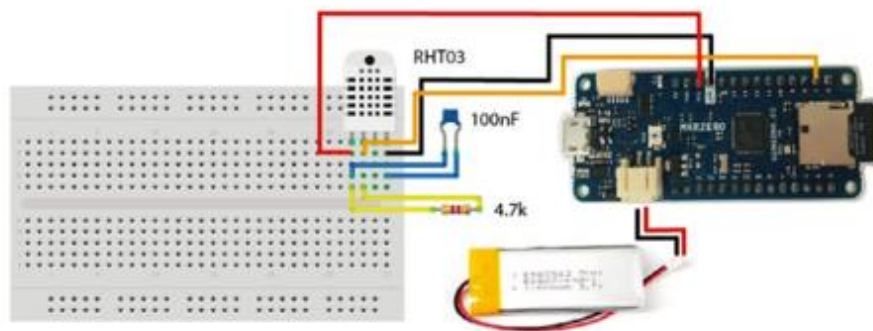
Εικόνα 7: Ενδεικτική εικόνα έκτου αντίστοιχου συστήματος

- Περιγραφή: Η συγκεκριμένη κατασκευή [6], υλοποιεί ένα απλό ενσωματωμένο σύστημα το οποίο καταγράφει τις περιβαλλοντικές συνθήκες (θερμοκρασία, υγρασία) και τις αποθηκεύει σε κάρτα αποθήκευσης τύπου SD (SD card), χωρίς να παρουσιάζει κάποιο είδος επικοινωνίας. Παρόλο που η μελέτη της διπλωματικής εργασίας, έχει επικεντρωθεί μέχρι στιγμής μόνο σε συστήματα, που έχουν

δυνατότητα αποστολής δεδομένων μέσω διαδικτύου και αλληλεπίδρασης με τον χρήστη, είτε ασύρματα είτε ενσύρματα, πολλές φορές δεν δύναται η χρησιμοποίηση τέτοιων μεθόδων, λόγω αδυναμίας δικτύου ή ακόμα και μη ύπαρξη δικτύου στην περιοχή ή κάλλιστα, ο σκοπός του συστήματος είναι να τοποθετηθεί σε απρόσιτη και απομακρυσμένη περιοχή, ώστε να συλλέγει και να καταγράφει τιμές τοπικά. Ωστόσο με αυτή την προσέγγιση, καθίσταται δυνατή αυτή η καταγραφή.

- Υλικά hardware: Το σύστημα αποτελείται από έναν μικροεπεξεργαστή τύπου Arduino MKRZero. Είναι μία μικρότερη έκδοση του Arduino Zero που τρέχει στα 3.3V, σε αντίθεση με τους μεγαλύτερους μικροεπεξεργαστές της οικογενείας του (Arduino, Genuino) που τρέχουν στα 5V. Επίσης ο συγκεκριμένος μικροεπεξεργαστής έχει ενσωματωμένη υποδοχή για κάρτα αποθήκευσης τύπου SD. Ακόμη χρησιμοποιούνται ένας DHT22 αισθητήρας για καταγραφή θερμοκρασίας και υγρασίας, ένας πυκνωτής χωρητικότητας 100nF, μία αντίσταση 4.75kΩ, μία βάση συναρμολόγησης (breadboard) και καλώδια για την σύνδεση μεταξύ τους.

Παρακάτω ακολουθεί η σχεδίαση του κυκλώματος.



Εικόνα 8: Σχεδίαση κυκλώματος έκτου συστήματος

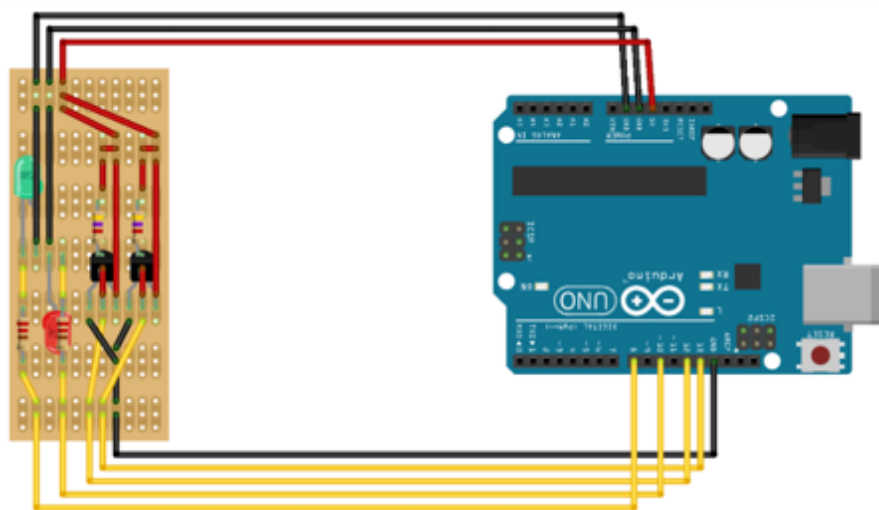
1.3.7 Measuring water and air temperature with Arduino GSM shield



Εικόνα 9: Ενδεικτική εικόνα έβδομου αντίστοιχου συστήματος

- Περιγραφή: Η κατασκευή [7] υλοποιεί ένα ενσωματωμένο σύστημα το οποίο, μετράει την θερμοκρασία νερού και αέρα και μέσω τεχνολογίας GSM (Global System for Mobile Communication), αποστέλλει τα δεδομένα μέσω ενός GPRS δικτύου (General Packet Radio Services). Τα δεδομένα αποστέλλονται σε διαδικτυακή cloud πλατφόρμα (αναλύθηκε εκτενώς στο πρώτο σύστημα).
- Υλικά hardware: Το σύστημα αποτελείται από έναν μικροεπεξεργαστή τύπου Arduino Uno, μία Arduino GSM πλακέτα, η οποία χρησιμοποιείται για την αποστολή δεδομένων, μέσω του κινητού δικτύου, έναν αισθητήρα θερμοκρασίας ονομασίας DS18B20, μία κάρτα SIM με συμφωνία μεταξύ τηλεφωνικού παρόχου για προπληρωμένο πλάνο δεδομένων, υποδοχή για μπαταρίες τύπου AA με σκοπό την τροφοδοσία, διάφορες αντιστάσεις και LED, όπως και καλώδια για σύνδεση μεταξύ τους.

Παρακάτω ακολουθεί η σχεδίαση του κυκλώματος.



Εικόνα 10: Σχεδίαση κυκλώματος έβδομου συστήματος

1.4 Ανάλυση και συμπεράσματα αντίστοιχων συστημάτων

Αφού μελετήθηκαν προσεκτικά όλα τα συστήματα, οι τεχνολογίες και υπηρεσίες που χρησιμοποιήθηκαν, τα εξαρτήματα και οι αισθητήρες, παρατηρήθηκε, ότι όλα τα συστήματα λειτουργούν σύμφωνα με τα παρακάτω πλαίσια:

- Έχουν μοναδική μορφή επικοινωνίας και αποστολής δεδομένων. Κανένα σύστημα δεν δίνει την δυνατότητα στον χρήστη για πολλαπλή επικοινωνία, ή την δυνατότητα για κάποια εναλλακτική μορφή αποθήκευσης ή επικοινωνίας, σε περίπτωση που δεν υφίσταται η συγκεκριμένη για την οποία προορίζεται. Παράδειγμα μπορεί να θεωρηθεί, η αποτυχημένη αποστολή δεδομένων, μέσω ασύρματης επικοινωνίας με το διαδίκτυο (Wi-Fi) λόγω παρουσίας κάποιου τεχνικού προβλήματος στο δίκτυο Wi-Fi, που έχει ως αποτέλεσμα, την μη αποστολή των δεδομένων, διότι δεν υλοποιήθηκε εναλλακτικό πλάνο αποστολής ή αποθήκευσης. Ενέργεια, που αυτομάτως υφίσταται ως αποτυχία του σχεδίου και αστοχία της κατασκευής.

- **Χρησιμοποίηση εξαρτημάτων και υπηρεσιών με σχετικό κόστος.** Στην πλειοψηφία των συστημάτων, δεν υλοποιήθηκαν διαδικτυακές εφαρμογές, ωστόσο οι ερευνητές κατέφυγαν στην εύκολη, εύχρηστη και συνάμα περιορισμένη λύση των διαδικτυακών εφαρμογών τύπου cloud, στοιχείο που φανερώνει, είτε αδυναμία υλοποίησης διαδικτυακής εφαρμογής ή την μη επιθυμία για περαιτέρω αύξηση της πολυπλοκότητας. Παράλληλα, το κόστος ορισμένων αισθητήρων που χρησιμοποιήθηκαν, ξεπερνάει κατά πολύ το προσδοκώμενο κόστος, ολόκληρου του συστήματος που αναπτύχθηκε, στην παρούσα διπλωματική εργασία.
- **Δεν πραγματοποιήθηκε καμία μελέτη στην μείωση κατανάλωσης ενέργειας και λειτουργίας των συστημάτων.** Παρατηρήθηκε σε όλα τα συστήματα, ότι δεν έγινε καμία έρευνα για μείωση της κατανάλωσης ενέργειας μέσω προγραμματιστικού κώδικα, στοιχείο που δηλώνει ότι τα συστήματα αυτά είτε είναι αποκλειστικά εξαρτημένα, από την συνεχή παροχή ηλεκτρικής ενέργειας, είτε δεν παρουσιάζουν καμία αυτονομία σε μεγάλη διάρκεια χρήσης.

Συνοψίζοντας όλα τα ωφέλιμα στοιχεία, όλες τις περιπτώσεις χρήσης που μπορεί να προκύψουν, καθώς και το μέγιστο δυνατό υπολογισμό στην σχέση κόστους-ποιότητας, πάρθηκε η απόφαση για δημιουργία ενός “έξυπνου” αυτοματοποιημένου ενσωματωμένου συστήματος, που λαμβάνει συστηματικές μετρήσεις σε καθορισμένα χρονικά διαστήματα, για έξι περιβαλλοντικά στοιχεία (θερμοκρασία, υγρασία, ατμοσφαιρική πίεση, UV ακτινοβολία, αίσθηση βροχής, συγκέντρωση σκόνης), προσφέρει στον χρήστη δυνατότητες για αποστολή δεδομένων ασύρματα και ενσύρματα με κατάλληλους ελέγχους (Wi-Fi και Ethernet), με παράλληλη δυνατότητα αποθήκευσης δεδομένων σε κάρτα αποθήκευσης SD, σε περίπτωση απώλειας δικτύου, με σκοπό την αποστολή τους σε μελλοντικό χρόνο αφού επανέλθει η σύνδεση με το διαδίκτυο. Τέλος, βελτιστοποιήθηκε στο να έχει την μικρότερη δυνατή κατανάλωση ενέργειας, λόγω του προγραμματισμού του, που το θέτει σε κατάσταση χαμηλής λειτουργίας (sleep mode), αμέσως μετά την αποστολή μετρήσεων και επανέρχεται σε κανονική λειτουργία, λίγο πριν την επόμενη αποστολή. Παράλληλα με το ενσωματωμένο σύστημα, πάρθηκε η απόφαση για υλοποίηση διαδικτυακής εφαρμογής, η οποία σερβίρεται από εξυπηρετητή, με υλοποιημένη βάση

δεδομένων για αποθήκευση των μετρήσεων και ανάκλησή τους όταν αυτό ζητηθεί, με φιλική διεπαφή χρήστη (UI friendly), καθώς και αποκριτική σχεδίαση (responsive design) ώστε να φαίνεται σε κάθε είδος συσκευής, οποιονδήποτε διαστάσεων, με την ίδια ευκρίνεια και απόκριση.

1.5 Σύνοψη διπλωματικής εργασίας

Στο κείμενο αυτό αναλύεται η διαδικασία σχεδίασης και υλοποίησης του ενσωματωμένου συστήματος καθώς και της διαδικτυακής εφαρμογής. Αποτελείται από πέντε κεφάλαια.

Το τρέχων κεφάλαιο αποτελεί την εισαγωγή στο θέμα το οποίο πραγματεύεται η διπλωματική εργασία καθώς και τον λόγο για τον οποίο προέκυψε.

Στο δεύτερο κεφάλαιο γίνεται πλήρη καταγραφή και ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν, κατά την διάρκεια της κατασκευής του ενσωματωμένου συστήματος καθώς και της διαδικτυακής εφαρμογής, προκειμένου να προσαρμοστεί ο αναγνώστης στο γνωστικό αντικείμενο.

Το τρίτο κεφάλαιο πραγματεύεται τις απαιτήσεις του συστήματος για το σχεδιασμό της βάσης δεδομένων, την λεπτομερή περιγραφή των οντοτήτων, των δεδομένων και των συσχετίσεων που δημιουργήθηκαν. Επιπλέον, περιγράφονται οι τεχνικές που χρησιμοποιήθηκαν για την ασφάλεια του συστήματος.

Στο τέταρτο κεφάλαιο παρουσιάζεται το υλικό μέρος της εργασίας, τα τεχνικά χαρακτηριστικά κάθε εξαρτήματος, ο τρόπος λειτουργίας τους, καθώς επίσης και αναλυτικά σχεδιαγράμματα των κυκλωμάτων που υλοποιήθηκαν.

Στο πέμπτο κεφάλαιο αναπτύσσεται η δημιουργία του λογισμικού μέρους της διαδικτυακής εφαρμογής. Παρουσιάζονται και αναλύονται οι λειτουργίες και η διεπαφή χρήστη μέσω διαφόρων στιγμιότυπων και πιθανών σεναρίων χρήσης του συστήματος.

Στο έκτο, και τελευταίο, κεφάλαιο παρουσιάζονται τα συμπεράσματα από την ανάπτυξη του συστήματος, τα προβλήματα που προέκυψαν κατά την υλοποίηση, και πιθανές μελλοντικές επεκτάσεις και βελτιώσεις.

Κεφάλαιο 2ο – Θεωρητικό υπόβαθρο

Σε αυτό το κεφάλαιο, παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την εκπόνηση της εργασίας. Αποσαφηνίζονται θεωρητικές έννοιες που αφορούν τον προγραμματισμό διαδικτύου, της αρχιτεκτονικής Arduino και τέλος, αναλύονται οι πλατφόρμες ανάπτυξης λογισμικού που χρησιμοποιήθηκαν. Διότι, ο όγκος των τεχνολογιών, που χρησιμοποιήθηκαν στην διαδικτυακή εφαρμογή, είναι αρκετά μεγάλος και θα ήταν εξαιρετικά χρονοβόρο, να αναλυθούν όλες στον ίδιο βαθμό, θα αναλυθούν ιδιαίτερα οι τεχνολογίες που χρησιμοποιήθηκαν κατά κόρον και σε μικρότερο βαθμό, αυτές που χρησιμοποιήθηκαν ως υποστηρικτικές.

2.1 Προγραμματισμός διαδικτύου

Η επιστήμη της πληροφορικής που ασχολείται με τη δημιουργία διαδικτυακών εφαρμογών ιστοχώρου ονομάζεται Προγραμματισμός Διαδικτύου (Web Development) [8]. Η διαδικασία αυτή, της δημιουργίας ενός διαδικτυακού ιστοχώρου ή ιστοσελίδας (ισότιμοι όροι που περιγράφουν το ίδιο στοιχείο), μπορεί να κυμαίνεται από την ανάπτυξη μίας ενιαίας απλής σελίδας έως μία σειρά πολύπλοκων ιστοχώρων. Ο προγραμματισμός διαδικτύου περιλαμβάνει πολλές ενέργειες, μερικές από τις οποίες είναι ο σχεδιασμός του ιστοχώρου, η δημιουργία του περιεχομένου, ο προγραμματισμός, η διαμόρφωση της ασφάλειας του δικτύου, η ανάπτυξη σεναρίων για τον εξυπηρετητή (server) και τον πελάτη (client), καθώς και η δημιουργία συστημάτων για τη διαχείριση του περιεχομένου γνωστά ως Content Management Systems (CMS [33]), τα οποία αποτελούν τον ενδιάμεσο κρίκο μεταξύ του χρήστη και της βάσης δεδομένων.

Ο προγραμματισμός διαδικτύου μπορεί να διαιρεθεί στις εξής τρεις κατηγορίες: στον υπολογιστή εξυπηρετητή (web server), στον υπολογιστή πελάτη (web client) και τέλος στο σύστημα διαχείρισης βάσεων δεδομένων (Database Management System ή DBMS [32]). Ο όρος web server [9] αναφέρεται σε εκείνον τον υπολογιστή ο οποίος είναι υπεύθυνος για την διεκπεραίωση των αιτημάτων που του θέτουν οι web clients. Σε αυτόν τον υπολογιστή εκτελούνται γλώσσες προγραμματισμού εξυπηρετητή (server-side

programming) για την παραγωγή HTML εγγράφων. Η έννοια web client αναφέρεται στον υπολογιστή του χρήστη στον οποίο συνήθως εκτελείται ένα πρόγραμμα περιήγησης στο διαδίκτυο (web browser). Ο περιηγητής δημιουργεί και αποστέλλει αιτήματα προς τον web server και στην συνέχεια παραλαμβάνει τις απαντήσεις τις οποίες εν τέλει εμφανίζει στον χρήστη. Μία ακόμα αρμοδιότητα του browser είναι να εκτελεί γλώσσες προγραμματισμού πελάτη (client-side programming). Το σύστημα βάσης δεδομένων [10] θεωρείται μία ξεχωριστή κατηγορία παρόλο που μπορεί να εκτελείται και στον υπολογιστή εξυπηρετητή. Ναι μεν είναι αναπόσπαστο κομμάτι του προγραμματισμού διαδικτύου, αλλά παρόλα αυτά, είναι σύστημα γενικού σκοπού, το οποίο χρησιμοποιείται σε πολλές άλλες εφαρμογές οι οποίες δεν είναι απαραίτητα διαδικτυακές εφαρμογές. Τα συστήματα βάσεων δεδομένων είναι υπεύθυνα για την αποθήκευση και την ανάκτηση τις δυναμικής πληροφορίας που προέρχεται από τους χρήστες και από συσκευές καταγραφής δεδομένων. Χαρακτηριστικό παράδειγμα, που κάλλιστα θα μπορούσε να περιγράψει και την λειτουργία του συστήματος, παράλληλα με την διαδικτυακή εφαρμογή της διπλωματικής εργασίας, αποτελεί μία συσκευή, οι οποία λαμβάνει μέτρηση για την θερμοκρασία, ανά τακτά χρονικά διαστήματα και στέλνει στον server, την τρέχουσα τιμή, του χώρου στον οποίο βρίσκεται.

2.1.1 HTML

Η HTML (HyperText Markup Language ή Γλώσσα Σήμανσης Κειμένου) [11] είναι η βασική γλώσσα σήμανσης στο διαδίκτυο για την δημιουργία ενός ιστοχώρου. Η HTML δεν θεωρείται γλώσσα προγραμματισμού αλλά είναι ένα έγγραφο με οδηγίες προς τους browsers για το πως να εμφανίσουν το περιεχόμενο στους χρήστες. Όπως είναι φανερό η HTML “εκτελείται” στον υπολογιστή πελάτη μέσω του browser. Τα βασικά δομικά συστατικά της HTML είναι τα HTML στοιχεία (HTML elements) τα οποία αποτελούνται από ετικέτες (tags). Τα tags είναι λέξεις ή γράμματα που περικλείονται από γωνιώδεις αγκύλες και υπάρχουν ανά ζεύγη. Δηλαδή, υπάρχει ένα tag που σηματοδοτεί την έναρξη και ένα που σηματοδοτεί την λήξη με μερικές εξαιρέσεις. Το tag λήξης περιέχει μία πλάγια γραμμή ‘/’ για να διαχωρίζεται από το tag έναρξης. Ένα παράδειγμα tag, που είναι και ένα από τα πιο σημαντικά στοιχεία, για την δήλωση ενός HTML εγγράφου είναι το εξής:

<html>...</html>. Τα tags μέσα τους περικλείουν κείμενο ή και άλλα εσωτερικά tags (εμφωλευμένα tags). Μερικά βασικά tags παραθέτονται στον πίνακα 1 που ακολουθεί.

HTML στοιχείο	Περιγραφή
<!DOCTYPE>	Αποτελεί οδηγία που καθορίζει την έκδοση της HTML που χρησιμοποιείται
<html>	Περικλείει όλο το HTML έγγραφο
<head>	Παρέχει πληροφορίες σχετικές με το έγγραφο όπως γλώσσα, κωδικοποίηση καθώς και μεταδεδομένα
<title>	Ορίζει τον τίτλο του ιστοχώρου
<body>	Περιγράφει το οπτικό περιεχόμενο της σελίδας, το οποίο θα εμφανιστεί στον χρήστη
<div>	Ομαδοποιεί στοιχεία μεταξύ τους και τα καθορίζει σαν ένα ξεχωριστό τμήμα της σελίδας. (Χρησιμοποιείται σχεδόν πάντα)
<!-- ... -->	Ορίζει σχόλια
<h1>,<h2>,...,<h6>	Ορίζει επικεφαλίδες τμημάτων κειμένου
<p>	Ορίζει παράγραφο
<button>	Ορίζει κουμπί
<form>	Ορίζει φόρμα
<input>	Ορίζει πεδίο εισαγωγής

Πίνακας 1: Παρουσίαση βασικών στοιχείων HTML

Από τις αρχές του διαδικτύου και των ιστοχώρων, υπάρχουν αρκετές εκδόσεις HTML οι οποίες μεταλλάσσονται στον χρόνο, με την πιο πρόσφατη έκδοση που χρησιμοποιείται μέχρι και σήμερα να είναι η HTML5. Η HTML γενικότερα, δημιουργεί τη βασική δομή ενός ιστοχώρου με αποτέλεσμα να υπάρχουν και άλλες τεχνολογίες που τον συμπληρώνουν διαμορφώνοντας έναν πιο ευπαρουσίαστο ιστοχώρο, όπως η CSS. Από την άλλη η Javascript βελτιώνει την λειτουργικότητα του, καθώς δίνει και την δυνατότητα αλληλεπίδρασης με τον χρήστη.

2.1.2 CSS

Η CSS (Cascading Style Sheets ή Διαδοχικά Φύλλα Στυλ) [12] χρησιμοποιείται για να περιγράψει τον τρόπο με τον οποίο το HTML έγγραφο θα εμφανιστεί στην οθόνη του χρήστη, σε οθόνες διαφορετικού μεγέθους αλλά και σε διάφορες άλλες περιπτώσεις. Η CSS η οποία δεν θεωρείται ούτε αυτή καθαρή γλώσσα προγραμματισμού, δημιουργήθηκε για να διαχωριστούν οι εντολές της εμφάνισης από τις εντολές του περιεχομένου μίας

ιστοσελίδας. Η CSS βοηθάει στο να δημιουργηθεί μία ομορφότερη ιστοσελίδα, οπτικά. Δεν υπάρχει ιστοσελίδα στο διαδίκτυο η οποία να έχει υποστεί στιλιστική επεξεργασία δίχως την χρήση της CSS. “Εκτελείται”, επίσης, από τον browser του χρήστη. Το συντακτικό της είναι σχετικά απλό. Αποτελείται από μία λέξη ή ένα γράμμα τα οποία υποδηλώνουν ένα συγκεκριμένο στοιχείο της HTML και αυτό ονομάζεται επιλογέας (css selector). Έπειτα, ακολουθούν μία ή περισσότερες εντολές που περικλείονται μέσα σε αγκύλες τύπου αγκίστρου (curly braces). Κάθε εντολή βρίσκεται στην δική της γραμμή όπου και τερματίζεται με το ελληνικό ερωτηματικό. Τέλος, η κάθε εντολή αποτελείται από δύο μέρη, την ιδιότητα, η οποία δείχνει ποια ιδιότητα του HTML στοιχείου θα τροποποιηθεί και την τιμή, που θα λάβει η τρέχουσα ιδιότητα. Η ιδιότητα διαχωρίζεται από την τιμή με την άνω-κάτω τελεία. Σήμερα η έκδοση που χρησιμοποιείται είναι η CSS3. Στην παρακάτω εικόνα απεικονίζεται ένα παράδειγμα σύνταξης CSS.

```
background: yellow;
}
body {
  width: 80%;
  margin: 0 auto;
  background: white;
}
header{
  height: 50px;
  margin-bottom: 1em;
```

Εικόνα 11: Παράδειγμα σύνταξης CSS

2.1.3 Javascript

Η Javascript (JS: συντομογραφία, ECMAScript: επίσημη ονομασία, VanillaJS: συνώνυμη ονομασία για Javascript) [13] είναι μία γλώσσα προγραμματισμού σεναρίου (scripting language) η οποία εκτελείται από τον browser στον υπολογιστή πελάτη. Ως scripting γλώσσα απαιτεί την ύπαρξη κάποιας μηχανής εκτέλεσης, κάτι το οποίο αναλαμβάνει να το κάνει ο browser. Πλέον όλοι οι σύγχρονοι browsers μπορούν να εκτελούν την JS χωρίς την ανάγκη κάποιου προσθέτου. Η JS χρησιμοποιείται για να βελτιωθεί η εμπειρία χρήσης και δίνει την δυνατότητα αλληλεπίδρασης με τον χρήστη.

Μπορεί να ανανεώνει δυναμικά τα κομμάτια μίας ιστοσελίδας χωρίς να χρειάζεται η πλήρης ανανέωση της, γεγονός που απαιτεί περισσότερο χρόνο και είναι εμφανές στον χρήστη. Επίσης, η JS μπορεί να ζητήσει δεδομένα στο παρασκήνιο από τον web server χωρίς να διαγράψει το περιεχόμενο που κατά πάσα πιθανότητα ο χρήστης διαβάζει και μόλις αυτό είναι διαθέσιμο το εισάγει δυναμικά στην τρέχουσα ιστοσελίδα. Ουσιαστικά η JS έχει την δυνατότητα να αφαιρεί και να προσθέτει HTML στοιχεία, CSS κανόνες, να τροποποιεί συγκεκριμένες ιδιότητες των HTML στοιχείων μεταβάλλοντας τις τιμές τους. Με την πάροδο των χρόνων, έκαναν την εμφάνισή τους, διάφορες εκδόσεις της JS. Σήμερα, η έκδοση που “καταλαβαίνουν” όλοι οι browsers είναι η ES5 (ECMAScript 5), η οποία εμφανίστηκε το 2009. Η τελευταία μεγάλη αναβάθμιση της γλώσσας έγινε το 2015 με την έλευση της ES6 (ECMAScript 2015). Από τότε μέχρι και σήμερα, ανά χρόνο, γίνεται μία ελαφριά αναβάθμιση της γλώσσας με καινούργιες λειτουργίες. Ωστόσο, από την τελευταία μεγάλη αναβάθμιση της γλώσσας μέχρι και σήμερα, έχει δημιουργηθεί ένα πρόβλημα όσον αφορά την συμβατότητα της γλώσσας με τους browsers και αξίζει να αναφερθεί. Όλοι οι browsers στην ουσία, αποτελούν λογισμικά, υλοποιημένα από διαφορετικές ομάδες προγραμματιστών και εταιρείες, που επιτρέπουν τον χρήστη να εισέλθει στο διαδίκτυο. Ωστόσο, καθώς η γλώσσα ανανεώνεται και προσθέτονται καινούργια στοιχεία, είναι ευθύνη της εκάστοτε εταιρείας να αναβαθμίσει και αυτή το λογισμικό της, ώστε να καταλαβαίνει τις αλλαγές και τις καινούργιες εντολές. Αυτή η διαδικασία σαφώς απαιτεί χρόνο και πραγματοποιείται ταχύτερα από κάποιες εταιρείες, πιο αργά από άλλες ή και καθόλου από ορισμένες. Το βάρος αυτομάτως πέφτει πάνω στα χέρια του προγραμματιστή, ο οποίος, στην δημιουργία ενός ιστοχώρου, θα πρέπει να λάβει υπόψη ότι μπορεί να υπάρχουν χρήστες, που εισέρχονται από διαφορετικούς browser. Βεβαίως αυτό μπορούσε να αποτελέσει ένα μεγάλο και πολλές φορές μοιραίο εμπόδιο διότι, θα πρέπει να συγγράψει κώδικα που να καταλαβαίνουν όλοι οι browsers, πράγμα ουσιαστικά αδύνατο. Την λύση στο πρόβλημα, δίνουν διάφοροι μεταγλωττιστές (compilers), που σαν αποστολή έχουν την μετατροπή της γλώσσας σε μία “κοινή διάλεκτο” που καταλαβαίνουν όλοι οι browsers. Ένας από τους πιο δημοφιλείς μεταγλωττιστές χρησιμοποιήθηκε και στην διαδικτυακή εφαρμογή της διπλωματικής εργασίας (αναλύεται παρακάτω). Τέλος, αξίζει να σημειωθεί ότι, ενώ η JavaScript ήταν καθαρά μία γλώσσα προγραμματισμού που εκτελούνταν αυστηρά στον client, πλέον έχει

αρχίσει να χρησιμοποιείται και στην πλευρά των υπολογιστών εξυπηρετητών. Παράδειγμα κώδικα σε JS παρουσιάζει η εικόνα παρακάτω.

```
function addSubmitEvent(form, func) {
  var oldSubmit = form.onsubmit;
  if (typeof oldSubmit != 'function') {
    form.onsubmit = func;
  }
  else {
    form.onsubmit = function() {
      return oldSubmit() && func();
    }
  }
}
```

Εικόνα 12: Παράδειγμα σύνταξης JS

2.1.4 MYSQL

Η MySQL [14] είναι ένα σύστημα το οποίο συγκαταλέγεται στην κατηγορία των σχεσιακών συστημάτων διαχείρισης βάσεων δεδομένων (RDBMS). Μία βάση δεδομένων αποτελεί τον πυρήνα κάθε συστήματος, μιας και εκεί αποθηκεύονται όλες οι πληροφορίες. Η βάση δεδομένων είναι αναπόσπαστο κομμάτι μίας εφαρμογής που έχει σαν αποστολή, να αποθηκεύει, να ανακτά και να εμφανίζει δεδομένα. Παρόλα αυτά συνιστά ξεχωριστή κατηγορία γιατί είναι πολύπλοκη, και επειδή σχετίζεται με άλλες εφαρμογές εκτός του προγραμματισμού διαδικτύου. Ένα DBMS είναι υπεύθυνο για την αποδοτική αποθήκευση των δεδομένων σε κάποιο μέσο μόνιμης αποθήκευσης με σκοπό την όσο το δυνατόν πιο γρήγορη ανάκτηση τους όταν αυτή ζητηθεί. Η επικοινωνία με μία βάση δεδομένων επιτυγχάνεται μέσω ενδιάμεσων προγραμμάτων, τους οδηγούς (drivers). Οι τελευταίοι λαμβάνουν μία συγκεκριμένη μορφή εντολών, την επεξεργάζονται κατάλληλα και την μεταφράζουν σε γλώσσα κατανοητή από την βάση δεδομένων. Στην συνέχεια την αποστέλλουν και έπειτα αναμένουν την πληροφορία την οποία μεταφέρουν στον αιτούντα. Οι εντολές αυτές ανήκουν στην γλώσσα SQL (Structured Query Language). Υπάρχουν δύο ειδών συστήματα διαχείρισης βάσεων δεδομένων, τα σχεσιακά και μη σχεσιακά. Η επιλογή του συστήματος έχει να κάνει με την πολυπλοκότητα της εφαρμογής που πρόκειται να χτιστεί, την πολυπλοκότητα στις σχέσεις μεταξύ των διάφορων οντοτήτων,

καθώς και στον συνδυασμό των τεχνολογιών που πρόκειται να χρησιμοποιηθούν. Παράδειγμα εντολής SQL παρουσιάζει η εικόνα παρακάτω.

```
CREATE TABLE public."bridge.customeractions" AS
SELECT
  e."id",
  e."cust_application_id" AS "customer_object_id",
  e."user_id" AS "customer_consolidated_id",
  e."created_at" AS "activity_date",
  CASE
    WHEN e."code_action" = 'new' THEN 'Registered'
    WHEN e."code_action" = 'rejected' THEN 'Rejected'
    WHEN e."code_action" = 'canceled' THEN 'Cancelled'
    ELSE 'XNA'
  END AS "status",
  sys."customer_consolidated_id" AS "customerid"
FROM inbound."in.event" e
```

Εικόνα 13: Παράδειγμα SQL εντολής

2.1.5 PHP

Η PHP [15] είναι μία σεναριακή γλώσσα προγραμματισμού εξυπηρετητή (scripting server-side language), που είναι κατάλληλη για τη δημιουργία ιστοχώρων με δυναμικό περιεχόμενο. Πρόκειται ίσως για την πιο διαδεδομένη γλώσσα προγραμματισμού εξυπηρετητή, του κλάδου του προγραμματισμού διαδικτύου, λόγω της εύκολης και γρήγορης διαμόρφωσης που προσφέρει (configuration). Πάνω σε αυτή, έχει υλοποιηθεί το μεγαλύτερο ποσοστό ιστοχώρων στο διαδίκτυο, το οποίο κατέχει μέχρι και σήμερα και μάλιστα, την επέλεξαν ιστοχώροι οι οποίοι εξελίχτηκαν σε διαδικτυακούς κολοσσούς όπως είναι το Facebook, το Wikipedia, το Yahoo κ.α. Ένα αρχείο PHP μπορεί να περιέχει κείμενο, κώδικα HTML, CSS, JavaScript και PHP, με τον κώδικα να εκτελείται στον server και το αποτέλεσμα να επιστρέφει στον browser ως απλές σελίδες HTML. Πιο συγκεκριμένα, η PHP μπορεί να δημιουργήσει, να ανοίξει, να κλείσει, να διαβάσει, να γράψει και να διαγράψει αρχεία σε έναν server. Αποτελεί τον συνδετικό κρίκο μεταξύ του ιστοχώρου και της βάσης δεδομένων, αφού μόνο μέσω της PHP πραγματοποιούνται ενέργειες δημιουργίας, επεξεργασίας και διαγραφής δεδομένων σε αυτή. Ακόμη μπορεί να κρυπτογραφήσει δεδομένα. Η PHP, ως γλώσσα σεναριακού προγραμματισμού, ενσωματώνεται πολύ εύκολα στον κώδικα της HTML. Ο κώδικας της πρώτης πρέπει να βρίσκεται εντός της ετικέτας <? php . . .?> και τα αρχεία που περιέχουν κώδικα PHP και

HTML πρέπει να έχουν υποχρεωτικά την κατάληξη .php. Η PHP τρέχει σε όλες τις πλατφόρμες, όπως των Windows, Linux, Unix, Mac OS X κ.λπ. Είναι συμβατή με όλους σχεδόν τους server που χρησιμοποιούνται σήμερα (Apache, IIS, Enginx κ.λπ.). Υποστηρίζει ένα ευρύ φάσμα από Βάσεις Δεδομένων, είναι δωρεάν και είναι εύκολη στην εκμάθηση. Επίσης, η PHP, ως γλώσσα δομημένου προγραμματισμού, είναι ιδιαίτερα επεκτάσιμη και παρέχει πληθώρα έτοιμων βιβλιοθηκών συναρτήσεων (Functions) προς χρήση, ενώ ταυτόχρονα δίνει τη δυνατότητα στον προγραμματιστή να δημιουργήσει και τις δικές του συναρτήσεις. Τέλος, η PHP επιτρέπει την χρήση αντικειμενοστραφούς προγραμματισμού (Object Oriented Programming ή OOP), που πλέον στην σημερινή εποχή χρησιμοποιείται σαν μοτίβο για την ανάπτυξη οποιασδήποτε εφαρμογής, σε οποιαδήποτε γλώσσα προγραμματισμού. Η σημερινή έκδοση που χρησιμοποιείται, είναι η PHP 7.2. Παράδειγμα κώδικα σε JS παρουσιάζει η εικόνα παρακάτω.

```
<?php
if (!isset($_POST['categorie']))
{
    include("formulaire_recherche.php");
}
// Pour une recherche sans mot_cle
elseif (isset($_POST['mot_cle1']) AND isset($_POST['mot_cle2']) AND isset($_POST['mot_cle3']))
{
    $reponse = $bdd->prepare('
        SELECT * FROM document WHERE categorie = ?') or die(print_r($bdd->errorInfo()));

    $reponse->execute(array(
        $_POST['categorie']));
}
```

Εικόνα 14: Παράδειγμα σύνταξης PHP

2.2 Δομές (frameworks), βιβλιοθήκες (libraries), μεταγλωττιστές (compilers), εργαλεία

Στην υποενότητα που ακολουθεί, θα γίνει μία περιεκτική αναφορά στις βιβλιοθήκες (libraries), στις δομές (frameworks), στους μεταγλωττιστές (compilers) και σε διάφορα άλλα εργαλεία που χρησιμοποιήθηκαν στην υλοποίηση της διαδικτυακής εφαρμογής. Αποτελεί ένα πολύ σημαντικό κομμάτι αναφοράς της διπλωματικής διότι, ουσιαστικά, η υλοποίηση της εφαρμογής οφείλεται κατά κόρον σε αυτά τα στοιχεία. Το

μεγαλύτερο ποσοστό, αποτελείται από τεχνολογίες “του σήμερα”, οι οποίες χρησιμοποιούνται για την δημιουργία σύγχρονων διαδικτυακών εφαρμογών και όχι μόνο.

2.2.1 Laravel και αρχιτεκτονική MVC

Το Laravel [16] αποτελεί ένα δωρεάν, ανοικτού κώδικα, framework της PHP. Ως framework [17] ορίζεται μία δομή λογισμικού, υλοποιημένη σε μία συγκεκριμένη γλώσσα προγραμματισμού και περιλαμβάνει βιβλιοθήκες καθώς και βοηθητικά προγράμματα, με σκοπό ένας προγραμματιστής να γράφει πιο γρήγορα εφαρμογές. Μερικά γνωρίσματα των frameworks είναι:

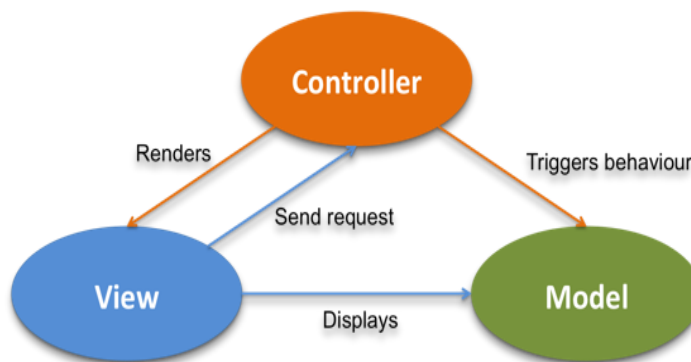
- Ενσωμάτωση συχνών χρησιμοποιούμενων «ρουτίνων» (Functions), για αξιοποίηση χρόνου και αποφυγή διπλότυπου κώδικα.
- Προβλεπόμενη και πρότυπη ακολουθία λύσεων (συνήθως) σε συχνά προβλήματα όπως (σύνδεση χρηστών, έλεγχος δεδομένων φόρμας κλπ.).
- Έτοιμες λύσεις για ασφάλεια της εφαρμογής.
- Συχνή προτροπή για καλύτερη δομή της εφαρμογής με τελικό σκοπό ο προγραμματιστής να μπορεί τελικά να την διαχειριστεί ευκολότερα όσο μεγαλώνει.
- Αποτελεί μία πιο «καλογραμμένη» έκδοση της βασικής γλώσσας και απλοποιεί σημαντικά την ανάγνωση/κατανόηση της.

Το Laravel, δημιουργήθηκε το 2011, από τον Taylor Otwell, με κύριο στόχο την δημιουργία διαδικτυακών εφαρμογών που χρησιμοποιούν την MVC αρχιτεκτονική (αναλύεται παρακάτω). Αποτελεί ίσως το πιο δημοφιλές PHP framework αυτή την στιγμή και ένας σημαντικός λόγος που η PHP παραμένει στο προσκήνιο, έναντι των άλλων γλωσσών προγραμματισμού στην δημιουργία διαδικτυακών εφαρμογών. Η σημερινή του έκδοση είναι η 5.7. Βασικά πλεονεκτήματα που προσφέρει, αναφέρονται μονολεκτικά παρακάτω:

1. Εργαλείο templating engine (Blade)
2. Απλό και γρήγορο σύστημα Eloquent Model ORM (Object-Relational Mapping)
3. Γραμμή εντολών (Artisan Command Line)

4. Σύστημα Modular Packaging
5. Σύστημα Migration
6. Ενσωματωμένες βιβλιοθήκες (Carbon για διαχείριση χρόνου, Bcrypt για κρυπτογράφηση δεδομένων κ.α.)
7. Εξαιρετικές λειτουργίες ασφαλείας για (SQL injections, Cross Site Request Forgery ή CSRF, Cross-site Scripting ή XSS)
8. Δυνατότητα δημιουργίας διάφορων tests για ανάπτυξη οδηγούμενη από τον έλεγχο (Test Driven Development ή TDD)

Τέλος, ως αρχιτεκτονική MVC (Model-View-Controller) [18] ορίζεται, ένα μοντέλο - πρότυπο αρχιτεκτονικής λογισμικού το οποίο χρησιμοποιείται για την δημιουργία εφαρμογών κάθε τύπου (mobile, desktop, web). Στο μοντέλο αυτό η εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στον χρήστη από την 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. Είναι μία πολύ συχνή τεχνική πλέον στο χτίσιμο εφαρμογών, η οποία υιοθετείται από ολοένα και περισσότερα frameworks και βιβλιοθήκες. Ο τρόπος λειτουργίας περιγράφεται στην παρακάτω εικόνα.



Εικόνα 15: Τρόπος λειτουργίας MVC αρχιτεκτονικής

2.2.2 Bootstrap και responsive design

Το Bootstrap [19] είναι ένα framework HTML και CSS με επεκτάσεις Javascript και διευθετεί πολλά θέματα εμφάνισης για φόρμες, κουμπιά και άλλα. Το Bootstrap περιέχει σύνολο κανόνων CSS για διάφορα HTML στοιχεία προσδίδοντας τους μία καλύτερη εμφάνιση. Επίσης περιέχει βιβλιοθήκες Javascript για την δυναμική αλλαγή της εμφάνισης των ιστοσελίδων. Το Bootstrap είναι μία πολύ καλή επιλογή για μία καινούρια εφαρμογή διαδικτύου διότι μπορεί να δώσει μία ωραία εμφάνιση στην ιστοσελίδα χωρίς να χρειάζεται η συγγραφή καθόλου CSS. Επίσης διαθέτει έτοιμα μοτίβα κανόνων παρουσίασης και εναλλαγής της διεπαφής χρήστη (UI), που ποικίλουν ανά μέγεθος, διαστάσεις οθόνης ή είδος συσκευής. Δεδομένο που από μόνο του προσδίδει μία “ασφάλεια” στο γεγονός ότι, σε οποιαδήποτε αλλαγή διαστάσεων της εφαρμογής, η ίδια θα παραμείνει το ίδιο ελκυστική προς τον χρήστη και παράλληλα λειτουργική (το λεγόμενο responsive design). Η σημερινή του έκδοση είναι η 4.1.3.

2.2.3 jQuery

Η jQuery [20] είναι μία βιβλιοθήκη (library) της Javascript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στην πλευρά του πελάτη (client-side) της HTML και υποστηρίζει πολλαπλούς φυλλομετρητές Ιστού. Πιο συγκεκριμένα, παρέχει έτοιμες εντολές που πραγματοποιούν τις ίδιες λειτουργίες με την Javascript με την χρήση λιγότερου κώδικα. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον Τζον Ρέριγκ (John Resig). Χρησιμοποιείται σε πάνω από το 65% των 10.000 ιστοτόπων με τη μεγαλύτερη επισκεψιμότητα. Είναι προαπαιτούμενο για την λειτουργία των Javascript πακέτων του Bootstrap. Η σημερινή του έκδοση είναι η 3.3.1. Παράδειγμα εντολής jQuery παρουσιάζει η εικόνα παρακάτω.

```
$(document).ready(function() {
  changeColor();
});

function changeColor() {
  var rndColors = ["#00FF00", "#CCCCCC", "#995499", "#FFFFFF", "#FF9900"];
  var selColor = Math.floor(Math.random() * rndColors.length);
  $('div').css("background-color", rndColors[selColor]);
  setTimeout(changeColor, 1000);
}
```

Εικόνα 16: Παράδειγμα εντολής jQuery

2.2.4 React

Η React [21] (ReactJS ή React.js) είναι μία ανοικτού κώδικα (open source) βιβλιοθήκη της Javascript για τη δημιουργία διεπαφών χρήστη (UI) όσον αφορά το μπροστινό κομμάτι αλληλεπίδρασης (frontend). Είναι το στρώμα προβολής (view layer) για εφαρμογές ιστού. Δημιουργήθηκε (2013), αναπτύσσεται και συντηρείται μέχρι και σήμερα, ως επί το πλείστον, από την ομάδα προγραμματιστών του Facebook. Πρόκειται για ένα από τα πιο σύγχρονα και τεχνολογικά εξελιγμένα εργαλεία για δημιουργία σύνθετων εφαρμογών, που δεν περιορίζονται μόνο στο διαδικτυακό κομμάτι. Η σαφής λειτουργία της είναι, η δυνατότητα που προσφέρει για ταυτόχρονη αλληλεπίδραση των δεδομένων με τον χρήστη (rendering), η οποία συμβαίνει ταχύτατα, παράλληλα με την δυνατότητα για το χτίσιμο html στοιχείων εντός Javascript. Ωστόσο, το σημαντικό στοιχείο είναι ότι, διαθέτει παράπλευρα πακέτα προς εγκατάσταση, που το καθένα από αυτά αποτελεί ξεχωριστή λειτουργία. Αν συνδυαστούν, δημιουργούν ένα τελειοποιημένο εργαλείο για την δημιουργία ακόμα και της πιο σύνθετης εφαρμογής. Ένα πολύ σημαντικό στοιχείο που προκύπτει είναι το εξής: δύναται ή δημιουργία ενός διαφορετικού είδους εφαρμογής, η λεγόμενη εφαρμογή μίας σελίδας (Single Page Application ή SPA) [22], η οποία δεν αποτελεί τίποτα παραπάνω από ένα HTML έγγραφο, το οποίο περιέχει σαν εξάρτηση ένα Javascript έγγραφο, που περιέχει όλο τον απαραίτητο κώδικα. Αυτό πρακτικά οδηγεί στην δυνατότητα αποφόρτισης του εξυπηρετητή και στην μετατροπή του από ένα πολυσύνθετο σύστημα που ευθύνεται για πολλές λειτουργίες παράλληλα (αυθεντικοποίηση, φόρτωση html, ασφάλεια, αλληλεπίδραση με την βάση δεδομένων, κ.α.) σε ένα αρκετά πιο βασικό και πιο εύκολο στην δημιουργία του, το λεγόμενο

Application Programming Interface ή API [23]. Ένας από τους λόγους που γίνεται αυτή η μετάβαση είναι διότι, ο εξυπηρετητής, πλέον, έχει αρκετά λιγότερες υποχρεώσεις και παράλληλα μπορεί να εξυπηρετεί περισσότερα αιτήματα. Τέλος, αξίζει να αναφερθεί ότι υπάρχουν και βιβλιοθήκες τρίτου προσώπου (third party libraries) που συνεργάζονται ιδανικά με την React, όπως επίσης η ίδια συνεργάζεται ιδανικά με τις κυριότερες γλώσσες προγραμματισμού εξυπηρετητή καθώς και frameworks. Η σημερινή της έκδοση είναι η 16.6. Το βασικό στοιχείο της React είναι το λεγόμενο “component”, παράδειγμα του οποίου παρουσιάζει η φωτογραφία παρακάτω.

```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hello {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="Taylor" />,
  document.getElementById('hello-example')
);
```

Εικόνα 17: Παράδειγμα React Component

2.2.5 Axios

Το Axios [24], αποτελεί ένα εργαλείο που πραγματοποιεί τα αιτήματα προς τον εξυπηρετητή, το οποίο διαθέτει κάποιες σημαντικές προ εγκατεστημένες λειτουργίες παράλληλα με ορισμένες, που περιέχονται στις καινούργιες εκδόσεις της Javascript. Αντίστοιχο εργαλείο με παρόμοια λειτουργία με το Axios είναι το ευρέως γνωστό Ajax (Asynchronous Javascript and XML) ωστόσο στην σύγκρισή τους, υστερεί σε ορισμένους τομείς. Παράδειγμα εντολής axios παρουσιάζεται στην επόμενη φωτογραφία.


```
axios.post('/user', {
  firstName: 'Fred',
  lastName: 'Flintstone'
})
.then(function (response) {
  console.log(response);
})
.catch(function (error) {
  console.log(error);
});
```

Εικόνα 18: Παράδειγμα εντολής axios

2.2.6 Redux

Το Redux [25] είναι μία ανοικτού κώδικα (open source) βιβλιοθήκη της Javascript, που λειτουργεί ως διαχειριστής της κατάστασης της εφαρμογής (state management). Πρόκειται για μία τεχνολογία αρκετά σύγχρονη όπως η React, όμως αρκετά πολύπλοκη και δυσνόητη στην αρχή, όσο για τον λόγο ύπαρξής της, καθώς και για εκμάθηση και υλοποίηση, ωστόσο τρομερά σημαντική για την λειτουργία της εφαρμογής. Η λειτουργία της απαιτείται σε μεγάλου μεγέθους εφαρμογές. Στην ουσία μπορεί να παρομοιαστεί σαν ένα “καλάθι” ή σαν μία μικρού μεγέθους βάση δεδομένων, στην οποία επιλέγει να τοποθετήσει η εφαρμογή, πληροφορίες, που ενδέχεται να τις χρειαστεί άμεσα ή αρκετά συχνά. Δεν αντικαθιστά την κανονική βάση δεδομένων του συστήματος, απλώς συνυπάρχει μαζί της. Έτσι επιτυγχάνεται άμεση απόκριση της εφαρμογής, χωρίς την χρήση αιτημάτων προς τον εξυπηρετητή. Συνεργάζεται ιδανικά με την React και αντίστοιχα frontend frameworks και βιβλιοθήκες.

2.2.7 Webpack

Το Webpack [26], αποτελεί ένα εργαλείο το οποίο ενώνει πολλαπλά αρχεία (Javascript, CSS, κ.α.) σε ένα ή δύο κύρια αρχεία του ίδιου είδους (static module bundler). Είναι ίσως το κορυφαίο εργαλείο στο είδος του αυτή την στιγμή και τρομερά σημαντικό

διότι, καταφέρνει και ελαττώνει σε πολύ μεγάλο βαθμό το μέγεθος των αρχείων που ουσιαστικά είναι η εφαρμογή, αποτέλεσμα με άμεση αντανάκλαση στην ταχύτητα και απόκριση της εφαρμογής. Έρχεται προ εγκατεστημένο πλέον με έτοιμες ρυθμίσεις (pre-build) στα περισσότερα frameworks (και στο Laravel) ή έτοιμα πακέτα για δημιουργία εφαρμογών, διότι ο αριθμός των βιβλιοθηκών που χρησιμοποιούνται είναι δυσανάλογα μεγάλος για το τελικό μέγεθος της εφαρμογής που απαιτείται.

2.2.8 Babel

Το Babel [27], αποτελεί έναν μεταγλωττιστή της Javascript (compiler), που μεταφράζει τις καινούργιες εκδόσεις της γλώσσας στην γνωστή σε όλους τους browsers ES5, ώστε η διαδικτυακή εφαρμογή να τρέχει το ίδιο ομαλά σε όλους τους browsers. Αναλύθηκε προηγουμένως στην ενότητα της Javascript, το πρόβλημα που επιλύει. Είναι το κορυφαίο εργαλείο στο είδος του, ακριβώς όπως και με το webpack, έρχεται προ εγκατεστημένο παντού και πιο συγκεκριμένα, υπάρχει έτοιμη η σύνδεση του με το webpack, ώστε ο προγραμματιστής να μην χρειαστεί να επέμβει ριζικά. Αρκετά εργαλεία που αναφέρθηκαν προηγουμένως, χρησιμοποιούν καινούργιες λειτουργίες της Javascript οπότε καθιστά απαραίτητη η προσθήκη του.

2.2.8 Pusher API

Το Pusher API [28], αποτελεί μία βιβλιοθήκη τρίτου προσώπου (third-party library), που καθιστά εφικτή την αποστολή διάφορων συμβάντων (events), με σκοπό την ενημέρωση του εκάστοτε χρήστη σε πραγματικό χρόνο (Real-Time). Δημιουργεί μία μοναδική εμπειρία χρήσης (UI experience) καθώς ο χρήστης θα λάβει ειδοποίηση για κάποιο συμβάν χωρίς την ανανέωση της εφαρμογής. Τα events μπορεί να είναι ενέργειες που πραγματοποιούνται στον εξυπηρετητή και συνήθως τις προκαλούν οι διαχειριστές της εφαρμογής, είτε για να στείλουν μία γενική ενημέρωση, είτε για κάποια επεξεργασία ενός χρήστη κτλ.

2.2.9 Sass

Η Sass [29] (Syntactically Awesome StyleSheets) αποτελεί μία προσθήκη της CSS, η οποία προσθέτει δύναμη και πολλές δυνατότητες στον προγραμματιστή όσον αφορά, την μορφοποίηση των HTML εγγράφων με CSS. Αποτελεί κομμάτι του Laravel και δέχεται επεξεργασία μέσω του webpack. Παράδειγμα κώδικα εμφανίζει η φωτογραφία παρακάτω.

```
// Fonts
@import url('https://fonts.googleapis.com/css?family=Montserrat:400,700,200');

$fa-font-path:    "../webfonts";
@import "~@fortawesome/fontawesome-free/scss/fontawesome.scss";
@import "~@fortawesome/fontawesome-free/scss/solid.scss";
```

Εικόνα 19: Παράδειγμα κώδικα Sass

2.3 Αρχιτεκτονική Arduino

Το Arduino [30] είναι μία υπολογιστική πλατφόρμα ανοικτού κώδικα (open-source hardware) η οποία είναι μικρή σε μέγεθος και βασίζεται στην τεχνολογία των μικρο-ελεγκτών (microcontrollers) [31]. Διαθέτει αναλογικές και ψηφιακές εισόδους/εξόδους (analog and digital input/output) για την λήψη δεδομένων από το πραγματικό περιβάλλον (sensors) και αντιδρά μέσω των συσκευών ανάδρασης (actuators). Η αρχιτεκτονική-σχεδιασμός του Arduino είναι ανοικτού κώδικα, κάτι που σημαίνει ότι ο καθένας μπορεί να έχει πρόσβαση στα σχέδια του Arduino ώστε να το ανακατασκευάσει. Το Arduino χρησιμοποιεί κατά βάση του μικρο-ελεγκτές ATMEΛ, με κάποια καινούρια μοντέλα να χρησιμοποιούν και συνδυασμό επεξεργαστών ARM. Επίσης, ο όρος του Arduino περιλαμβάνει και το λογισμικό κομμάτι, δηλαδή το περιβάλλον ανάπτυξης κώδικα για το Arduino το οποίο ονομάζεται Arduino IDE. Το Arduino IDE είναι και αυτό opensource. Η γλώσσα προγραμματισμού του Arduino είναι η C και η C++. Το Arduino διατίθεται σε αρκετές παραλλαγές. Η καθεμία, ωστόσο, έχει τις δικές της ιδιαιτερότητες και διαφοροποιήσεις από τις άλλες. Οι διαφορές εμφανίζονται στις επιδόσεις του κάθε


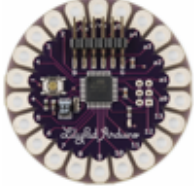



συστήματος, στο πλήθος των εισόδων/εξόδων που διαθέτουν και σε χαρακτηριστικά που κάποιες έχουν και κάποιες άλλες όχι. Τέλος, σε κάποια συστήματα Arduino μπορούμε να συνδέσουμε πλακέτες οι οποίες επεκτείνουν τις δυνατότητες του εκάστοτε συστήματος Arduino, γνωστά και ως Shields.

2.3.1 Είδη Arduino

Ξεκινώντας από τις μικρότερες σε μέγεθος και δυνατότητες εκδόσεις Arduino, υπάρχουν τα Arduino Nano και LilyPad. Ακολουθεί το Arduino Uno, το οποίο είναι η δημοφιλέστερη και η πιο συχνά χρησιμοποιούμενη έκδοση του Arduino σε όλα τα συστήματα. Υπάρχουν ορισμένες διαφοροποιήσεις της πλακέτας Uno όσον αφορά τις δυνατότητες που προσφέρει. Ειδικότερα, υπάρχει ο ενσωματωμένος συνδυασμός υλοποίησης της πλακέτας με εξαρτήματα, που πραγματοποιούν ορισμένες λειτουργίες (σύνδεση στο διαδίκτυο ενσύρματα ή ασύρματα κ.τ.λ.) Το Arduino Ethernet πρόκειται για μία τέτοια συσκευή Arduino Uno. Σε αυτό έχει ενσωματωθεί το Ethernet Shield με αποτέλεσμα να συνυπάρχουν σε μία κοινή μητρική πλακέτα. Έπειτα υπάρχουν παράλληλα το Arduino Mega με το Arduino Due (λειτουργεί στα 3.3V), τα οποία είναι στην ουσία ένα Arduino Uno στο οποίο έχουν αυξηθεί αρκετά οι επιδόσεις, οι μνήμες και οι εισοδοί/εξοδοί. Ενδεικτικά υπάρχουν εκδόσεις Arduino, οι οποίες εκτός του βασικού Atmel μικρο-ελεγκτή είναι εξοπλισμένες και με άλλους επεξεργαστές, όπως τους ARM που αναφέρθηκαν προηγουμένως (παράδειγμα Arduino Due). Τέτοιες εκδόσεις μπορούν να τρέξουν ακόμα και το λειτουργικό σύστημα Linux.

Στα πλαίσια της διπλωματικής εργασίας, για τις ανάγκες του ενσωματωμένου συστήματος, χρησιμοποιήθηκε η πλακέτα Arduino Mega 2560, η οποία αποτελεί ίσως η δυνατότερη από τις πλακέτες Arduino και ο λόγος που επιλέχτηκε, είναι ο μεγάλος αριθμός των αισθητήρων παράλληλα με ορισμένα εξαρτήματα, καθώς και το Ethernet shield που επιλέχτηκε (αναλύεται παρακάτω). Ο συνδυασμός όλων αυτών, καθώς και των βιβλιοθηκών που χρησιμοποιούν, ήταν απαγορευτικός για την Uno πλακέτα ως προς την χωρητικότητα, κυρίως.

Παρακάτω παρουσιάζεται ένας πίνακας με ενδεικτικές φωτογραφίες των πλακετών.

Arduino Nano	
Arduino LilyPad	
Arduino Uno	
Arduino Due	
Arduino Mega	

Πίνακας 2: Είδη Arduino

2.3.2 Arduino Shields

Τα Arduino Shields είναι πλακέτες οι οποίες περιέχουν υλικό που επεκτείνει τις δυνατότητες του βασικού συστήματος. Ένα δημοφιλές παράδειγμα επέκτασης είναι η προσθήκη μία πλακέτας Ethernet Shield, η οποία δίνει την δυνατότητα στο Arduino να αποκτήσει πρόσβαση στο διαδίκτυο ενσύρματα. Παρακάτω παρουσιάζεται μία εικόνα του συγκεκριμένου Ethernet shield.



Εικόνα 20: Ethernet Shield

2.4 Πλατφόρμες Ανάπτυξης Λογισμικού

Στα πλαίσια της υλοποίησης της παρούσας διπλωματικής εργασίας ήταν απαραίτητη και η ανάπτυξη λογισμικού. Χρησιμοποιήθηκαν, επομένως, εργαλεία λογισμικού που διευκόλυναν την συγγραφή αυτού. Μία βασική ενότητα λογισμικών ανάπτυξης είναι το ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment ή IDE). Το IDE είναι πρόγραμμα λογισμικού. Διαθέτει στον προγραμματιστή ένα περιβάλλον στο οποίο παρέχονται συγκεντρωμένες δυνατότητες οι οποίες είναι απαραίτητες για την συγγραφή λογισμικού. Τα βασικά μέρη ενός IDE είναι ο επεξεργαστής πηγαίου κώδικα, εργαλεία για την αυτοματοποιημένη μεταγλώττιση και αποσφαλματωτές. Τα περισσότερα σύγχρονα IDE διαθέτουν επίσης μηχανισμούς για την αυτόματη συμπλήρωση εντολών κατά την συγγραφή κώδικα, μηχανισμούς για την εύρεση λανθασμένης σύνταξης, περιηγητές κλάσεων και αντικειμένων ακόμα και σύστημα διαχείρισης εκδόσεων (Version Control System ή VCS).

2.4.1 PHPStorm

Το PHPStorm είναι ένα σύγχρονο IDE το οποίο περιέχει όλα τα χαρακτηριστικά που προαναφέρθηκαν. Εξειδικεύεται σε συστήματα τα οποία χρησιμοποιούν την php, ως γλώσσα προγραμματισμού καθώς και διάφορα frameworks της PHP. Το PHPStorm δεν είναι ένα λογισμικό δωρεάν και ανοικτού κώδικα. Ωστόσο, υπάρχουν εκπτώσεις σε ορισμένες κατηγορίες και παρέχεται δωρεάν σε μαθητές, φοιτητές, καθηγητές και ομάδες που κατασκευάζουν λογισμικά ανοικτού κώδικα. Τέλος, να τονιστεί πως παρόλο που το PHPStorm, χρησιμοποιείται κατ' ονομασία για την PHP, ωστόσο διαθέτει ιδανική υποστήριξη για frontend libraries όπως React και Vuejs καθώς και άλλων εργαλείων και αυτός είναι και ο κυριότερος λόγος για τον οποίο επιλέχτηκε.

2.4.2 Visual Studio Code

Το Visual Studio Code (VSCode) είναι ένα σύγχρονο IDE, υλοποιημένο από την Microsoft, ανοικτού κώδικα, δωρεάν και με τρομερή απήχηση μεταξύ των προγραμματιστών. Πρόκειται για ένα από τα ελάχιστα δωρεάν IDE's με την ποιότητα και τις λειτουργίες που προσφέρει. Επίσης διαθέτει στο διαδίκτυο μία τεράστια ποικιλία από εξωτερικά εργαλεία και συνδέσεις (plugins). Χρησιμοποιήθηκε στο πρώιμο στάδιο της υλοποίησης της διαδικτυακής εφαρμογής.

2.4.3 Arduino IDE

Αναφέρθηκε και νωρίτερα ως το λογισμικό ανάπτυξης για τον προγραμματισμό των Arduino πλακετών, το οποίο υποστηρίζεται από την εταιρεία καθώς και την κοινότητα των Arduino προγραμματιστών.

2.5 Εργαλεία Virtualization – Έτοιμα κουτιά ανάπτυξης εφαρμογών

Στο πλαίσιο ανάπτυξης της διαδικτυακής εφαρμογής, χρησιμοποιήθηκαν ορισμένα εργαλεία virtualization και έτοιμα κουτιά ανάπτυξης εφαρμογών όπως, Laravel Homestead, VirtualBox και Vagrant. Αξίζει να αναφερθεί περιληπτικά ότι, για την υλοποίηση σύγχρονων εφαρμογών μέσω frameworks ή άλλων σύγχρονων μεθόδων, απαιτείται η προεγκατάσταση αρκετών τεχνολογιών, τις οποίες χρειάζονται τα frameworks, ώστε να επιτρέψουν την υλοποίηση. Ωστόσο αυτό μπορεί να αποβεί βαρύ φορτίο για τον προσωπικό υπολογιστή του προγραμματιστή, καθώς επίσης μπορεί να προκύψουν διαμάχες (conflicts) μεταξύ εκδόσεων διάφορων τεχνολογιών. Γι' αυτό τον σκοπό έχουν δημιουργηθεί έτοιμα τέτοια πακέτα με προ εγκατεστημένες όλες τις τεχνολογίες, που τρέχουν μέσω προγράμματος virtualization και αποτρέπουν τις προαναφερθείσες ενέργειες. Το αντίστοιχο πακέτο που χρησιμοποιεί το Laravel είναι το Laravel Homestead, το οποίο αποτελεί ένα κουτί περιβάλλοντος Vagrant με προ εγκατεστημένες όλες τις απαραίτητες τεχνολογίες που απαιτεί το Laravel (Git, Nginx, PHP7.1-7.3, MySQL, SQLite3, PostgreSQL, Composer, Node, Redis κ.α.).

2.6 Λογισμικά σχεδίασης

Τα λογισμικά αυτά βοηθούν στην δημιουργία σχηματικών διαγραμμάτων και επεξεργασία των εικόνων.

2.6.1 Fritzing

Το Fritzing είναι ένα πρόγραμμα λογισμικού. Σε αυτό μπορούν να μεταφερθούν οι πρωτότυπες κατασκευές του Arduino από breadboard και οποιαδήποτε άλλη ηλεκτρονική κατασκευή. Τα προαναφερθέντα σχέδια χρησιμοποιούνται είτε για τον διαμοιρασμό με άλλους σχεδιαστές είτε και για την εξαγωγή ηλεκτρονικών σχεδιαγραμμάτων. Ακόμα, μπορούν να χρησιμοποιηθούν και για την δημιουργία PCB πλακέτας για την αποστολή σε

βιομηχανία δημιουργίας πλακετών. Το Fritzing χρησιμοποιήθηκε για τον σχεδιασμό της πρωτότυπης κατασκευής του Arduino και έπειτα για την εξαγωγή του ηλεκτρονικού σχεδίου.

2.7 Σύνοψη κεφαλαίου

Σε αυτό το κεφάλαιο, παρουσιάστηκαν όλες οι πληροφορίες και τα εργαλεία που χρειάστηκαν για την ανάπτυξη του ενσωματωμένου συστήματος καθώς και της διαδικτυακής εφαρμογής. Πιο συγκεκριμένα, αναλύθηκαν οι έννοιες του θεωρητικού υπόβαθρου, παρουσιάστηκαν οι γλώσσες προγραμματισμού, οι τεχνολογίες και τα εργαλεία του λογισμικού που χρησιμοποιήθηκαν. Επίσης, δόθηκαν κάποια στιγμιότυπα (screenshot) για την καλύτερη κατανόηση των εννοιών, όπως αυτές εφαρμόστηκαν στην πράξη. Στο επόμενο κεφάλαιο, παρουσιάζονται οι απαιτήσεις του συστήματος, οι λεπτομέρειες για το σχεδιασμό και την ανάπτυξη της βάσης δεδομένων της εφαρμογής, καθώς και τεχνικές που εφαρμόστηκαν για την ασφάλεια του συστήματος.

Κεφάλαιο 3ο – Δομικά μέρη συστήματος

Σε αυτό το κεφάλαιο, γίνεται παρουσίαση των απαιτήσεων του συστήματος καθώς και των δομικών μερών του. Αρχικά, γίνεται αναλυτική αναφορά στον σχεδιασμό και την υλοποίηση της βάσης δεδομένων. Επιπλέον, αναλύονται, ο τρόπος επίτευξης επικοινωνίας ενσωματωμένου συστήματος και διαδικτυακής εφαρμογής, περιγράφονται οι ρόλοι των χρηστών στην εφαρμογή και τέλος, παρουσιάζονται τεχνικές και εργαλεία που χρησιμοποιούνται για την επίτευξη της ασφάλειας του συστήματος.

3.1 Απαιτήσεις συστήματος

Για την επιτυχή ολοκλήρωση κάθε έργου, ένας πολύ σημαντικός παράγοντας είναι ο σωστός καθορισμός των απαιτήσεων από την αρχική περιγραφή του, μιας και αυτές καθορίζουν την πορεία του έργου. Συγκεκριμένα, για το παρόν έργο, στόχος ήταν η δημιουργία ενός ενσωματωμένου συστήματος, παράλληλα με μία διαδικτυακή εφαρμογή και ταυτόχρονα, η ακριβής καταγραφή και αποστολή μετρήσεων μεταξύ των δύο. Παρακάτω παρατίθενται αναλυτικά οι απαιτήσεις που εξήχθησαν συνοδευόμενες από επεξήγηση.

- **Το σύστημα παρέχει λειτουργία εγγραφής-επιβεβαίωσης χρηστών καθώς και ανάκτησης κωδικού πρόσβασης.** Για την χρήση του συστήματος απαραίτητη προϋπόθεση αποτελεί, η δημιουργία λογαριασμού χρήστη σε αυτό. Με την δημιουργία του λογαριασμού, παράλληλα αποστέλλεται email επιβεβαίωσης, στην διεύθυνση αλληλογραφίας (email) που δήλωσε ο χρήστης κατά την εγγραφή του, ώστε να γίνει επιτυχής ταυτοποίηση προσώπου. Ο χρήστης θεωρείται ενεργός και έχει πρόσβαση στην εφαρμογή, αφού πρώτα επιβεβαιώσει τον λογαριασμό του. Επίσης, δίνεται η δυνατότητα της ανάκτησης κωδικού σε περίπτωση απώλειας του από τον χρήστη.
- **Το σύστημα διαχειρίζεται τρεις κατηγορίες χρηστών.** Αυτές αποτελούνται από: τους απλούς χρήστες, τους διαχειριστές και τον δημιουργό. Οι απλοί χρήστες έχουν

δυνατότητες δημιουργίας και επεξεργασίας προσωπικών σταθμών, συλλογής μετρήσεων από τους προσωπικούς τους σταθμούς, δυνατότητα προβολής και σύγκρισης μετρήσεων μεταξύ των σταθμών τους καθώς και σταθμών άλλων χρηστών, που έχουν επιλεγεί ως δημόσιοι και ενεργοί και τέλος, έχουν δυνατότητα επεξεργασίας του λογαριασμού τους και των στοιχείων του. Οι διαχειριστές αποτελούν στην ουσία χρήστες, με σαφώς αυξημένες δυνατότητες επεξεργασίας, δημιουργίας και ελέγχου, καθώς τους δίνονται οι δυνατότητες για δημιουργία κατηγοριών που επιλέγονται στις μετρήσεις, επεξεργασίας σταθμών χρηστών ή άλλων διαχειριστών, καθώς και αλλαγή ιδιοκτησίας αυτών, αναβάθμιση, υποβάθμιση, διαγραφή και δημιουργία χρηστών και διαχειριστών αντίστοιχα και τέλος, προβολή και σύγκριση μετρήσεων μεταξύ των σταθμών όλων των χρηστών. Διαχειριστής δημιουργείται ή αναβαθμίζεται από τον δημιουργό ή από άλλους διαχειριστές. Τέλος ο δημιουργός, αποτελεί τον διαχειριστή με όλες τις δυνατότητες που του προσφέρει η εφαρμογή. Είναι μοναδικός και κανένας δεν έχει δικαιοδοσία επεξεργασίας αυτού ή των σταθμών του.

- **Το σύστημα διαχειρίζεται την καταχώρηση συσκευών.** Το βασικό κομμάτι αυτού του συστήματος είναι οι συσκευές του (συνώνυμη ονομασία: σταθμοί ή ενσωματωμένα συστήματα). Έτσι, πρέπει να υπάρχει ο κατάλληλος μηχανισμός για την επιτυχή σύζευξη και ταυτοποίηση του συστήματος με τις συσκευές.
- **Ειδοποίηση και ενημέρωση χρηστών.** Κάθε χρήστης του συστήματος δέχεται ενημέρωση για διάφορες λειτουργίες και ενέργειες σε πραγματικό χρόνο όπως η συλλογή νέων μετρήσεων, η ανάθεση σταθμού από διαχειριστή, η δημιουργία καινούργιας κατηγορίας μετρήσεων και ούτω καθεξής. Οι αλλαγές στο UI γίνονται ασύγχρονα, γεγονός που βοηθάει στην απόκριση και ποιότητα της εφαρμογής, καθώς ο χρήστης δεν απαιτείται να ανανεώνει χειροκίνητα την σελίδα, για κάθε αλλαγή που συμβαίνει.
- **Απλό και φιλικό περιβάλλον χρήσης καθώς και αποκριτική σχεδίαση της εφαρμογής (UI friendly and responsive design).** Το περιβάλλον χρήσης της εφαρμογής είναι δομημένο και υλοποιημένο με τέτοιο τρόπο, ώστε να γίνεται

κατανοητό και εύκολα διαχειρίσιμο, ακόμα και από άτομα με ελάχιστες γνώσεις ή εμπειρία στην χρήση υπολογιστή. Η απλότητά του έχει ως αποτέλεσμα όχι μόνο την ευκολία στην χρήση, αλλά και να καθοδηγεί τον χρήστη στην εύρεση των λειτουργιών που αναζητεί. Τέλος η αποκριτική του σχεδίαση επιτρέπει την διατήρηση της ποιότητας, της εμφάνισης και των λειτουργιών της εφαρμογής, από οποιαδήποτε συσκευή χρησιμοποιείται, οποιωνδήποτε διαστάσεων (κινητό τηλέφωνο, tablet ή Η/Υ).

- **Απαιτείται πρόσβαση στο διαδίκτυο, για αποστολή μετρήσεων μεταξύ διαδικτυακής εφαρμογής και σταθμών.** Η επικοινωνία και η μεταφορά μετρήσεων μεταξύ σταθμών και εφαρμογής του συστήματος επιτυγχάνεται μόνο μέσω διαδικτύου (ασύρματα ή ενσύρματα), δηλαδή με την χρήση του TCP/IP μοντέλου επικοινωνίας.
- **Κρίνεται αναγκαία η εφαρμογή της ασφάλειας σε κάθε πτυχή του συστήματος.** Αυτό σημαίνει πως η επικοινωνία μεταξύ της εφαρμογής και των σταθμών καθώς και του χρήστη με την διαδικτυακή εφαρμογή, υποστηρίζει πολλές μορφές ασφαλείας, κάθε μορφή υπεύθυνη για τον τομέα ευθύνης της.
- **Επεκτάσιμο και συντηρήσιμο.** Οι έννοιες αυτές ορίζουν πως το σύστημα θα πρέπει να κατασκευαστεί, είτε σε επίπεδο κώδικα είτε σε επίπεδο υλικής κατασκευής, με τρόπο εύκολα κατανοητό από κάποιον τρίτο. Στόχος αυτού είναι ο κάθε εμπλεκόμενος με το σύστημα να μπορεί εύκολα να κατανοεί τη ροή των γεγονότων και να επεμβαίνει όπου χρειάζεται. Τέλος πρέπει να τονιστεί πως μία από τις σημαντικές αρχές για την δημιουργία επεκτάσιμου και συντηρήσιμου κώδικα είναι η αποφυγή της επανάληψης κομματιών κώδικα.

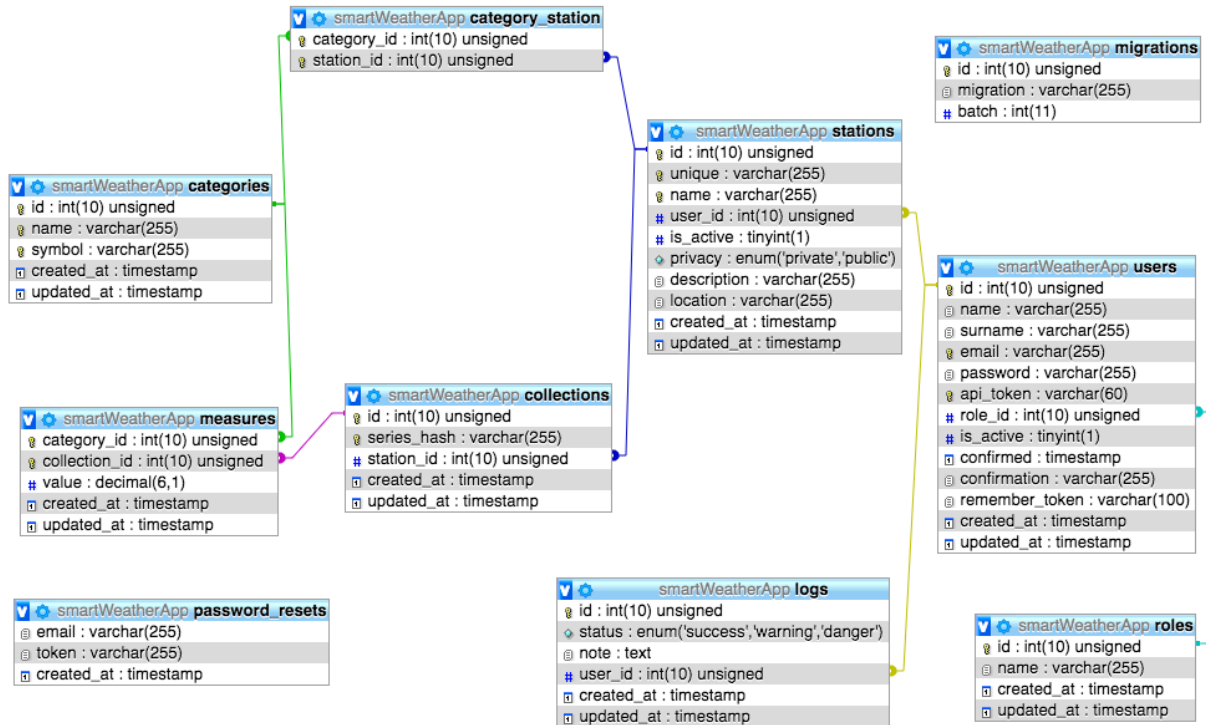
3.2 Ανάλυση βάσης δεδομένων

Μετά την ανάλυση και τον προσδιορισμό των απαιτήσεων του συστήματος, το επόμενο βήμα για τον προγραμματιστή είναι ο σχεδιασμός της βάσης δεδομένων. Ο κατά το δυνατόν καλύτερος σχεδιασμός της βάσης δεδομένων αποτελεί υψίστης σημασίας

εργασία και χτίζεται με έμφαση στον τρόπο που οι πίνακες συνδέονται μεταξύ τους, με σχέσεις ανάμεσα στις εγγραφές τους. Ο βέλτιστος σχεδιασμός μπορεί να επιτευχθεί με την κανονικοποίηση (normalization) της βάσης δεδομένων, χάρη στην οποία αφαιρείται η πλεονάζουσα πληροφορία, αυξάνεται η σαφήνεια της οργάνωσης της βάσης και απλοποιείται η διαχείρισή της. Επιπλέον, η κανονικοποίηση της βάσης δεδομένων επιτρέπει την ευκολότερη και αποδοτικότερη τροποποίησή της στο μέλλον, έτσι ώστε να καλύπτει τις προδιαγραφές μίας συνεχώς αναβαθμιζόμενης και λειτουργικά επεκτεινόμενης εφαρμογής. Παρακάτω παρατίθεται αρχικά, το διάγραμμα οντοτήτων συσχετίσεων, ώστε ο αναγνώστης να λάβει μία πλήρη εικόνα για το πως είναι δομημένη η βάση με τους πίνακες και τι συσχετίσεις υπάρχουν μεταξύ των πινάκων και έπειτα, ακολουθεί η παρουσίαση κάθε πίνακα ξεχωριστά, μαζί με μία αναλυτική περιγραφή για κάθε στήλη, κάτω από κάθε πίνακα, για να αποσαφηνιστεί ο λόγος ύπαρξής της. Τέλος αξίζει να αναφερθεί ότι, αρκετοί πίνακες και συσχετίσεις πινάκων, δημιουργήθηκαν ακολουθώντας τα πρότυπα (patterns) που ορίζει και προτείνει το Laravel framework, ως προς την δημιουργία μίας βάσης δεδομένων, πρότυπα που εξαλείφουν και αποτρέπουν δημιουργία διπλότυπων κελιών, βελτιστοποιούν την ποιότητα σχεδίασης με χρήση κανονικοποίησης ανάλογα με τις περιπτώσεις χρήσης και τέλος, συνδυάζονται αρμονικά με εντολές και συναρτήσεις του Laravel, οι οποίες βοηθούν σημαντικά στον προγραμματιστή, ώστε να παράξει λειτουργικά σωστό και ποιοτικό κώδικα.

3.2.1 Σχεσιακό διάγραμμα βάσης δεδομένων

Στην εικόνα που ακολουθεί, αποτυπώνεται μία ολοκληρωμένη αναπαράσταση της δομής της βάσης δεδομένων. Πιο συγκεκριμένα, φαίνονται όλοι οι πίνακες με τα πεδία τους, τους τύπους δεδομένων και τις εξαρτήσεις τους.



Εικόνα 21: Σχεσιακό διάγραμμα βάσης δεδομένων SmartWeatherApp

3.2.2 Δομή πίνακα “users”

Στον πίνακα που ακολουθεί, αποτυπώνεται η δομή του πίνακα “users” στον οποίο καταχωρούνται στοιχεία σχετικά με τους χρήστες που έχουν εγγραφεί στην διαδικτυακή εφαρμογή του συστήματος.

Όνομα	Τύπος	Χαρακτηριστικά	Προεπιλογή	Πρόσθετα
id	int (10)	unsigned, primary key	none	auto_increment
name	varchar(255)		none	
surname	varchar(255)		none	
email	varchar(255)	unique	none	
password	varchar(255)		none	
api_token	varchar(60)	unique, nullable	null	
role_id	int(10)	unsigned, index, nullable	null	
is_active	tinyint(1)		0	
confirmed	timestamp	nullable	null	
confirmation	varchar(255)	nullable	null	
remember_token	varchar(100)	nullable	null	
created_at	timestamp	nullable	null	
updated_at	timestamp	nullable	null	

Πίνακας 3: Πίνακας "users"

Ανάλυση στηλών του πίνακα "users":

id: Το πρωταρχικό κλειδί του πίνακα. Πρόκειται για πεδίο αυτόματης αρίθμησης, το οποίο χρησιμοποιείται για να αναγνωρισθεί μοναδικά ο χρήστης στην εφαρμογή. Χρησιμοποιείται ως συνδετικό κλειδί σε άλλους πίνακες. Δεν δέχεται αρνητικούς χαρακτήρες.

name: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Χρησιμοποιείται για την αποθήκευση του μικρού ονόματος του χρήστη.

surname: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Χρησιμοποιείται για την αποθήκευση του επωνύμου του χρήστη.

email: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Χρησιμοποιείται για την διεύθυνση της ηλεκτρονικής αλληλογραφίας (email) του χρήστη. Κάθε στοιχείο που εισάγεται σε αυτό το πεδίο είναι μοναδικό (unique).

password: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Χρησιμοποιείται για την αποθήκευση του κρυπτογραφημένου από τον αλγόριθμο CRYPT_BLOWFISH [34] κωδικού πρόσβασης του χρήστη.

api_token: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 60 χαρακτήρων. Αποτελεί ένα από τα πιο σημαντικά στοιχεία ολόκληρης της εφαρμογής. Στην ουσία είναι ένα πεδίο το οποίο αρχικά έχει μηδενική τιμή (nullable). Με την επιτυχημένη σύνδεση του χρήστη στην εφαρμογή, ανανεώνεται σε μία παράσταση 60 τυχαίων χαρακτήρων. Στην συνέχεια με την χρήση αυτού του πεδίου, ο χρήστης έχει πρόσβαση στο api της εφαρμογής (δυνατότητα για αιτήματα (requests) στον server και στην βάση δεδομένων). Η ταυτοποίησή του γίνεται μέσω αυτού του πεδίου. Κατά την αποσύνδεση του χρήστη, το πεδίο ανανεώνεται σε μηδενική τιμή. Κάθε πεδίο είναι μοναδικό ανά χρήστη.

role_id: Αριθμητικό πεδίο, μέγιστης χωρητικότητας 10 χαρακτήρων. Αποτελεί συσχέτιση με τον πίνακα "roles" ο οποίος δείχνει τις κατηγορίες χρηστών (απλός χρήστης-διαχειριστής). Το πεδίο δέχεται μηδενική τιμή αλλά όχι αρνητικές τιμές.

is_active: Αριθμητικό πεδίο δύο πιθανών τιμών (boolean). Χρησιμοποιείται για την δήλωση του χρήστη ως ενεργού ή ανενεργού.

confirmed: Αποτελεί στιγμιότυπο χρονικής στιγμής (timestamp). Χρησιμοποιείται για την επαλήθευση της επιβεβαίωσης του χρήστη από την εφαρμογή. Δέχεται μηδενική τιμή.

confirmation: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Εδώ αποθηκεύεται ένας τυχαίος κωδικός που περιέχεται στο email επιβεβαίωσης που αποστέλλεται στον χρήστη, κατά την εγγραφή του στην εφαρμογή, μέσω του οποίου γίνεται ταυτοποίηση και εν συνεχεία επιβεβαίωση. Δέχεται μηδενική τιμή.

remember_token: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 100 χαρακτήρων. Δημιουργείται αυτόματα από το Laravel. Χρησιμοποιείται για την επαλήθευση και την επανασύνδεση του χρήστη, μετά την απομάκρυνση του από τον browser. Δέχεται μηδενική τιμή.

created_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που δημιουργήθηκε ο χρήστης. Δέχεται μηδενική τιμή.

updated_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που πραγματοποιήθηκε αλλαγή σε κάποιο πεδίο του χρήστη. Δέχεται μηδενική τιμή.

3.2.3 Δομή πίνακα “roles”

Στον πίνακα που ακολουθεί, αποτυπώνεται η δομή του πίνακα “roles” στον οποίο καταχωρούνται στοιχεία σχετικά με τις κατηγορίες των χρηστών. Η δημιουργία των κατηγοριών, έχει προγραμματιστεί να γίνεται αυτόματα, στο χτίσιμο των αρχείων της εφαρμογής (production build) και δεν τροποποιείται.

Όνομα	Τύπος	Χαρακτηριστικά	Προεπιλογή	Πρόσθετα
id	int (10)	unsigned, primary_key	none	auto_increment
name	varchar(255)		none	
created_at	timestamp	nullable	null	
updated_at	timestamp	nullable	null	

Πίνακας 4: Πίνακας "roles"

Ανάλυση στηλών του πίνακα "roles":

id: Το πρωταρχικό κλειδί του πίνακα. Πρόκειται για πεδίο αυτόματης αρίθμησης, το οποίο χρησιμοποιείται για να αναγνωριστεί μοναδικά η κατηγορία στην εφαρμογή.

Χρησιμοποιείται ως συνδετικό κλειδί σε άλλους πίνακες. Δεν δέχεται αρνητικούς χαρακτήρες.

name: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Χρησιμοποιείται για την αποθήκευση της ονομασίας που δίνεται στην κατηγορία (admin, user κτλ.).

created_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που δημιουργήθηκε η κατηγορία. Δέχεται μηδενική τιμή.

updated_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που πραγματοποιήθηκε αλλαγή σε κάποιο πεδίο της κατηγορίας. Δέχεται μηδενική τιμή.

3.2.4 Δομή πίνακα “stations”

Στον πίνακα που ακολουθεί, αποτυπώνεται η δομή του πίνακα “stations” στον οποίο καταχωρούνται οι σταθμοί, που δημιουργούν οι χρήστες της εφαρμογής. Αξίζει να σημειωθεί πως έχει διατυπωθεί προγραμματιστικά, ότι, σε περίπτωση διαγραφής χρήστη που υπάρχει σαν συσχέτιση σε πεδίο, διαγράφεται ολόκληρη η καταχώρηση του συγκεκριμένου πεδίου (on delete cascade).

Όνομα	Τύπος	Χαρακτηριστικά	Προεπιλογή	Πρόσθετα
id	int (10)	unsigned, primary_key	none	auto_increment
unique	varchar(255)	unique	none	
name	varchar(255)	unique	none	
user_id	varchar(255)	unsigned, index	none	
is_active	tinyint(1)		0	
privacy	enum		private	
description	varchar(255)	nullable	null	
location	varchar(100)	nullable	null	
created_at	timestamp	nullable	null	
updated_at	timestamp	nullable	null	

Πίνακας 5: Πίνακας "stations"

Ανάλυση στηλών του πίνακα "stations":

id: Το πρωταρχικό κλειδί του πίνακα. Πρόκειται για πεδίο αυτόματης αρίθμησης, το οποίο χρησιμοποιείται για να αναγνωριστεί μοναδικά ο σταθμός στην εφαρμογή. Χρησιμοποιείται ως συνδετικό κλειδί σε άλλους πίνακες. Δεν δέχεται αρνητικούς χαρακτήρες.

unique: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Χρησιμοποιείται για την αποθήκευση του μοναδικού κωδικού του σταθμού. Με αυτό τον κωδικό γίνεται ταυτοποίηση του σταθμού από την εφαρμογή, κατά την διαδικασία αποστολής μετρήσεων. Κάθε πεδίο έχει μοναδικό κωδικό. Προγραμματιστικά ελέγχεται ώστε να μην παίρνει ελληνικούς χαρακτήρες.

name: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Χρησιμοποιείται για την αποθήκευση του ονόματος που δίνεται στον σταθμό. Κάθε πεδίο έχει μοναδικό όνομα.

user_id: Αριθμητικό πεδίο, μέγιστης χωρητικότητας 10 χαρακτήρων. Αποτελεί συσχέτιση με τον πίνακα "users" ο οποίος δείχνει τους χρήστες. Ουσιαστικά δείχνει τον ιδιοκτήτη του σταθμού. Το πεδίο δεν δέχεται αρνητικές τιμές.

is_active: Αριθμητικό πεδίο δύο πιθανών τιμών (boolean). Χρησιμοποιείται για την δήλωση του σταθμού ως ενεργού ή ανενεργού.

privacy: Πεδίο απαρίθμησης. Εδώ επιλέγεται η ιδιωτικότητα του σταθμού. Οι διαθέσιμες επιλογές που δίνονται στους χρήστες είναι public (δημόσιος) και private (ιδιωτικός).

description: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Το συγκεκριμένο πεδίο χρησιμοποιείται για την συμπλήρωση κάποιας περιγραφής που μπορεί να δώσει ο χρήστης για τον σταθμό του. Είναι προαιρετικό για συμπλήρωση. Δέχεται μηδενική τιμή.

location: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Εδώ αποθηκεύεται η πόλη στην οποία βρίσκεται ο σταθμός. Δίνεται στον χρήστη μία λίστα με όλες τις πόλεις της Ελλάδας και επιλέγει από εκεί την απάντηση του και δεν εισάγει αυθαίρετα δική του επιλογή.

created_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που δημιουργήθηκε ο σταθμός. Δέχεται μηδενική τιμή.

updated_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που πραγματοποιήθηκε αλλαγή σε κάποιο πεδίο του σταθμού. Δέχεται μηδενική τιμή.

3.2.5 Δομή πίνακα “categories”

Στον πίνακα που ακολουθεί, αποτυπώνεται η δομή του πίνακα “categories”, στον οποίο καταχωρούνται οι διαθέσιμες κατηγορίες των μετρήσεων, που μπορούν οι χρήστες να επιλέξουν για τους σταθμούς τους. Η δημιουργία και επεξεργασία των κατηγοριών, αποτελεί δικαίωμα μόνο των διαχειριστών.

Όνομα	Τύπος	Χαρακτηριστικά	Προεπιλογή	Πρόσθετα
id	int (10)	unsigned, primary_key	none	auto_increment
name	varchar(255)	unique	none	
symbol	varchar(255)	unique	none	
created_at	timestamp	nullable	null	
updated_at	timestamp	nullable	null	

Πίνακας 6: Πίνακας "categories"

Ανάλυση στηλών του πίνακα "categories":

id: Το πρωταρχικό κλειδί του πίνακα. Πρόκειται για πεδίο αυτόματης αρίθμησης, το οποίο χρησιμοποιείται για να αναγνωριστεί μοναδικά η κατηγορία στην εφαρμογή. Χρησιμοποιείται ως συνδετικό κλειδί σε άλλους πίνακες. Δεν δέχεται αρνητικούς χαρακτήρες.

name: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Χρησιμοποιείται για την αποθήκευση της ονομασίας που δίνεται στην κατηγορία (temperature, humidity κτλ.). Προγραμματιστικά ελέγχεται ώστε να μην παίρνει ελληνικούς χαρακτήρες. Κάθε πεδίο έχει μοναδικό όνομα.

symbol: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Χρησιμοποιείται για την αποθήκευση του συμβόλου της κατηγορίας (°C για temperature κτλ.). Κάθε πεδίο έχει μοναδικό σύμβολο.

created_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που δημιουργήθηκε η κατηγορία. Δέχεται μηδενική τιμή.

updated_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που πραγματοποιήθηκε αλλαγή σε κάποιο πεδίο της κατηγορίας. Δέχεται μηδενική τιμή.

3.2.6 Δομή πίνακα “category_station”

Στον πίνακα που ακολουθεί, αποτυπώνεται η δομή του πίνακα “category_station”, στον οποίο καταχωρούνται οι συσχετίσεις μεταξύ των σταθμών και των κατηγοριών. Αποτελεί ένα από τα πρότυπα του Laravel, για περίπτωση συσχέτισης πολλών δεδομένων μεταξύ δύο οντοτήτων (many to many relationship). Στην συγκεκριμένη υλοποίηση δεν απαιτείται ξεχωριστό πρωτεύον κλειδί, αλλά επιλέγεται σαν πρωτεύον κλειδί ο συνδυασμός των συσχετίσεων. Επίσης, ελέγχεται και αποτρέπεται από το σύστημα η εισαγωγή διπλότυπου συνδυασμού συσχετίσεων, το οποίο σημαίνει ότι κάθε συνδυασμός συσχετίσεων είναι μοναδικός. Τέλος έχει διατυπωθεί προγραμματιστικά ότι, σε περίπτωση διαγραφής κατηγορίας ή σταθμού που υπάρχει σε πεδίο, διαγράφεται ολόκληρη η συσχέτιση (on delete cascade).

Όνομα	Τύπος	Χαρακτηριστικά	Προεπιλογή	Πρόσθετα
category_id	int(10)	primary_key, unsigned, index	none	
station_id	int(10)	primary_key, unsigned, index	none	

Πίνακας 7: Πίνακας "category_station"

Ανάλυση στηλών του πίνακα "category_station":

category_id: Το συγκεκριμένο πεδίο χρησιμοποιείται για την καταγραφή της συσχέτισης της κατηγορίας. Δεν δέχεται αρνητικές τιμές.

station_id: Το συγκεκριμένο πεδίο χρησιμοποιείται για την καταγραφή της συσχέτισης του σταθμού. Δεν δέχεται αρνητικές τιμές.

3.2.7 Δομή πίνακα “collections”

Στον πίνακα που ακολουθεί, αποτυπώνεται η δομή του πίνακα “collections”, στον οποίο καταχωρούνται οι συλλογές μετρήσεων. Σαν συλλογή μετρήσεων ορίζεται μία ομάδα μετρήσεων, που πάρθηκε από έναν σταθμό σε μία χρονική στιγμή (πχ. μέτρηση για

temperature, rain, dust από x σταθμό σε χρονική στιγμή y). Αυτές οι μετρήσεις αφού αποθηκευτούν στην βάση δεδομένων δημιουργούν μία συσχέτιση (την συλλογή μετρήσεων), η οποία "δείχνει" σε αυτή την ομάδα μετρήσεων. Με αυτή την προσέγγιση γίνεται ευκολότερη η αναζήτηση και η σωστή και τμηματική καταχώρηση των μετρήσεων. Τέλος έχει διατυπωθεί προγραμματιστικά ότι, σε περίπτωση διαγραφής σταθμού που υπάρχει σε πεδίο, διαγράφεται ολόκληρη η καταχώρηση του συγκεκριμένου πεδίου (on delete cascade).

Όνομα	Τύπος	Χαρακτηριστικά	Προεπιλογή	Πρόσθετα
id	int (10)	unsigned, primary_key	none	auto_increment
series_hash	varchar(255)	unique	none	
station_id	int(10)	unsigned, index	none	
created_at	timestamp	nullable	null	
updated_at	timestamp	nullable	null	

Πίνακας 8: Πίνακας "collections"

Ανάλυση στηλών του πίνακα "collections":

id: Το πρωταρχικό κλειδί του πίνακα. Πρόκειται για πεδίο αυτόματης αρίθμησης, το οποίο χρησιμοποιείται για να αναγνωριστεί μοναδικά η συλλογή στην εφαρμογή. Χρησιμοποιείται ως συνδετικό κλειδί σε άλλους πίνακες. Δεν δέχεται αρνητικούς χαρακτήρες.

series_hash: Πεδίο χαρακτήρων μεταβλητού μήκους, μέγιστης χωρητικότητας 255 χαρακτήρων. Χρησιμοποιείται για την αποθήκευση μίας παράστασης 20 τυχαίων χαρακτήρων. Στην ουσία αποτελεί το όνομα της συλλογής. Δημιουργείται αυτόματα και κάθε πεδίο έχει μοναδικό κωδικό.

station_id: Αριθμητικό πεδίο, μέγιστης χωρητικότητας 10 χαρακτήρων. Αποτελεί συσχέτιση με τον πίνακα "stations" ο οποίος δείχνει τους σταθμούς. Ουσιαστικά δείχνει τον σταθμό που πραγματοποίησε την μέτρηση. Το πεδίο δεν δέχεται αρνητικές τιμές.

created_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που δημιουργήθηκε η συλλογή. Δέχεται μηδενική τιμή.

updated_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που πραγματοποιήθηκε αλλαγή σε κάποιο πεδίο της συλλογής. Δέχεται μηδενική τιμή.

3.2.8 Δομή πίνακα “logs”

Στον πίνακα που ακολουθεί, αποτυπώνεται η δομή του πίνακα “logs”, στον οποίο καταχωρούνται σαν αρχείο, οι ενέργειες που πραγματοποιούνται στην εφαρμογή, σε μορφή περιγραφής (πχ η δημιουργία ενός σταθμού ή μίας κατηγορίας, η καταγραφή καινούργιας συλλογής μετρήσεων κτλ.). Αποτελεί αυτοματοποιημένη διαδικασία.

Όνομα	Τύπος	Χαρακτηριστικά	Προεπιλογή	Πρόσθετα
id	int (10)	unsigned, primary_key	none	auto_increment
status	enum		success	
note	text		none	
user_id	int(10)	unsigned, index	none	
created_at	timestamp	nullable	null	
updated_at	timestamp	nullable	null	

Πίνακας 9: Πίνακας "logs"

Ανάλυση στηλών του πίνακα "logs":

id: Το πρωταρχικό κλειδί του πίνακα. Πρόκειται για πεδίο αυτόματης αρίθμησης, το οποίο χρησιμοποιείται για να αναγνωριστεί μοναδικά η καταγραφή στην εφαρμογή. Δεν δέχεται αρνητικούς χαρακτήρες.

status: Πεδίο απαρίθμησης. Εδώ επιλέγεται το είδος της καταγραφής ανάμεσα σε καταγραφή επιτυχημένου τύπου (success), ειδοποίησης (warning) ή κινδύνου (danger).

note: Πεδίο κειμένου. Εδώ δίνεται εν συντομία, η περιγραφή της ενέργειας που έχει συμβεί.

user_id: Αριθμητικό πεδίο, μέγιστης χωρητικότητας 10 χαρακτήρων. Αποτελεί συσχέτιση με τον πίνακα "users" ο οποίος δείχνει τους χρήστες. Ουσιαστικά δείχνει τον χρήστη που πραγματοποίησε την συγκεκριμένη ενέργεια. Το πεδίο δεν δέχεται αρνητικές τιμές.

created_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που δημιουργήθηκε η καταγραφή. Δέχεται μηδενική τιμή.

updated_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που πραγματοποιήθηκε αλλαγή σε κάποιο πεδίο της καταγραφής. Δέχεται μηδενική τιμή.

3.2.9 Δομή πίνακα “measures”

Στον πίνακα που ακολουθεί, αποτυπώνεται η δομή του πίνακα “measures”, στον οποίο καταχωρούνται οι μετρήσεις που καταγράφονται και αποστέλλονται από τους σταθμούς. Στην συγκεκριμένη υλοποίηση δεν απαιτείται ξεχωριστό πρωτεύον κλειδί, αλλά επιλέγεται σαν πρωτεύον κλειδί ο συνδυασμός των συσχετίσεων. Επίσης, ελέγχεται και αποτρέπεται από το σύστημα η εισαγωγή διπλότυπου συνδυασμού συσχετίσεων, το οποίο σημαίνει ότι κάθε συνδυασμός συσχετίσεων είναι μοναδικός. Τέλος έχει διατυπωθεί προγραμματιστικά ότι, σε περίπτωση διαγραφής κατηγορίας ή συλλογής μετρήσεων που υπάρχει σε πεδίο, διαγράφεται ολόκληρη η συσχέτιση (on delete cascade).

Όνομα	Τύπος	Χαρακτηριστικά	Προεπιλογή	Πρόσθετα
category_id	int (10)	primary_key, unsigned, index	none	
collection_id	int (10)	primary_key, unsigned, index	none	
value	decimal(6,1)		none	
created_at	timestamp	nullable	null	
updated_at	timestamp	nullable	null	

Πίνακας 10: Πίνακας "measures"

Ανάλυση στηλών του πίνακα "measures":

category_id: Το συγκεκριμένο πεδίο χρησιμοποιείται για την καταγραφή της συσχέτισης της κατηγορίας. Δεν δέχεται αρνητικές τιμές.

category_id: Το συγκεκριμένο πεδίο χρησιμοποιείται για την καταγραφή της συσχέτισης της συλλογής μετρήσεων. Δεν δέχεται αρνητικές τιμές.

value: Δεκαδικό πεδίο, μέγιστης χωρητικότητας 6 ψηφίων, με δικαίωμα ενός δεκαδικού ψηφίου. Εδώ αποθηκεύεται η τιμή της κάθε μέτρησης. Έχει δοθεί εσκεμμένα μεγάλο μήκος ψηφίων, διότι, η τιμές μετρήσεων των διάφορων κατηγοριών μπορεί να διαφέρουν σημαντικά.

created_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που δημιουργήθηκε η μέτρηση. Δέχεται μηδενική τιμή.

updated_at: Αποτελεί στιγμιότυπο χρονικής στιγμής. Δημιουργείται αυτόματα από το Laravel. Αποτυπώνει την χρονική στιγμή που πραγματοποιήθηκε αλλαγή σε κάποιο πεδίο της μέτρησης. Δέχεται μηδενική τιμή.

3.2.10 Δομή πίνακα "migrations"

Ο παρακάτω πίνακας που ακολουθεί, δημιουργείται αυτόματα και χρησιμοποιείται μόνο από το Laravel. Η λειτουργία του έχει να κάνει με την καταγραφή όλων των πινάκων. Δεν επηρεάζει τον προγραμματιστή σε κάποια λειτουργία, απλώς συνυπάρχει με τους υπόλοιπους πίνακες. Για αυτό το λόγο θα αναφερθούν αλλά δεν θα αναλυθούν τα πεδία του.

Όνομα	Τύπος	Χαρακτηριστικά	Προεπιλογή	Πρόσθετα
id	int (10)	unsigned, primary_key	none	auto_increment
migration	varchar(255)		none	
batch	int(11)		none	

Πίνακας 11: Πίνακας "migrations"

3.2.11 Δομή πίνακα "password_resets"

Ο παρακάτω πίνακας που ακολουθεί, δημιουργείται αυτόματα και χρησιμοποιείται μόνο από το Laravel. Η λειτουργία του έχει να κάνει με την καταγραφή ενεργειών ανάκτησης κωδικού πρόσβασης. Ουσιαστικά λειτουργεί σαν αρχείο καταγραφής. Δεν επηρεάζει τον προγραμματιστή σε κάποια λειτουργία, απλώς συνυπάρχει με τους υπόλοιπους πίνακες. Για αυτό το λόγο θα αναφερθούν αλλά δεν θα αναλυθούν τα πεδία του.

Όνομα	Τύπος	Χαρακτηριστικά	Προεπιλογή	Πρόσθετα
email	varchar(255)	index	none	
token	varchar(255)	unique	none	
created_at	timestamp	nullable	null	

Πίνακας 12: Πίνακας "password_resets"

3.3 Ασφάλεια συστήματος

Τα τελευταία χρόνια η επίτευξη της ασφάλειας έχει γίνει άκρως απαραίτητη προϋπόθεση και δεν αποτελεί προαιρετικό χαρακτηριστικό. Ο όρος ασφάλεια καλύπτει πολλά θέματα, από την επίθεση στο σύστημα (hardware/software) από έναν hacker, ως τη

δυσλειτουργία του συστήματος. Η αναφορά σε ζητήματα ασφάλειας διαδικτυακών εφαρμογών φέρνει στο μυαλό μας εικόνες επιτιθέμενων σε ιστότοπους, οι οποίοι «αρπάζουν» στοιχεία πιστωτικών καρτών ή εκτελούν επιθέσεις άρνησης εξυπηρέτησης (denial of service). Ωστόσο, αυτό είναι μόνο ένα μέρος του συνολικότερου προβλήματος, το οποίο καλούμαστε να αντιμετωπίσουμε, όταν επιθυμούμε να προστατεύσουμε μία διαδικτυακή εφαρμογή. Οι κώδικας που αναπτύσσει ο προγραμματιστής, εκτός από τη σωστή λειτουργικότητα, θα πρέπει να είναι και ασφαλής (με κάθε έννοια του όρου). Ως εκ τούτου, θα πρέπει να χρησιμοποιούνται όλες οι βέλτιστες πρακτικές ασφαλείας και οι συνιστώμενες τεχνικές. Σε αυτό το σημείο μπαίνουν στο παιχνίδι, τα διάφορα frameworks και οι έτοιμες λειτουργίες που προσφέρουν περί ασφαλείας. Λόγω των πολλών κατηγοριών επιθέσεων που μπορεί να προκύψουν, οι οποίες ολοένα και αυξάνονται με τα χρόνια και γίνονται ακόμα πιο πολύπλοκες, ο προγραμματιστής θα πρέπει να διαθέτει και άριστες γνώσεις δικτύων, servers, πρωτοκόλλων κτλ. πράγμα το οποίο φαντάζει αρκετά δύσκολο. Επίσης λόγω ημιμάθειας, παρατηρείται συχνά το φαινόμενο, όπου ένας προγραμματιστής "νομίζει" ότι έχει λάβει όλα τα απαραίτητα μέτρα ασφαλείας αντιγράφοντας κομμάτια κώδικα που χρησιμεύουν για αυτό τον λόγο, δίχως ωστόσο να τεστάρει την εφαρμογή του δημιουργώντας στο τέλος, αρκετά κενά ασφαλείας, αφήνοντας έτσι την εφαρμογή του εκτεθειμένη. Για να εξαλειφθούν τέτοιες καταστάσεις, μία ασφαλής και σίγουρη τεχνική είναι, ο προγραμματιστής να "αφήνει" κατά κάποιον τρόπο την ασφάλεια στα "χέρια" του framework ώστε αυτός να παραμένει προσηλωμένος στην ανάπτυξη της εφαρμογής. Το Laravel ως θερμός οπαδός της ασφάλειας, παρέχει όλα τα εργαλεία που χρειάζεται μία σύγχρονη διαδικτυακή εφαρμογή για να θεωρείται ασφαλής. Παρακάτω παρουσιάζονται διάφορες τεχνικές ασφαλείας που χρησιμοποιήθηκαν στην διαδικτυακή εφαρμογή.

- **Ασφάλεια από CSRF (Cross Site Request Forgery) επιθέσεις [35].** Αποτελεί μία από τις πιο γνωστές επιθέσεις που μπορεί να συμβούν σε μία διαδικτυακή εφαρμογή. Περιληπτικά αυτό το είδος επίθεσης μπορεί να συμβεί από μία third-party πηγή, που παράγει ψεύτικα requests σε περιοχές της εφαρμογής που περιέχουν forms, έτσι μπορεί να περάσει επικίνδυνα δεδομένα στον server ή στην βάση δεδομένων. Μία λύση στην επίθεση αυτή είναι η δημιουργία ενός τυχαίου αλφαριθμητικού μεγάλου μήκους

(`csrf_token`), το οποίο δημιουργείται κρυφά σε κάθε form και "προσθ εται" στο request μαζί με τα δεδομένα που εισάγει ο χρήστης. Έπειτα, στον server, ελέγχεται αρχικά η ύπαρξή του και δεύτερον, εάν είναι ίδιος με αυτόν που στάλθηκε από το form. Έτσι αποτρέπεται αυτή η επίθεση. Το Laravel διαθέτει προκαθορισμένα (default) αυτή την λειτουργία.

- **Ασφάλεια από XSS (Cross Site Scripting) επιθέσεις [36].** Αποτελεί μία πολύ εύκολη επίθεση η οποία μπορεί να πραγματοποιηθεί ακόμα και από έναν χρήστη του συστήματος. Τέτοιου είδους επίθεση μπορεί να συμβεί εισχωρώντας κώδικα javascript με κακόβουλο περιεχόμενο, σε form input ή ακόμα και στο url, με αποτέλεσμα να εκτελεστεί αμέσως ο κώδικας με καταστροφικές συνέπειες για τον server ή την βάση δεδομένων. Μία λύση στην επίθεση αυτή είναι ο έλεγχος της πληροφορίας για html elements και την μετατροπή της σε string. Το Laravel διαθέτει προκαθορισμένα αυτή την λειτουργία.
- **Ασφάλεια από επιθέσεις ψεκασμού εντολών SQL (SQL injection) [37].** Αποτελεί επίθεση που στοχεύει στην αλλαγή των εντολών SQL που επεξεργάζονται την βάση δεδομένων. Χωρίς την χρήση κάποιου είδους προστασίας, μπορεί ο χρήστης να περάσει μαζί με μία προκαθορισμένη εντολή SQL και μία δεύτερη που θα "ρίχνει" (drop) έναν πίνακα της βάσης. Μία λύση στην επίθεση αυτή είναι η χρήση της λειτουργίας PDO [38], η οποία προτείνεται από την γλώσσα PHP σαν ένας τρόπος προστασίας στην συγκεκριμένη επίθεση. Στην ουσία "σπάει" σε τμήματα (bindings) τα δεδομένα που δίνονται σε μία εντολή SQL και τα ελέγχει ατομικά. Το Laravel διαθέτει προκαθορισμένα αυτή την λειτουργία.
- **Λειτουργία συνοδού (session) και στρωμάτων ασφαλείας (middlewares) στο κομμάτι αυθεντικοποίησης, λειτουργία κλειδιού (api_token) στο κομμάτι του api.** Η διαδικτυακή εφαρμογή χωρίζεται σε δύο κομμάτια. Στο κομμάτι αυθεντικοποίησης και στο api. Στο πρώτο κομμάτι, αφού ο χρήστης συνδεθεί, του δίνεται συνοδός (session) 120 λεπτών για πλοήγηση στην εφαρμογή (με το πέρας των 120 λεπτών, η εφαρμογή αυτομάτως τον αποσυνδέει). Παράλληλα του δημιουργείται ένα api_token

που τον βοηθάει στην ταυτοποίησή του από τον server, στα requests που θα κάνει στο api. Ο χρήστης δεν γνωρίζει το api_token ούτε του δίνεται κάποια πληροφορία για το που βρίσκεται. Έπειτα ο χρήστης οδηγείται στο κεντρικό κομμάτι της εφαρμογής, όπου αναλαμβάνει η React μαζί με το React-router και το Redux. Εκεί πλοηγείται στις ενότητες της εφαρμογής, με μοναδικό απαιτούμενο στοιχείο, το api_token. Με την βοήθεια των στρωμάτων ασφαλείας (middlewares), ο εκάστοτε χρήστης σε κάθε ανακατεύθυνση της σελίδας, ελέγχεται για την ιδιότητά του (διαχειριστής, χρήστης) και του επιτρέπεται ή του απορρίπτεται αντίστοιχα η πλοήγηση σε διάφορα τμήματα της εφαρμογής. Τα middlewares αποτελούν κομμάτι του Laravel, δημιουργούνται και ορίζονται από τον χρήστη.

- **Post request σε συγκεκριμένο api endpoint με την χρήση μοναδικού κωδικού (unique) για τον σταθμό.** Κατά την διαδικασία δημιουργίας ενός σταθμού στην διαδικτυακή εφαρμογή, δίνονται οδηγίες στον χρήστη, για το πως θα προγραμματίσει την πραγματική του συσκευή (ενσωματωμένο σύστημα), να στέλνει σε συγκεκριμένο api endpoint τις μετρήσεις που θα λαμβάνει, ώστε να υπάρχει ταυτοποίηση της πραγματικής συσκευής και της εικονικής (σταθμός στην εφαρμογή).

3.4 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό, δόθηκε αναλυτικά η περιγραφή του τρόπου σχεδίασης και ανάπτυξης του συστήματος. Παρουσιάστηκαν οι απαιτήσεις του συστήματος και σχεδιασμός της βάσης δεδομένων και όλων των πινάκων που την απαρτίζουν. Τέλος, αναλύθηκαν λεπτομερώς όλες οι τεχνικές που χρησιμοποιήθηκαν για την επίτευξη της ασφάλειας του συστήματος. Ακολουθεί, η ανάλυση του υλικού μέρους του συστήματος.

Κεφάλαιο 4ο – Υλικό μέρος συστήματος

Σε αυτό το κεφάλαιο θα γίνει η παρουσίαση του υλικού μέρος του συστήματος, το οποίο είναι βασισμένο στον μικρο-επεξεργαστή Arduino Mega. Στον τελευταίο συνδέθηκαν όλες οι περιφερειακές συσκευές που απαιτούνται για τη δημιουργία της υλικής υποδομής ενός ευφυούς περιβαλλοντικού σταθμού. Συγκεκριμένα στο Arduino είναι συνδεδεμένες οι παρακάτω συσκευές και τα ηλεκτρονικά στοιχεία:

- Ethernet Shield
- ESP8266 WiFi module
- Βοηθητικός ανάπτορας για το ESP8266
- Αισθητήρας θερμοκρασίας, υγρασίας και ατμοσφαιρικής πίεσης ονομασίας GY-BME280
- Αισθητήρας καταγραφής συγκέντρωσης σκόνης και μικροσωματιδίων ονομασίας DSM501A
- Αισθητήρας αίσθησης βροχής ονομασίας Raindrop Sensor
- Αισθητήρας υπεριώδους ακτινοβολίας ονομασίας UV Sensor
- Εξάρτημα καταγραφής χρόνου ονομασίας DS3231 RTC
- RGB Led
- Τρανζίστορ τύπου MOSFET ονομασίας IRF520
- Αντιστάσεις

Οι συσκευές που συνδέονται στο Arduino μπορούν να χωριστούν σε συσκευές εισόδου, οι οποίες τροφοδοτούν το Arduino με δεδομένα, σε συσκευές εξόδου, οι οποίες παρέχουν πληροφορίες στον χρήστη, καθώς και σε συσκευές διεπαφής, οι οποίες αναλαμβάνουν να γεφυρώσουν την επικοινωνία μεταξύ του υλικού και του λογισμικού μέρους του συστήματος.

Το Ethernet Shield καθώς και το WiFi module, αποτελούν συσκευές διεπαφής και αναλαμβάνουν να εκτελέσουν την αμφίδρομη επικοινωνία μεταξύ της διαδικτυακής εφαρμογής και της συσκευής Arduino. Οι αισθητήρες που αναγράφονται παραπάνω,

αποτελούν συσκευές εισόδου και τροφοδοτούν με δεδομένα το Arduino, ενώ το RGB Led, έχει το ρόλο συσκευής εξόδου και πληροφορεί τον χρήστη για διάφορες καταστάσεις που βρίσκεται το Arduino.

Στις υποενότητες που ακολουθούν θα γίνει παρουσίαση των τεχνικών χαρακτηριστικών όλων των συσκευών και ηλεκτρονικών στοιχείων που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία. Θα δοθούν αναλυτικές πληροφορίες για τη λειτουργία του καθενός, καθώς και για τον τρόπο με τον οποίο συνδέονται στο Arduino. Έπειτα, θα παρουσιαστεί το τελικό σχέδιο συνδεσμολογίας και θα δοθεί επεξήγηση ορισμένων στοιχείων που προκύπτουν από αυτό. Τέλος θα αναλυθεί η κατανάλωση ενέργειας, θα παρουσιαστεί το ποσοστό εξοικονόμησης που επετεύχθη καθώς επίσης, θα δοθεί το συνολικό κοστολόγιο των υλικών (BOM ή Bill Of Material).

4.1 Arduino Mega 2560 Rev3

Το Arduino Mega 2560 Rev3 [39] αποτελεί την πιο εξελιγμένη πλακέτα της οικογένειας Arduino και προτείνεται για περίπλοκες κατασκευές που απαιτούν μεγαλύτερη μνήμη και περισσότερες εισόδους/εξόδους. Το μοντέλο Mega 2560 της οικογένειας Arduino, είναι ένα από τα πιο διαδεδομένα μοντέλα μικροεπεξεργαστών και είναι συμβατό με πλήθος αισθητήρων και επεκτάσεων, βασίζεται στην ίδια αρχιτεκτονική με την πλακέτα Uno και η βασική της διαφορά είναι η χωρητικότητα της μνήμης και το πλήθος των εισόδων-εξόδων για την σύνδεση με εξωτερικές συσκευές. Η έκδοση 3 (Rev3) είναι η τελευταία που κυκλοφορεί και είναι πλήρως συμβατή με καινούργιες και παλιές πλακέτες επέκτασης.

Το Arduino Mega βασίζεται στον μικροελεγκτή ATmega2560 της Atmel. Είναι μία ολοκληρωμένη πλακέτα που περιέχει ότι χρειάζεται για να μπορεί να προγραμματιστεί και να λειτουργήσει συνδέοντας την με ένα απλό καλώδιο USB στον υπολογιστή. Επίσης λόγω της φορητότητας της, μπορεί να λειτουργήσει με ένα powerbank ή με ένα τροφοδοτικό στην πρίζα ή ακόμα και με μία απλή μπαταρία. Αναλυτικά η πλακέτα διαθέτει 54 ψηφιακές εισόδους ή εξόδους από τις οποίες, 15 μπορεί να χρησιμοποιηθούν σαν PWM (Pulse Width Modulation) εξόδους [40], 16 αναλογικές εισόδους, 4 σειριακές θύρες (UARTs), 1 θύρα USB (τύπου B) για τον προγραμματισμό και την τροφοδοσία της

πλακέτας, 1 είσοδο τροφοδοσίας που μπορεί να χρησιμοποιηθεί εναλλακτικά για τροφοδοσία από τροφοδοτικό πρίζας ή από απλή μπαταρία, 1 υποδοχή ICSP και τέλος κουμπί για το reset της πλακέτας. Επίσης διαθέτει τρία πρωτόκολλα επικοινωνίας, τα οποία αποτελούν και τρόποι συνδεσμολογίας με αρκετά σύνθετα εξαρτήματα. Αυτά είναι το I2C [44], το SPI [45] και το UART [46]. Ο μικροελεγκτής είναι συγχρονισμένος στους 16 μεγακύκλους (Crystal 16MHz)

Η μνήμη Flash του Arduino Mega, η οποία χρησιμοποιείται στην αποθήκευση του προγράμματος (sketch) είναι 256KB, ικανή να δεχτεί τα περισσότερα περίπλοκα προγράμματα σε αντίθεση με την μικρότερη έκδοση της Arduino, το Arduino Uno.

Το Arduino mega 2560 λειτουργεί με τροφοδοσία 5V DC από την είσοδο του USB ή με 7V/12V DC (συνιστώμενη από κατασκευαστή) από την είσοδο της τροφοδοσίας χωρίς να υπάρχει κίνδυνος ηλεκτροπληξίας.

Τεχνικά χαρακτηριστικά:

- Μικροελεγκτής ATmega2560
- Αρχιτεκτονική ελεγκτή AVR
- Τάση λειτουργίας: 5 VDC
- Τάση εισόδου 7-12 V προτεινόμενη
- Τάση εισόδου 6-20 V limit, min-max
- Ψηφιακές εισόδους/εξόδους 54 (15 PWM εξόδους)
- PWM Ψηφιακές εισόδους/εξόδους 16
- Αναλογικές εισόδους 16
- Ρεύμα ανά είσοδο/έξοδο 20mA
- Ρεύμα ανά είσοδο/έξοδο 3.3V 50mA
- Μνήμη Flash 256 KB από τα οποία 8 KB χρησιμοποιούνται για το σύστημα
- Μνήμη SRAM 8 KB
- Μνήμη EEPROM 4 KB
- Ταχύτητα 16 MHz

Τέλος αξίζει να αναφερθεί ότι προτιμήθηκε η συγκεκριμένη πλακέτα έναντι της υποδεέστερης σε δυνατότητες και χαρακτηριστικά, Arduino Uno, για την υλοποίηση του περιβαλλοντικού σταθμού διότι, το μέγεθος του sketch και των συμπεριλαμβανομένων βιβλιοθηκών ήταν απαγορευτικό για τις μνήμες Flash και SRAM του Uno.

Παρακάτω παρουσιάζεται μία εικόνα της πλακέτας Arduino Mega 2560 Rev3.



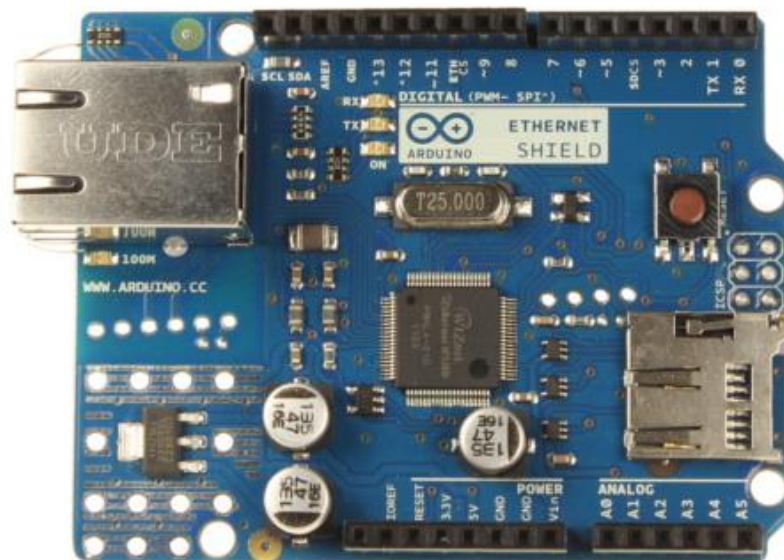
Εικόνα 22: Πλακέτα Arduino Mega 2560 Rev3

4.2 Ethernet Shield

Το Ethernet Shield [41], αποτελεί στην ουσία ένωση δύο ξεχωριστών εξαρτημάτων, του μικροσίπ για πρόσβαση στο διαδίκτυο (Wiznet W5100 ethernet chip) και του εξαρτήματος υποδοχής SD card. Η συγκεκριμένη πλακέτα προσδίδει στο Arduino τη δυνατότητα να συνδέεται στο διαδίκτυο ενσύρματα, καθώς και δυνατότητα ανάγνωσης, επεξεργασίας και αποθήκευσης δεδομένων σε κάρτα micro-SD. Συνδέεται από πάνω από το Arduino, στις υποδοχές που διαθέτει και τροφοδοτείται με ρεύμα από αυτό. Επικοινωνεί με το τελευταίο χρησιμοποιώντας το πρωτόκολλο επικοινωνίας SPI (μέσω της υποδοχής ICSP). Αυτό βρίσκεται στους ακροδέκτες 50, 51 και 52 του Mega. Μέσω του ακροδέκτη 10 επιλέγεται η χρήση του Ethernet Controller (Wiznet), ενώ μέσω του ακροδέκτη 4 επιλέγεται ο micro-SD card reader. Συνεπώς, οι 5 ακροδέκτες που μόλις αναφέρθηκαν δεν

μπορούν να χρησιμοποιούνται σαν είσοδοι/έξοδοι για το Arduino. Επίσης για να λειτουργήσει το Shield στο Mega θα πρέπει να δηλωθεί ο ακροδέκτης 53 (λειτουργεί ως ακροδέκτης επιλογής hardware) ως ακροδέκτης εξαγωγής (output pin). Στην παρούσα διπλωματική χρησιμοποιήθηκε η τρίτη έκδοση του shield (R3), η οποία βασίζεται στο Ethernet chip Wiznet W5100. Υποστηρίζει τα πρωτόκολλα TCP και UDP και μπορεί να έχει 4 ταυτόχρονες socket connections, ενώ για τον προγραμματισμό του χρησιμοποιεί τη βιβλιοθήκη Ethernet. Αξίζει να σημειωθεί πως δεν υποστηρίζει το πρωτόκολλο HTTPS.

Παρακάτω παρουσιάζεται μία εικόνα του Ethernet Shield R3.



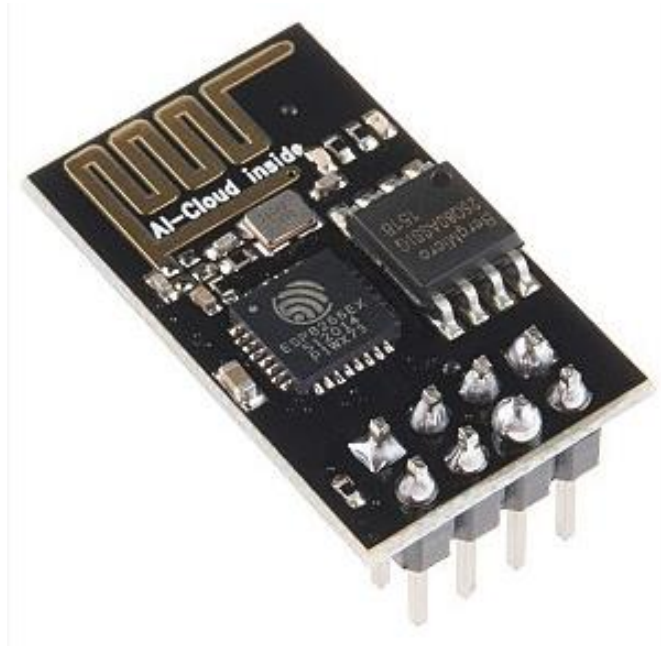
Εικόνα 23: Ethernet Shield R3

4.3 ESP8266 WiFi Module και βοηθητικός αντάπτορας

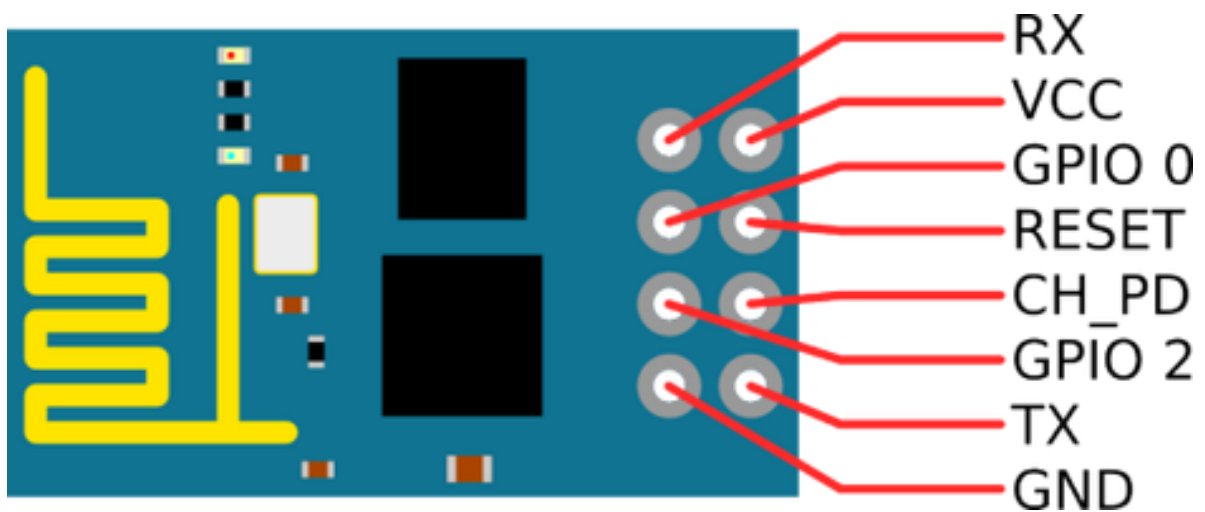
Για την σύνδεση του Arduino με το διαδίκτυο μέσω WiFi επικοινωνίας, υπάρχουν διάφορα modules και shields που το καθιστούν εφικτό. Τα εξαρτήματα αυτά ποικίλουν σε χαρακτηριστικά, δυνατότητες και κόστος. Ένα module που ίσως είναι και το πιο διαδεδομένο για τον σκοπό αυτό είναι το ESP8266 [42]. Αποτελεί ένα ξεχωριστό είδος εξαρτήματος διότι, το ίδιο το εξάρτημα αποτελεί έναν αυτόνομο μικρό μικροεπεξεργαστή. Αυτό σημαίνει ότι μπορεί να προγραμματιστεί με sketch χωρίς την σύνδεση του με το

Arduino, καθώς επίσης μπορεί να λειτουργήσει σαν ένα βοηθητικό κομμάτι σε πλακέτα Arduino, αλλά και σαν μεμονωμένο εξάρτημα μόνο με την παροχή ηλεκτρικής ενέργειας. Υπάρχουν 14 εκδόσεις ESP8266, διαφορετικών χαρακτηριστικών. Στην παρούσα διπλωματική χρησιμοποιήθηκε η πρώτη έκδοση. Λειτουργεί με το πρωτόκολλο επικοινωνίας (UART).

Στα θετικά του εξαρτήματος λογίζονται το μικρό του κόστος, καθώς και η μεγάλη κοινότητα από προγραμματιστές που το υποστηρίζει με κώδικα open-source και συμβουλές υλοποίησης. Στα αρνητικά αρχικά, μπορούμε να τοποθετήσουμε την κακή σχεδίαση των pins σε αρκετές εκδόσεις, καθώς λόγω της κατασκευής του δεν τοποθετείται εύκολα σε breadboard και χρειάζεται υποβοήθηση με καλώδια. Επίσης ένα ακόμη πρόβλημα μπορεί να θεωρηθεί, πως το εξάρτημα λειτουργεί με μέγιστη τάση τα 3.6V και με την πιθανή παροχή του με τάση ίση των 5V καταστρέφεται. Παρόλο που το Arduino διαθέτει pin που τροφοδοτεί 3.3V, ωστόσο η ποσότητα του ρεύματος που παρέχει, δεν επαρκεί για την λειτουργία του εξαρτήματος. Δεδομένο που μας οδηγεί στην χρήση εξαρτημάτων όπως logic level converters ή voltage regulators ή άλλων συνδυασμών από πυκνωτές και αντιστάσεις και ταυτόχρονη παροχή 5V. Όλα αυτά τα παραδείγματα αναφέρονται διότι μετατρέπουν τα 5V σε 3.3V και παράλληλα διατηρούν την ποσότητα του ρεύματος σε υψηλό επίπεδο και έτσι λειτουργεί το εξάρτημα. Οι συγκεκριμένες υλοποιήσεις απαιτούν καλές γνώσεις πάνω στον τομέα της ηλεκτρονικής ωστόσο χρειάζονται αρκετές δοκιμές ώστε να έρθει το επιθυμητό αποτέλεσμα. Μία καλή λύση, είναι η χρήση έτοιμων adapters που έχουν υλοποιηθεί καθαρά για αυτό το εξάρτημα και εξαλείφουν τις προηγούμενες τεχνικές. Τελευταίο και πιο σημαντικό πρόβλημα είναι ότι, το συγκεκριμένο εξάρτημα για να προγραμματιστεί και να λειτουργήσει ως κομμάτι του Arduino, απαιτεί την εγκατάσταση κάποιου λογισμικού και η διαδικασία εγκατάστασης του είναι δυσνόητη και αρκετά δύσκολη. Λαμβάνοντας υπ' όψη όλα τα αρνητικά στοιχεία, καταλαβαίνουμε ότι το συγκεκριμένο εξάρτημα δεν απευθύνεται σε αρχάριους χρήστες του Arduino. Παρακάτω ακολουθεί μία φωτογραφία του ESP8266 καθώς και ένα σχήμα που παρουσιάζει τους ακροδέκτες. Η ανάλυση τους ακολουθεί μετά τις εικόνες.



Εικόνα 24: ESP8266 WiFi module

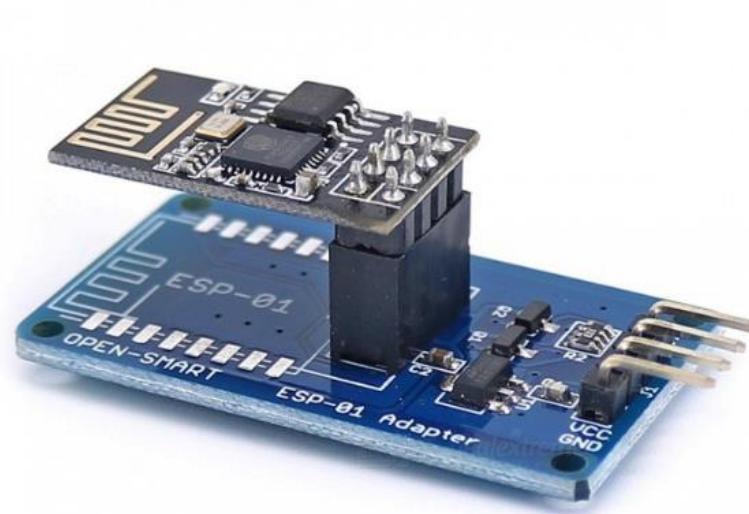


Εικόνα 25: Ακροδέκτες ESP8266

- **VCC:** Ακροδέκτης ισχύος μέσω του οποίου παρέχεται τάση 3.3V.
- **GND:** Ακροδέκτης γείωσης.
- **TX:** Ακροδέκτης που χρησιμοποιείται για τη μετάδοση σειριακών δεδομένων σε άλλες συσκευές.
- **RX:** Ακροδέκτης που χρησιμοποιείται για τη λήψη σειριακών δεδομένων από άλλες συσκευές.

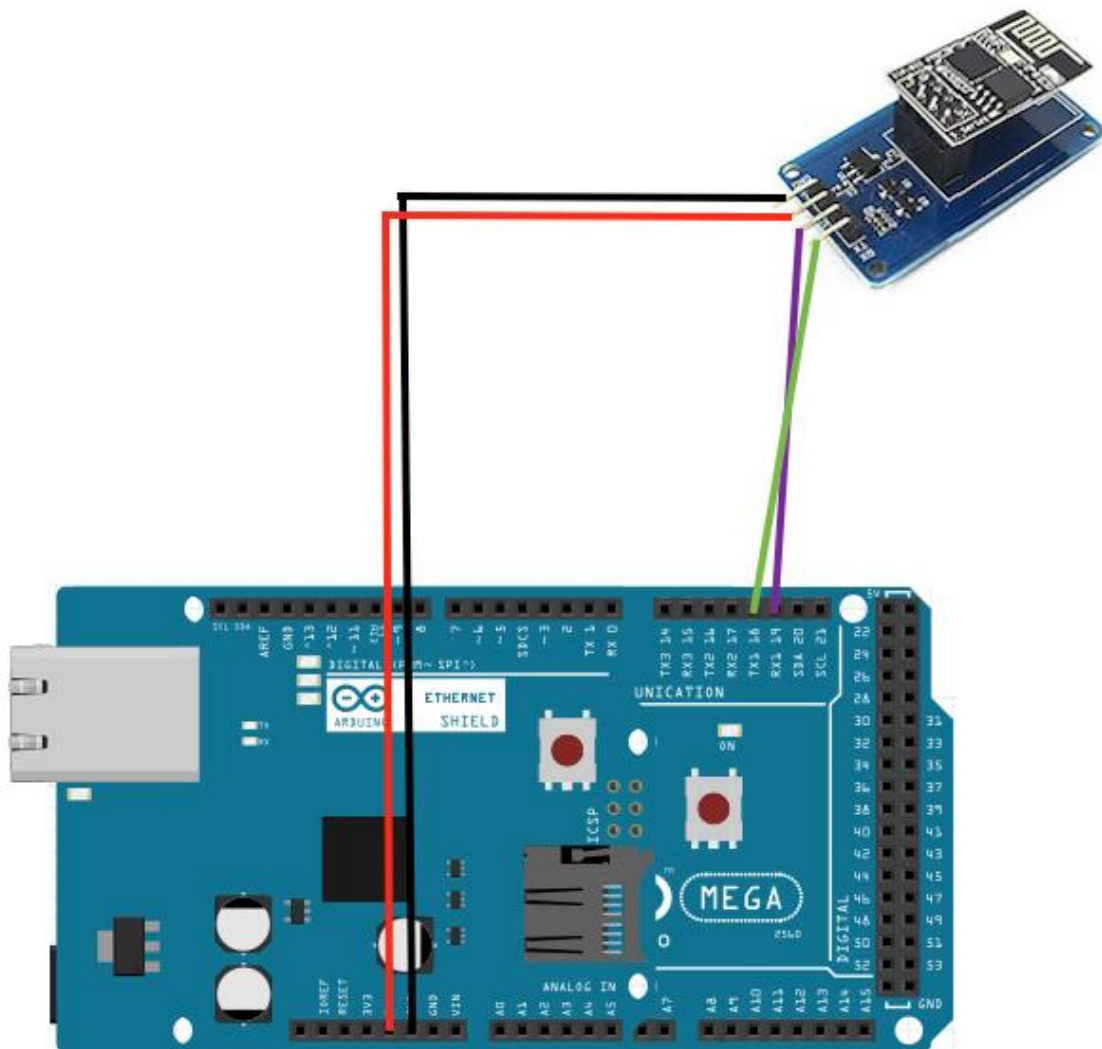
- **RST:** Είναι ο ακροδέκτης επαναφοράς (RESET pin). Το ESP8266 θα επαναφερθεί αν ο ακροδέκτης RESET λάβει σήμα LOW.
- **CH_PD:** Ακροδέκτης που ενεργοποιεί το τσιπ.
- **GPIO0:** Ο ακροδέκτης GPIO0 (Γενικής χρήσης I / O) έχει δύο λειτουργίες. Μία λειτουργία γενικού σκοπού και μία δεύτερη για ενεργοποίηση του προγραμματισμού του ESP8266.
- **GPIO2:** Ακροδέκτης γενικού σκοπού.

Παρακάτω ακολουθεί φωτογραφία απεικόνισης του ESP8266 τοποθετημένου σε adapter.



Εικόνα 26: ESP8266 με adapter

Τέλος ακολουθεί το παράδειγμα συνδεσμολογίας με το Arduino Mega.

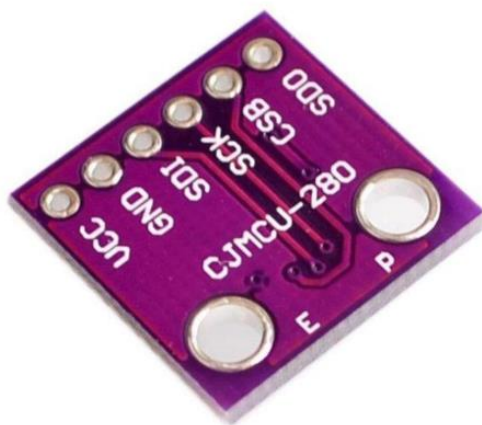


Εικόνα 27: Συνδεσμολογία ESP8266 με Arduino Mega

- **Arduino VCC (κόκκινο):** ESP8266 pin 2 (VCC)
- **Arduino GND (μαύρο):** ESP8266 pin 1 (GND)
- **Arduino PIN 18 (πράσινο):** ESP8266 pin 3 (TX)
- **Arduino PIN 19 (μωβ):** ESP8266 pin 4 (RX)

4.4 Αισθητήρας θερμοκρασίας, υγρασίας και ατμοσφαιρικής πίεσης ονομασίας GY-BME280

Ο αισθητήρας GY-BME280 (ή BMP280) είναι ένας ψηφιακός αισθητήρας θερμοκρασίας, υγρασίας και ατμοσφαιρικής πίεσης, χαμηλού κόστους και υψηλής ακριβείας [43]. Πρόκειται για μία σίγουρη επιλογή εάν απαιτείται καταγραφή μέτρησης κάποιας από τις κατηγορίες που αναφέρθηκαν προηγουμένως. Το θετικό του αισθητήρα έχει να κάνει με το γεγονός ότι, στην ουσία αποτελεί ένωση τριών αισθητήρων σε έναν. Αυτό μεταφράζεται ως μείωση κόστους, απαιτούμενου χώρου στην breadboard καθώς και προγραμματιστικού κώδικα. Έχει 6 ακροδέκτες σύνδεσης και παρέχει την επιλογή λειτουργίας με δύο πρωτόκολλα επικοινωνίας (I2C και SPI). Το πρωτόκολλο που επιλέχθηκε είναι το I2C. Παρακάτω ακολουθεί φωτογραφία απεικόνισης του GY-BME280 μαζί με τα τεχνικά του χαρακτηριστικά όπως αυτά αναφέρονται από τον κατασκευαστή.

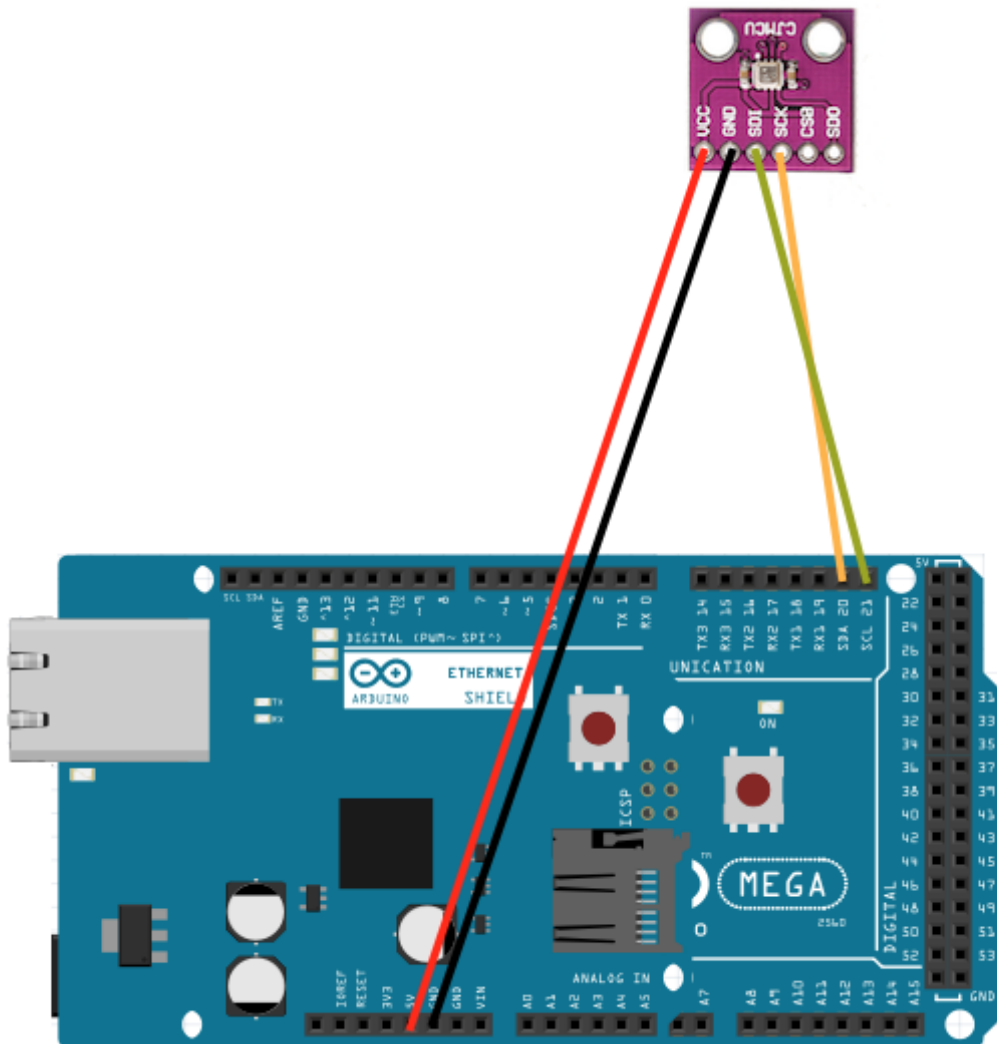


Εικόνα 28: GY-BME280

Τεχνικά χαρακτηριστικά:

- Τάση λειτουργίας: 1.8 - 5 VDC
- Διεπαφή: I2C (όριο στα 3.4MHz), SPI (όριο στα 10MHz)
- Λειτουργική εμβέλεια: Θερμοκρασία (-40 έως +85°C), Υγρασία (0 - 100%), Πίεση (300 - 1100 hPa)
- Ακρίβεια μετρήσεων: Θερμοκρασία (+-1°C), Υγρασία (+-3%), Πίεση (+-1Pa)

Τέλος ακολουθεί το παράδειγμα συνδεσμολογίας με το Arduino Mega.



Εικόνα 29: Συνδεσμολογία GY-BME280 με Arduino Mega

- **Arduino VCC (κόκκινο):** GY-BME280 pin 1 (VCC)
- **Arduino GND (μαύρο):** GY-BME280 pin 2 (GND)
- **Arduino PIN 20 (κίτρινο):** GY-BME280 pin 4 (SCK)
- **Arduino PIN 21 (πράσινο):** GY-BME280 pin 3 (SDI)

4.5 Αισθητήρας καταγραφής συγκέντρωσης σκόνης και μικροσωματιδίων ονομασίας DSM501A

Ο αισθητήρας DSM501A [47] αποτελεί ιδανική λύση για την καταγραφή της συγκέντρωσης σκόνης και μικροσωματιδίων. Δίνει μία καλή ένδειξη της ποιότητας του αέρα στο περιβάλλον όπου τοποθετείται. Η λειτουργία του βασίζεται σε μία εσωτερική αντίσταση που διαθέτει, η οποία καθώς πυρακτώνεται δημιουργεί ανοδικό ρεύμα αέρα (updraft) και έχει σαν αποτέλεσμα το εξάρτημα να "τραβάει" αέρα στο εσωτερικό του. Το επίπεδο συγκέντρωσης μικροσωματιδίων υπολογίζεται, μετρώντας τον χρόνο παλμικής κατανομής (Low Pulse Occurance ή LPO [48]) σε δεδομένη χρονική μονάδα. Ο αισθητήρας παρέχει μεγάλη ευαισθησία ακόμα και σε μικροσωματίδια διαμέτρου του ενός μικρομέτρου (1 μ m). Έχει 5 ακροδέκτες σύνδεσης, δύο από τους οποίους ορίζουν και την ευαισθησία, ωστόσο ένας από τους δύο επιλέγεται και συνδέεται στο Arduino. Παρακάτω ακολουθεί φωτογραφία απεικόνισης του DSM501A μαζί με τα τεχνικά του χαρακτηριστικά όπως αυτά αναφέρονται από τον κατασκευαστή.



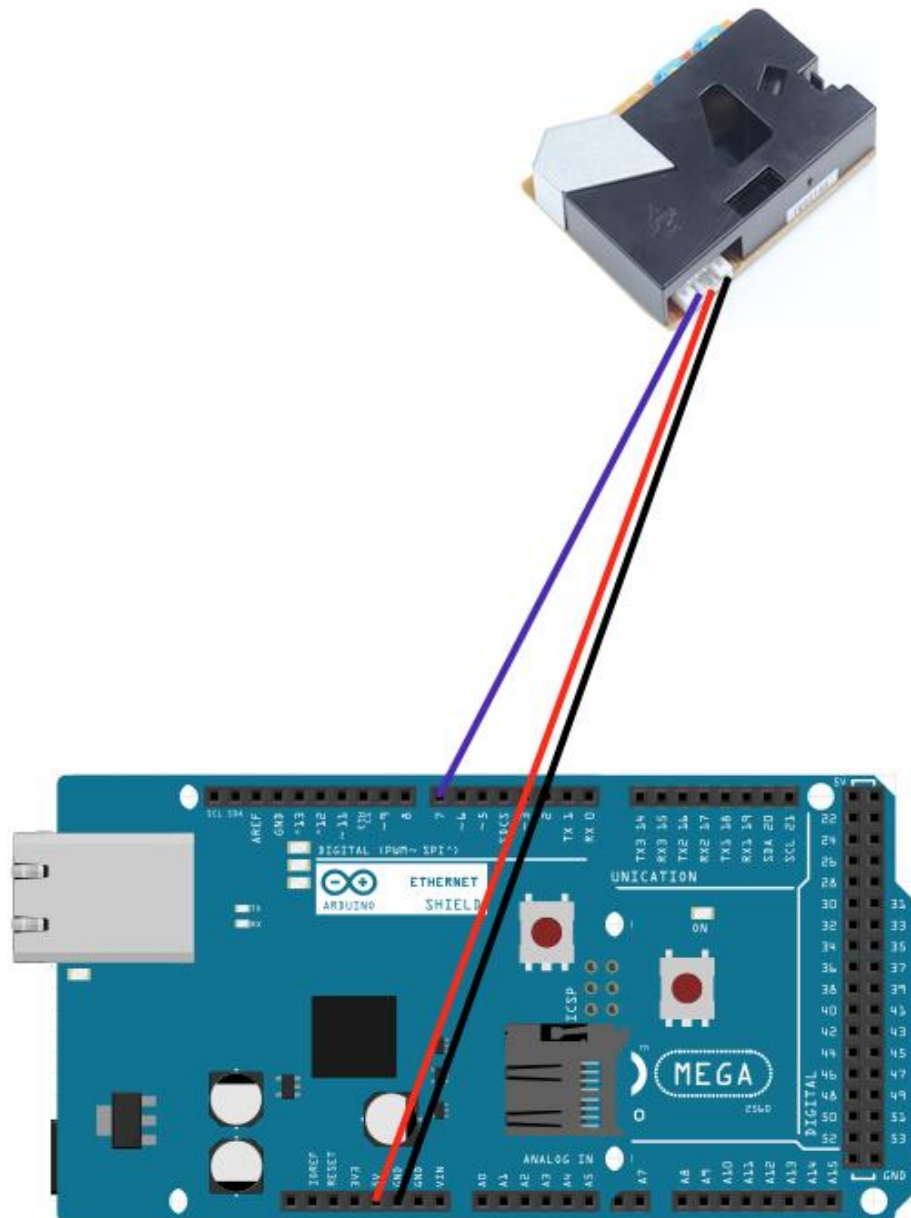
Εικόνα 30: DSM501A

Τεχνικά χαρακτηριστικά:

- Τάση λειτουργίας: 4.75 - 5.75 VDC
- Κατανάλωση ρεύματος σε κατάσταση standby: 90 mA

- Εμβέλεια ανιχνεύσιμης συγκέντρωσης: 0 ~ 28,000 / 0 ~ 8000 (pcs/liter / pcs/0.01cf) (επεξήγηση μονάδων: pcs (particles): μικροσωματίδια, liter: λίτρο, cf (cubic foot): κυβικό πόδι - μονάδα αντίστοιχη του λίτρου)
- Εμβέλεια ορθής λειτουργίας σε θερμοκρασία: 0 ~ 45 °C

Τέλος ακολουθεί το παράδειγμα συνδεσμολογίας με το Arduino Mega.

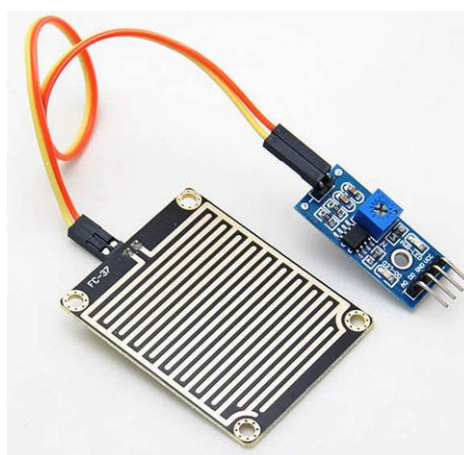


Εικόνα 31: Συνδεσμολογία DSM501A με Arduino Mega

- **Arduino VCC (κόκκινο):** DSM501A pin 3 (VCC)
- **Arduino GND (μαύρο):** DSM501A pin 5 (GND)
- **Arduino PIN 7 (μπλε):** DSM501A pin 2 ($> 1\mu\text{m}$)

4.6 Αισθητήρας αίσθησης βροχής ονομασίας Raindrop Sensor

Ο αισθητήρας Raindrop Sensor [49] αποτελεί ένα εξάρτημα το οποίο έχει την δυνατότητα να καταγράφει την αίσθηση βροχής και να ενημερώνει αντίστοιχα το Arduino. Διαθέτει μία πλαστική επιφάνεια με μεταλλικές χαράξεις στο πάνω μέρος, οι οποίες με την επαφή υγρού στοιχείου δημιουργούν διαφορά δυναμικού στους ακροδέκτες και έτσι, ο αισθητήρας μετατρέπει αυτή την διαφορά, σε πληροφορία που είναι κατανοητή για τον χρήστη και την αποστέλλει στο Arduino. Αξίζει να αναφερθεί πως το συγκεκριμένο εξάρτημα δεν μετράει ποσότητα υγρού ή συγκέντρωση υγρού αντίστοιχη στο χρόνο ή κάτι παρεμφερές. Απλώς δίνει μία ένδειξη για το αν υπάρχει υγρό στην επιφάνεια ή όχι. Όσον αφορά την περιβαλλοντική μέτρηση του σταθμού, το αποτέλεσμα που δίνει είναι εάν υπάρχει ένδειξη βροχής, ένδειξη για υγρασία ή μικρής έντασης βροχή (ψιχάλα) ή τίποτα από τα δύο. Διαθέτει τέσσερις ακροδέκτες και είναι αρκετά εύκολος στην τοποθέτηση του καθώς και στον προγραμματισμό του. Το αποτέλεσμα που γυρίζει στο Arduino είναι αναλογικό και για αυτό τοποθετείται αντίστοιχα σε αναλογικό ακροδέκτη. Παρακάτω ακολουθεί φωτογραφία απεικόνισης του Raindrop Sensor μαζί με τα τεχνικά του χαρακτηριστικά όπως αυτά αναφέρονται από τον κατασκευαστή.

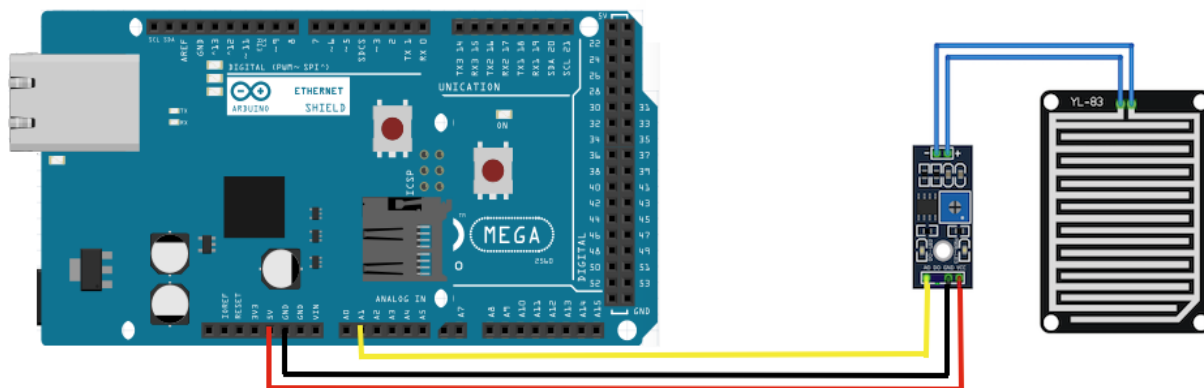


Εικόνα 32: Raindrop Sensor

Τεχνικά χαρακτηριστικά:

- Τάση λειτουργίας: 5 VDC
- Αντιοξειδωτική επιφάνεια χωρίς αγωγιμότητα
- Ύπαρξη ποτενσιόμετρου για προσαρμογή ευαισθησίας

Τέλος ακολουθεί το παράδειγμα συνδεσμολογίας με το Arduino Mega.



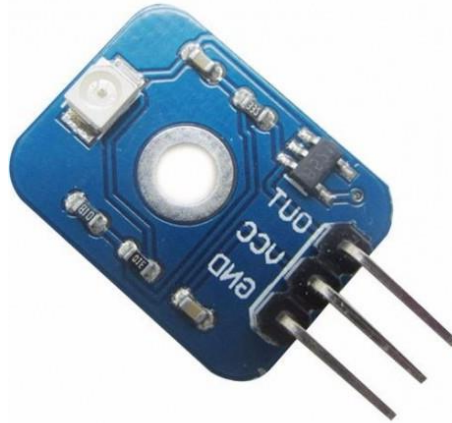
Εικόνα 33: Συνδεσμολογία Raindrop Sensor με Arduino Mega

- **Arduino VCC (κόκκινο):** Raindrop sensor pin 4 (VCC)
- **Arduino GND (μαύρο):** Raindrop sensor pin 3 (GND)
- **Arduino PIN A1 (κίτρινο):** Raindrop sensor pin 1 (A0)

4.7 Αισθητήρας υπεριώδους ακτινοβολίας ονομασίας UV Sensor

Ο αισθητήρας UV Sensor [50] αποτελεί ένα εξάρτημα το οποίο έχει την δυνατότητα να καταγράφει την υπεριώδης ακτινοβολία [51]. Λειτουργεί με παρόμοιο τρόπο όπως το Raindrop Sensor, καθώς η διαφορά δυναμικού είναι αυτή που τελικά μεταφράζεται αντιστοίχως και δίνει την πληροφορία. Επίσης το αποτέλεσμα που επιστρέφει είναι αναλογικό. Δεν υπάρχει κάτι άλλο το αξιοσημείωτο σχετικά με τον συγκεκριμένο αισθητήρα. Διαθέτει τρεις ακροδέκτες και η τοποθέτηση του καθώς και ο προγραμματισμός του είναι αρκετά εύκολες διαδικασίες. Παρακάτω ακολουθεί

φωτογραφία απεικόνισης του UV Sensor μαζί με τα τεχνικά του χαρακτηριστικά όπως αυτά αναφέρονται από τον κατασκευαστή.

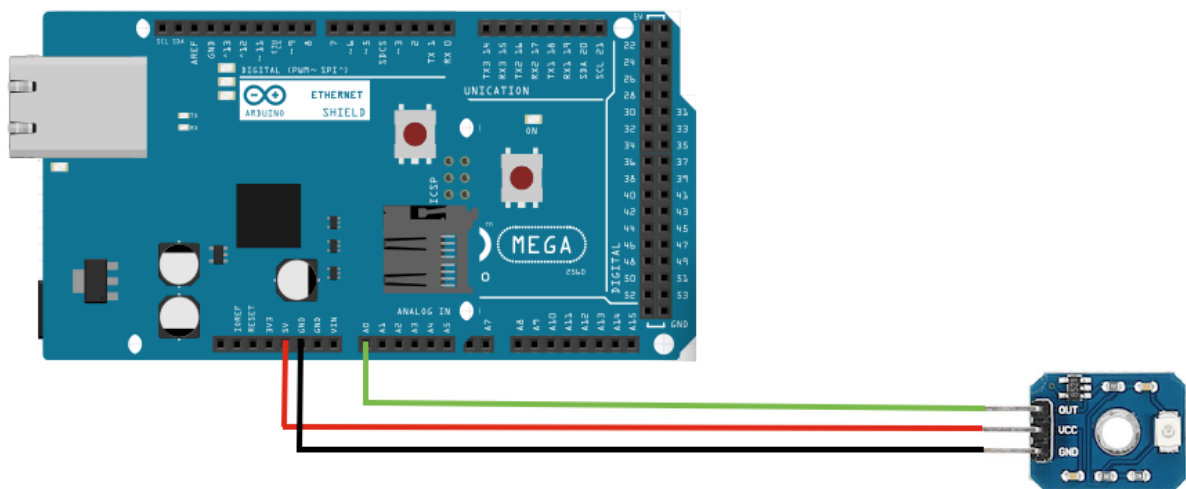


Εικόνα 34: UV Sensor

Τεχνικά χαρακτηριστικά:

- Τάση λειτουργίας: 5 VDC
- Μικρή κατανάλωση ενέργειας (300μΑ)

Τέλος ακολουθεί το παράδειγμα συνδεσμολογίας με το Arduino Mega.



Εικόνα 35: Συνδεσμολογία UV Sensor με Arduino Mega

- **Arduino VCC (κόκκινο):** UV sensor pin 2 (VCC)
- **Arduino GND (μαύρο):** UV sensor pin 3 (GND)
- **Arduino PIN A0 (πράσινο):** UV sensor pin 1 (OUT)

4.8 Εξάρτημα καταγραφής χρόνου ονομασίας DS3231 RTC

Το τσιπ DS3231 RTC (Real Time Clock) [52] αποτελεί στην ουσία έναν χρονιστή, ο οποίος κρατάει αποθηκευμένη την τρέχουσα ημερομηνία και ώρα. Επίσης αποτελεί ένα σημαντικό εργαλείο το οποίο μπορεί να χρησιμοποιηθεί, για την διακοπή της κατάστασης μειωμένης ενέργειας που μπορεί να τεθεί το Arduino (εξηγείται αναλυτικά σε επόμενη υποενότητα). Είναι εξάρτημα μέγιστης σημασίας για την λειτουργία του σταθμού διότι εκτός από το να κρατάει την ημερομηνία και ώρα και να την συμπληρώσει στο sketch σε ορισμένες περιπτώσεις, είναι αυτό που είναι υπεύθυνο στο να "ξυπνάει" το Arduino και να ρυθμίζει τα διαστήματα του ύπνου του (κατάσταση μειωμένης ενέργειας). Η λειτουργία του ξεκινάει από την στιγμή που θα το προγραμματίσουμε για πρώτη φορά, με την τωρινή ημερομηνία και ώρα. Έπειτα δουλεύει συνεχώς ανανεώνοντας τα στοιχεία αυτά. Η λειτουργία αυτή επιτυγχάνεται χάρις την μπαταρία 3V που διαθέτει, η οποία το καθιστά αυτόνομο για αρκετούς μήνες λειτουργίας. Το συγκεκριμένο μοντέλο διαθέτει έναν ακροδέκτη που χρησιμοποιείται καθαρά σαν διακόπτης (interrupt). Λειτουργεί με το πρωτόκολλο επικοινωνίας I2C και διαθέτει πέντε ακροδέκτες. Τέλος αξίζει να αναφερθεί ότι έχει και ενσωματωμένο αισθητήρα θερμοκρασίας. Παρακάτω ακολουθεί φωτογραφία απεικόνισης του DS3231 RTC μαζί με τα τεχνικά του χαρακτηριστικά όπως αυτά αναφέρονται από τον κατασκευαστή.

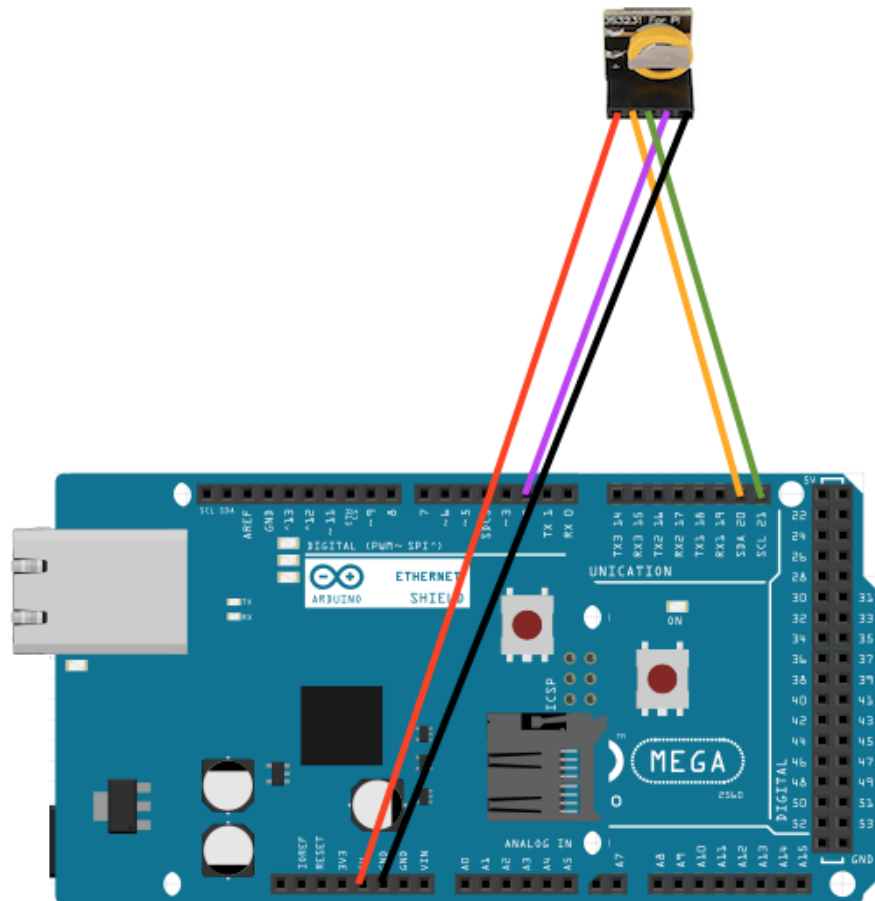


Εικόνα 36: DS3231 RTC

Τεχνικά χαρακτηριστικά:

- Τάση λειτουργίας: 3,3 ~ 5 VDC
- Μικρή κατανάλωση ενέργειας (300μΑ)

Τέλος ακολουθεί το παράδειγμα συνδεσμολογίας με το Arduino Mega.



Εικόνα 37: Συνδεσμολογία DS3231 RTC με Arduino Mega

- **Arduino VCC (κόκκινο):** RTC pin 1 (VCC)
- **Arduino GND (μαύρο):** RTC pin 5 (GND)
- **Arduino PIN 2 (μωβ):** RTC pin 4 (Interrupt)
- **Arduino PIN 20 (πορτοκαλί):** RTC pin 2 (SDA)
- **Arduino PIN 21 (πράσινο):** RTC pin 3 (SCL)

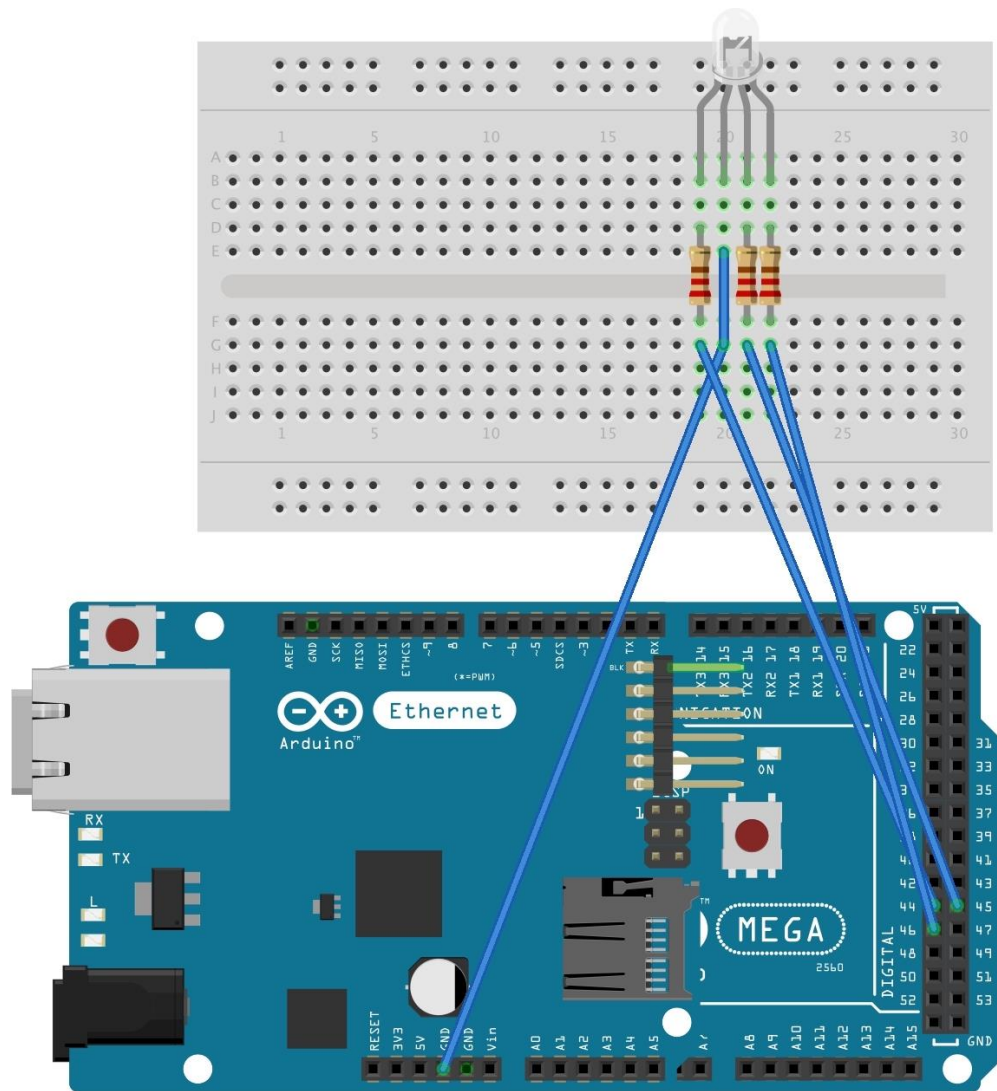
4.9 RGB Led

LED (Light Emitting Diode ή Δίοδος Εκπομπής Φωτός) [53], αποκαλείται ένας ημιαγωγός ο οποίος εκπέμπει φωτεινή ακτινοβολία στενού φάσματος όταν του παρέχεται μία ηλεκτρική τάση. Η χρήση του είναι αρκετά συχνή σε υλοποιήσεις αυτοματοποιημένων συστημάτων ή ακόμα και έξυπνων συσκευών και ο λόγος, έχει να κάνει με την ενημέρωση του χρήστη, για διάφορες καταστάσεις που βρίσκεται το σύστημα ή για πιθανές ενέργειες που πρόκειται να συμβούν. Στην παρούσα υλοποίηση του περιβαλλοντικού σταθμού, θεωρήθηκε αναγκαία η προσθήκη τέτοιων ημιαγωγών για την έγκαιρη και στοχευμένη ενημέρωση του χρήστη, για διάφορες καταστάσεις και ενέργειες του σταθμού. Ωστόσο λόγω των πολλών πιθανών καταστάσεων που μπορεί να βρεθεί ο σταθμός, επιλέχθηκαν αρκετές χρωματικές ενδείξεις, το οποίο συνεπάγεται με την παράλληλη χρήση πολλών απλών LED. Η λύση αυτή όμως δεν θεωρείται ιδανική, λόγω της αχρείαστης αύξησης της πολυπλοκότητας των καλωδιώσεων που θα προκύψουν. Για αυτό τον λόγο επιλέχθηκε το RGB Led, στο οποίο, η δημιουργία διαφοράς δυναμικού στους ακροδέκτες του έχει ως αποτέλεσμα να αλλάζει η ένδειξη του σε ένα μεγάλο εύρος χρωμάτων. Το RGB Led δεν διαφέρει σε τίποτα από ένα απλό LED, έτσι και αυτό χρειάζεται την τοποθέτηση αντιστάσεων στους ακροδέκτες του, ώστε να μην καταστραφεί από την ένταση του ηλεκτρικού ρεύματος. Στην παρούσα υλοποίηση τοποθετήθηκαν αντιστάσεις των 220Ω. Αποτελείται από τέσσερις ακροδέκτες. Παρακάτω ακολουθεί φωτογραφία απεικόνισης του RGB Led.



Εικόνα 38: RGB Led

Τέλος ακολουθεί το παράδειγμα συνδεσμολογίας με το Arduino Mega.



Εικόνα 39: Συνδεσμολογία RGB Led με Arduino Mega

- **Arduino GND:** RGB pin 2 (GND)
- **Arduino PIN 44:** RTC pin 1
- **Arduino PIN 45:** RTC pin 3
- **Arduino PIN 46:** RTC pin 4

4.10 Τρανζίστορ

Το τρανζίστορ (transistor) [54], που ονομάζεται στα ελληνικά κρυσταλλοτρίοδος, είναι διάταξη ημιαγωγών στερεάς κατάστασης, η οποία βρίσκει διάφορες εφαρμογές στην ηλεκτρονική, μερικές εκ των οποίων είναι η ενίσχυση, η σταθεροποίηση τάσης, η διαμόρφωση συχνότητας, η λειτουργία ως διακόπτης και ως μεταβλητή ωμική αντίσταση. Το τρανζίστορ μπορεί, ανάλογα με την τάση με την οποία πολώνεται, να ρυθμίζει την ροή του ηλεκτρικού ρεύματος που απορροφά από συνδεδεμένη πηγή τάσης. Τα τρανζίστορ κατασκευάζονται είτε ως ξεχωριστά ηλεκτρονικά εξαρτήματα είτε ως τμήματα κάποιου ολοκληρωμένου κυκλώματος. Στην παρούσα υλοποίηση του σταθμού χρησιμοποιήθηκε τρανζίστορ τύπου MOSFET [55], καθαρά ως ψηφιακός διακόπτης για την υλοποίηση μειωμένης κατανάλωσης ενέργειας. Ο συγκεκριμένος τύπος MOSFET χρησιμοποιείται ευρέως στις κατασκευές Arduino. Αποτελείται από τρεις ακροδέκτες και η ονομασία του είναι IRF520 [56]. Παρακάτω ακολουθεί φωτογραφία απεικόνισης του. Η συνδεσμολογία του θα παρουσιαστεί στο τελικό σχεδιάγραμμα συνδεσμολογίας, διότι αποτελεί εξάρτημα συμπληρωματικού χαρακτήρα στην υλοποίηση.

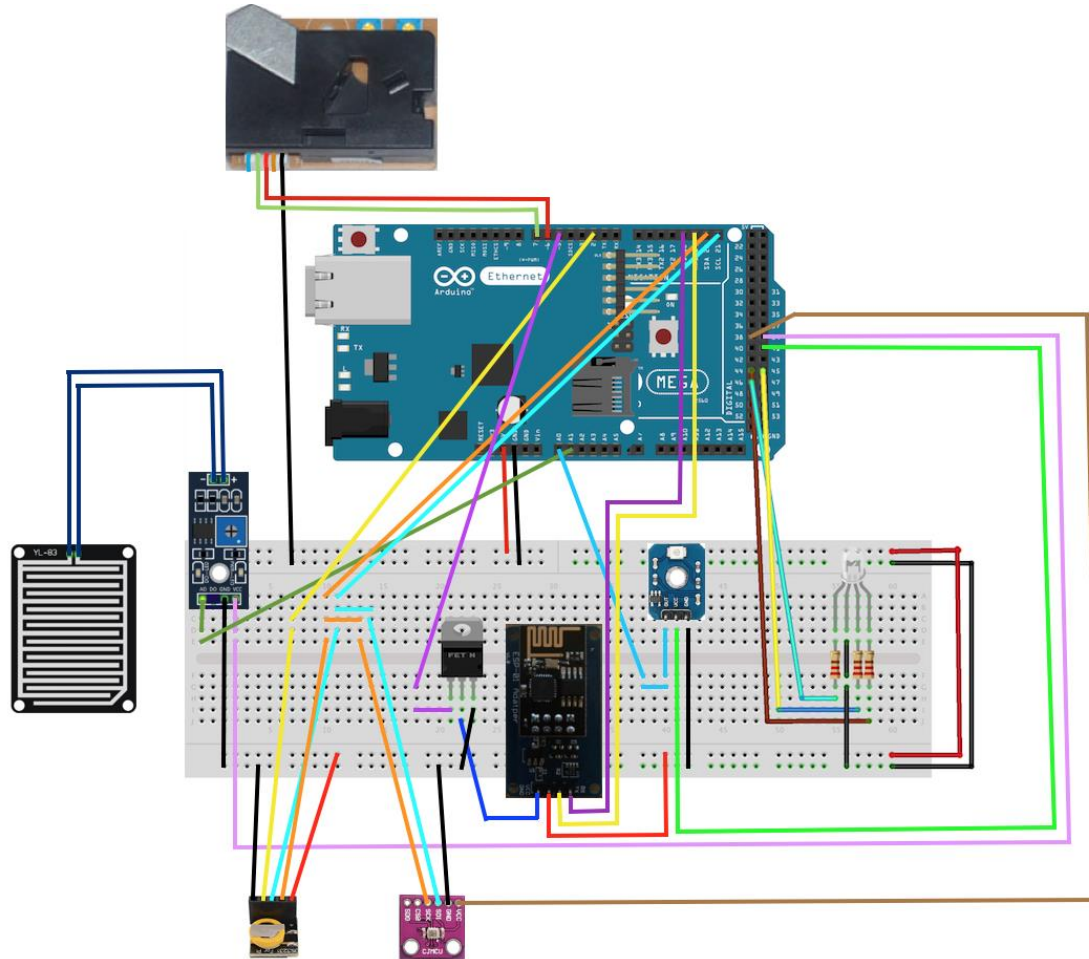


Εικόνα 40: IRF520 MOSFET

4.11 Τελικό κύκλωμα

Στο σημείο αυτό, έχει ολοκληρωθεί η καταγραφή των συσκευών και των ηλεκτρονικών στοιχείων που χρησιμοποιήθηκαν στην παρούσα υλοποίηση του ευφυούς περιβαλλοντικού σταθμού και έχει αναλυθεί η σύνδεση αυτών με το Arduino.

Ακολουθεί η παρουσίαση του σχεδίου συνδεσμολογίας όλων των εξαρτημάτων μεταξύ τους, καθώς και φωτογραφία της τελικής συσκευής που υλοποιήθηκε.

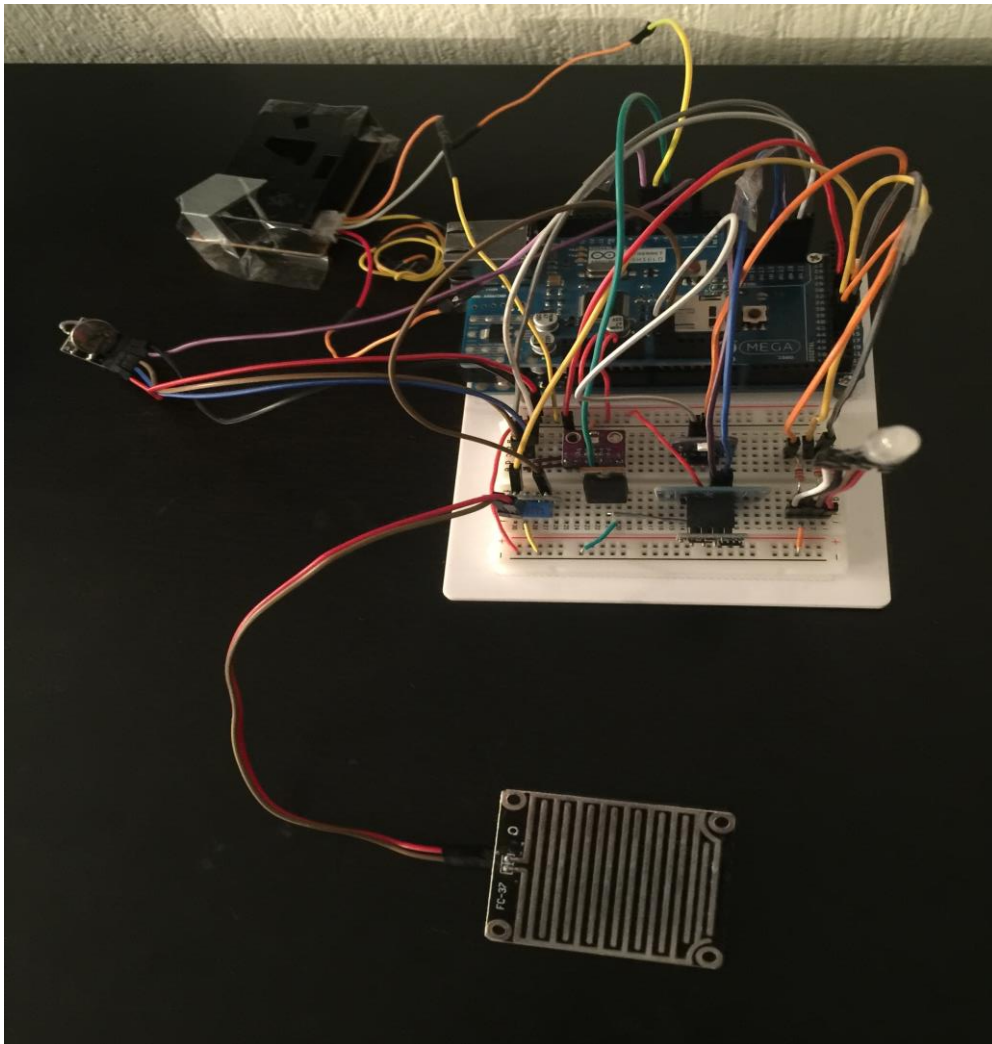


Εικόνα 41: Σχέδιο συνδεσμολογίας

Παρατηρώντας την παραπάνω υλοποίηση καταλήγουμε σε δύο σημαντικά συμπεράσματα:

- Δύο εξαρτήματα χρησιμοποιούν το ίδιο πρωτόκολλο επικοινωνίας (I2C) και αυτά είναι το DS3231 RTC και το GY-BME280. Ουσιαστικά είναι συνδεδεμένα παράλληλα και καταλήγουν σε κοινούς ακροδέκτες. Ωστόσο αυτό δεν θεωρείται πρόβλημα διότι, το συγκεκριμένο πρωτόκολλο επιτρέπει την ταυτόχρονη επικοινωνία και ανταλλαγή δεδομένων, μεταξύ συνδεδεμένων συσκευών και Arduino.

- Όλα τα εξαρτήματα εκτός του DS3231 RTC έχουν τους θετικούς ακροδέκτες τους συνδεδεμένους, σε αντίστοιχους ψηφιακούς ακροδέκτες στο Arduino και όχι σε θετικό ακροδέκτη παροχής ενέργειας. Ο λόγος έχει να κάνει καθαρά για την ελαχιστοποίηση της κατανάλωσης, καθώς οι ψηφιακοί ακροδέκτες μπορούν να προγραμματίζονται, για το πότε θα ενεργοποιούνται και θα επιτρέπουν την παροχή ενέργειας. Όσον αφορά το ESP8266, απαιτείται η σύνδεση του με θετικό ακροδέκτη παροχής ενέργειας διότι ο ψηφιακός ακροδέκτης δεν παρέχει την απαιτούμενη ποσότητα ρεύματος για να λειτουργήσει. Έτσι χρησιμοποιείται το MOSFET σαν ψηφιακός διακόπτης και το ESP8266 ενεργοποιείται ή απενεργοποιείται αντίστοιχα, απλώς μεταβάλλοντας την διαφορά δυναμικού στο MOSFET μέσω ψηφιακού ακροδέκτη του Arduino.



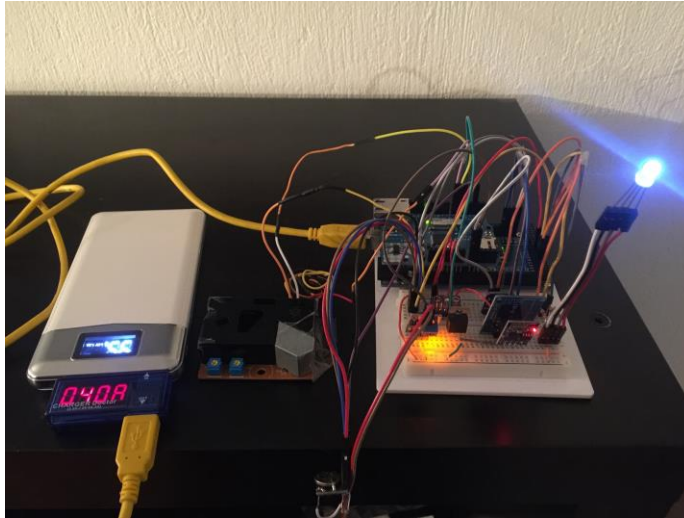
Εικόνα 42: Τελικό κύκλωμα

4.12 Στοιχεία κατανάλωσης ενέργειας

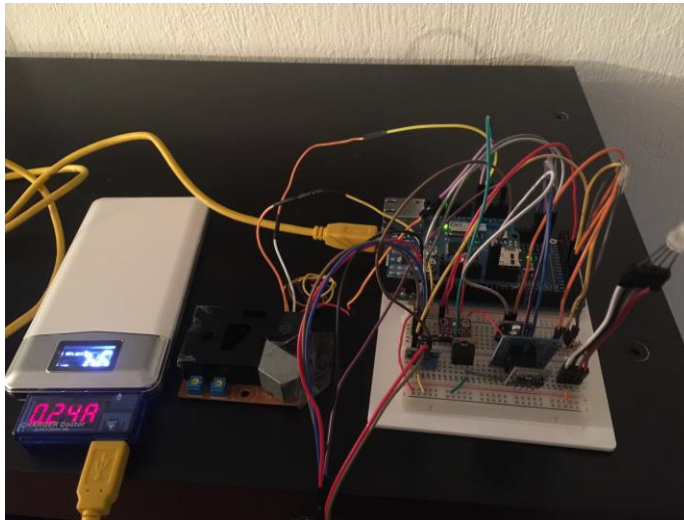
Στην υποενότητα αυτή θα δοθούν στοιχεία για την κατανάλωση ενέργειας και θα υπολογιστεί το ποσοστό εξοικονόμησης. Σε κατάσταση λειτουργίας το Arduino καταναλώνει συνολικά ~400 μιλιαμπέρ (milliamps). Σε κατάσταση μειωμένης ενεργειακής κατανάλωσης (επιτυγχάνεται με την χρήση βιβλιοθήκης), χωρίς όμως την διακοπή παροχής ενέργειας προς τους αισθητήρες η κατανάλωση είναι ~350 μιλιαμπέρ (μείωση ~50 μιλιαμπέρ). Με την διακοπή της παροχής ενέργειας προς τους αισθητήρες (τωρινή συσκευή), η κατανάλωση πέφτει στα ~200-240 μιλιαμπέρ. Το ποσό αυτό διαφέρει ανάλογα το είδος τροφοδοσίας. Ουσιαστικά υπάρχει μία εξοικονόμηση ενέργειας της τάξης του ~45-50%. Επίσης μπορεί να υπολογιστεί η κατανάλωση ανάλογα με τον χρόνο ως εξής. Θεωρώντας ότι ο τυπικός κύκλος λειτουργίας της συσκευής έχει υπολογιστεί ότι καταλαμβάνει 2 - 2.5 λεπτά και η κατάσταση ύπνου 27.5 - 28 λεπτά, αφού μετατρέψουμε τα λεπτά αυτά σε μιλιδευτερόλεπτα (12×10^4 και 168×10^4 αντίστοιχα) και τα πολλαπλασιάσουμε με τις αντίστοιχες καταναλώσεις στον ίδιο χρόνο, έχουμε 48×10^6 και 336×10^6 μιλιαμπέρ ανά μιλιδευτερόλεπτο (milliamps/sec). Διαιρώντας την κάθε τιμή με τον αριθμό 36×10^5 (μετατροπή ώρας σε μιλιδευτερόλεπτα) έχουμε αντίστοιχα την κατανάλωση ανά ώρα η οποία είναι 13.3 και 93.3 μιλιαμπέρ ανά ώρα (mAh). Τέλος με την πρόσθεση τους βρίσκουμε την συνολική κατανάλωση της συσκευής σε τριάντα λεπτά η οποία είναι 106.6 mAh και σε μία ώρα που είναι 213.2 mAh. Με αυτά τα δεδομένα μπορούμε πλέον να υπολογίσουμε θεωρητικά τον χρόνο που θα παραμείνει ενεργή η συσκευή.

Για παράδειγμα, αν η συσκευή τροφοδοτείται με ένα powerbank χωρητικότητας 30.000 mAh τότε με διαίρεση της τιμής κατανάλωσης ανά ώρα υπολογίζουμε ότι οι ώρες ανέρχονται σε ~141. Διαιρώντας με το 24 (ώρες ανά ημέρα) βρίσκουμε ότι η συσκευή θα παραμείνει ενεργή για ~6 ημέρες.

Παρακάτω δίνονται φωτογραφίες από την μέτρηση κατανάλωσης ενέργειας της συσκευής. Σημειώνεται ότι η συσκευή είναι συνδεδεμένη με powerbank χωρητικότητας 20.000 mAh και η μέτρηση πραγματοποιείται από ειδικό μετρητή-αντάπτορα.



Εικόνα 43: Μέτρηση κατανάλωσης ενέργειας συσκευής σε ενεργή κατάσταση



Εικόνα 44: Μέτρηση κατανάλωσης ενέργειας συσκευής σε κατάσταση ύπνου

4.13 Υπολογισμός κόστους εξαρτημάτων

Στην σύντομη αυτή υποενότητα, γίνεται μία καταγραφή των συσκευών και των ηλεκτρονικών στοιχείων που χρησιμοποιήθηκαν για την κατασκευή της συσκευής και παρατίθενται στοιχεία για την τιμή και την ποσότητα τους στον πίνακα που ακολουθεί. Οι τιμές που αναφέρονται παρακάτω, αφορούν τις γνήσιες πλακέτες.

Όνομα	Ποσότητα	Τιμή μονάδας (€)
Arduino Mega 2560 Rev3	1	20.32
Ethernet Shield Rev3	1	20.00
ESP8266 WiFi module	1	1.25
ESP8266 adapter	1	0.74
GY-BME280	1	2.00
DSM501A	1	4.39
Raindrop Sensor	1	0.52
UV Sensor	1	3.00
DS3231 RTC	1	1.13
RGB Led	1	0.35
IRF520	1	0.16
Αντιστάσεις	3	0.10
Powerbank 20.000 mAh	1	20.00
RTC μπαταρία 3V	1	0.10
καλώδια	30	0.50
Συνολικό κόστος: 74.56€		

Πίνακας 13: Λίστα και κοστολόγιο υλικών

Τέλος, στην παρακάτω φωτογραφία παρουσιάζεται η τελική συσκευή, αφού τοποθετήθηκε σε ειδικά διαμορφωμένο πλαστικό κουτί.



Εικόνα 45: Τελική συσκευή

4.14 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό, έγινε αναλυτική περιγραφή του υλικού μέρους του συστήματος. Παρουσιάστηκαν όλες οι πλακέτες, τα εξαρτήματα και οι αισθητήρες που χρησιμοποιήθηκαν. Έγινε ανάλυση των τεχνικών τους χαρακτηριστικών και αναπτύχθηκε η συνδεσμολογία με το Arduino. Παρουσιάστηκε το τελικό σχέδιο, έγινε επεξήγηση ορισμένων σημείων, μελετήθηκε η κατανάλωση ενέργειας και έγινε καταγραφή του κόστους των υλικών. Στο επόμενο κεφάλαιο ακολουθεί, η ανάλυση του λογισμικού μέρους του συστήματος.

Κεφάλαιο 5ο – Λογισμικό μέρος συστήματος

Όπως έχει ήδη αναφερθεί στην σύνοψη κεφαλαίων, το κεντρικό θέμα αυτού του κεφαλαίου αποτελεί το λογισμικό μέρος του συστήματος, με το οποίο και ολοκληρώνεται. Αναλύονται οι λειτουργίες και οι διεπαφές χρήστη της διαδικτυακής εφαρμογής, καθώς και οι περιπτώσεις χρήσης που παρουσιάζονται, ανάλογα την ιδιότητα του. Επίσης θα δοθεί μία περιγραφή για την συνολική λειτουργία του Arduino, την ροή του προγράμματος που ακολουθεί, τις ενέργειες που πραγματοποιεί ανάλογα με τις συνθήκες που αντιμετωπίζει και γενικότερα, θα αναλυθεί η "λογική" που έχει αναπτυχθεί από πίσω και το καθιστά αυτόνομο λειτουργικά και ευφύες.

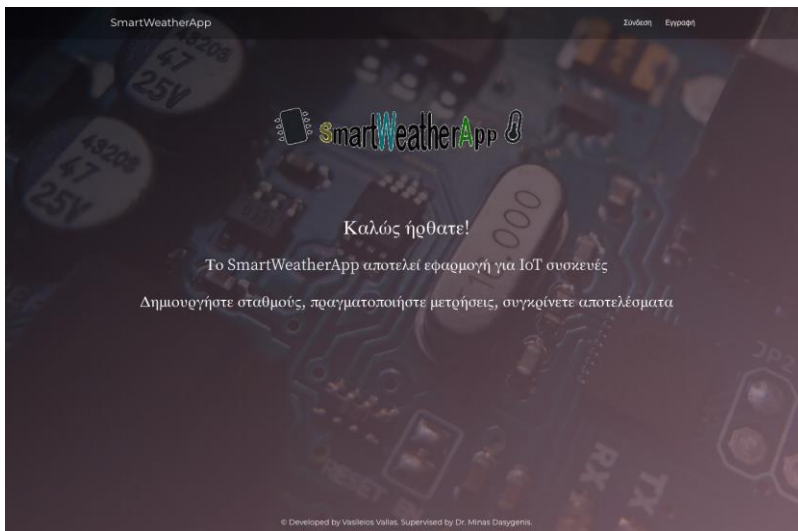
5.1 Ανάλυση λειτουργιών διαδικτυακής εφαρμογής και διεπαφές χρήστη

Ο σχεδιασμός της διεπαφής χρήστη σε ένα σύστημα αποτελεί ένα από τα βασικότερα στοιχεία που θα καθορίσουν την επιτυχία μίας εφαρμογής. Μία δύσχρηστη διεπαφή μπορεί να κάνει τον χρήστη να απορρίψει την εφαρμογή, παρά το γεγονός ότι αυτή μπορεί να λειτουργεί σωστά. Για αυτό τον λόγο η εφαρμογή παρέχει όλους τους απαραίτητους μηχανισμούς που θα συντελέσουν στην καλύτερη και ευκολότερη διαχείριση της από τους χρήστες. Παράλληλα με την διεπαφή, η εφαρμογή θα πρέπει να παρέχει όλες τις απαραίτητες λειτουργίες για τις οποίες προορίζεται, δίνοντας τον χρήστη πληθώρα επιλογών για ένα μεγάλο εύρος λειτουργιών. Στις υποενότητες που ακολουθούν παρουσιάζονται όλες οι διεπαφές χρήστη και οι λειτουργίες τους. Αξίζει να σημειωθούν και να αναλυθούν σε αυτό το σημείο ορισμένα στοιχεία της εφαρμογής, ώστε να μην αναφέρονται ξανά σε κάθε διεπαφή. Αρχικά, η εφαρμογή ακούει και πραγματοποιεί ότι αλλαγή συμβαίνει σε πραγματικό χρόνο χάρις στην βοήθεια του Pusher API. Αυτό πρακτικά σημαίνει πως κάθε χρήστης που βρίσκεται εντός της εφαρμογής, ακούει στο παρασκήνιο πιθανά συμβάντα (events) που μπορεί να λάβουν χώρα (δημιουργία μίας κατηγορίας ή μίας συλλογής μετρήσεων κτλ.). Έτσι με την πραγματοποίηση ενός συμβάντος, αυτομάτως γίνεται ανανέωση της διεπαφής χρήστη σε επιλεγμένα σημεία που επηρεάζονται από την ενέργεια. Έτσι ο χρήστης θα δει αυτομάτως την ανανεωμένη

πληροφορία καθώς και μία ειδοποίηση και δεν θα χρειαστεί να πραγματοποιήσει ανακατεύθυνση της σελίδας. Ένα εξίσου σημαντικό στοιχείο αφορά την επικύρωση των δεδομένων (validation). Σε όλη την εφαρμογή τα δεδομένα που εισάγονται στις φόρμες, επικυρώνονται δύο φορές, μία στο frontend κομμάτι (React) και μία στο backend (server). Έτσι δεν υπάρχει τρόπος να προσπελαστούν και οι δύο οι έλεγχοι και να περαστούν δεδομένα που δεν ελέγχθηκαν. Επίσης, οι έλεγχοι που πραγματοποιούνται στο κομμάτι του server, είναι πολύ πιο αυστηροί και σε περίπτωση εύρεσης σφάλματος, οδηγούν σχεδόν πάντα σε αποσύνδεση του χρήστη. Αυτή η υλοποίηση σχεδιάστηκε στοχευμένα έτσι διότι, υπάρχουν λογισμικά δημιουργίας http requests σε εφαρμογές και servers (για παράδειγμα το postman), παρακάμπτοντας το frontend κομμάτι της εφαρμογής. Τέλος, το Redux, αποτελεί το μέρος όπου αποθηκεύονται τα πιο σημαντικά δεδομένα κατά την σύνδεση του χρήστη με την εφαρμογή, όπως επίσης αποτελεί και το κέντρο πληροφοριών κάθε κατάστασης και διεπαφής. Αυτό που επιτυγχάνεται είναι, να παραμένει ο φόρτος εργασίας του server σε πολύ χαμηλό επίπεδο. Επίσης λόγω της συγκεκριμένης υλοποίησης, ο server στις περισσότερες αιτήσεις που δέχεται, επιστρέφει μία απλή πληροφορία και όχι πολύπλοκες μετατροπές δεδομένων από την βάση, ούτε αρχεία προβολής (view pages). Αυτό έχει σαν αποτέλεσμα, πολύ γρήγορη απόκριση της εφαρμογής ακόμα και σε πιθανώς, μεγάλο εύρος ενεργών χρηστών.

5.1.1 Αρχική σελίδα

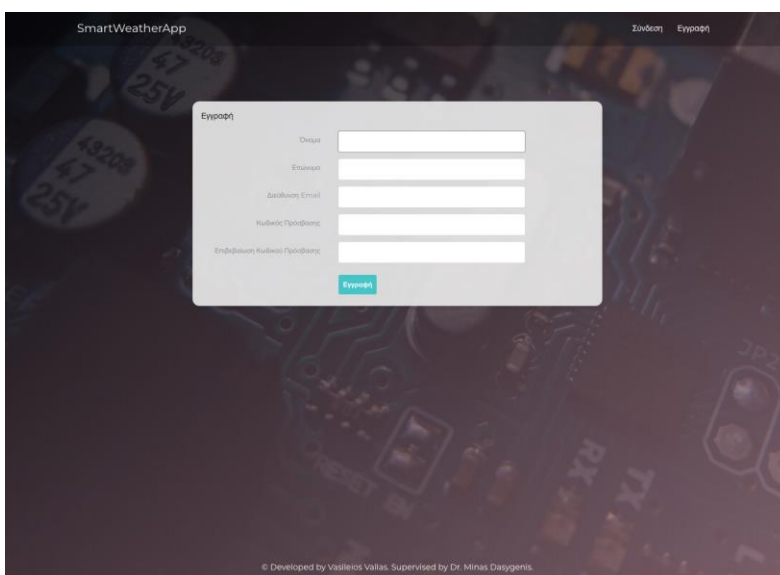
Η αρχική είναι η πρώτη σελίδα της εφαρμογής, την οποία αντικρίζει κάθε χρήστης που επιθυμεί να αποκτήσει πρόσβαση σε αυτή. Εδώ εμφανίζονται γενικές πληροφορίες σχετικά με το είδος της εφαρμογής. Ένα αρκετά σημαντικό στοιχείο είναι πως, η συγκεκριμένη σελίδα παίζει πολύ σημαντικό ρόλο στην πρώτη εντύπωση που θα αφήσει στον χρήστη. Έτσι, θα πρέπει να είναι όμορφη στιλιστικά και η σχεδίαση της να "προκαλεί" την παρατηρητικότητα του χρήστη και παράλληλα να τον αφήνει με μία περιέργεια, ώστε να συνεχίσει την πλοήγηση του. Στο πάνω μέρος της σελίδας υπάρχει μία μεγάλη γραμμή που ονομάζεται μενού πλοήγησης. Εκεί, βρίσκονται οι σύνδεσμοι για την εγγραφή και την σύνδεση του χρήστη στην εφαρμογή.



Εικόνα 46: Αρχική σελίδα

5.1.2 Εγγραφή, σύνδεση, αποσύνδεση, διαδικασία επιβεβαίωσης χρήστη και λειτουργία ανάκτησης κωδικού πρόσβασης

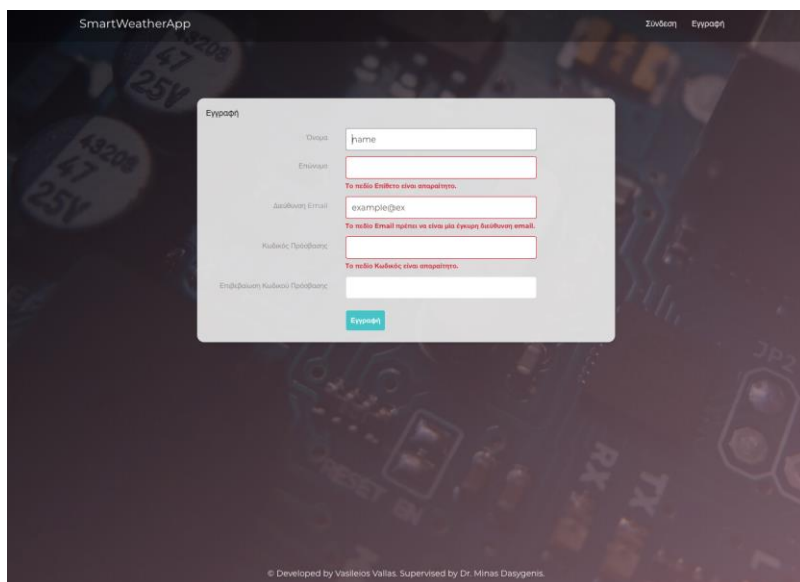
Όπως έχουμε αναφέρει και σε παραπάνω σημεία της διπλωματικής εργασίας, η διαδικτυακή εφαρμογή που υλοποιήθηκε, δεν υποστηρίζει μη εξουσιοδοτημένους χρήστες. Πρακτικά αυτό σημαίνει πως για να χρησιμοποιήσει κάποιος την εφαρμογή, θα πρέπει πρώτα να εγγραφεί σε αυτήν.



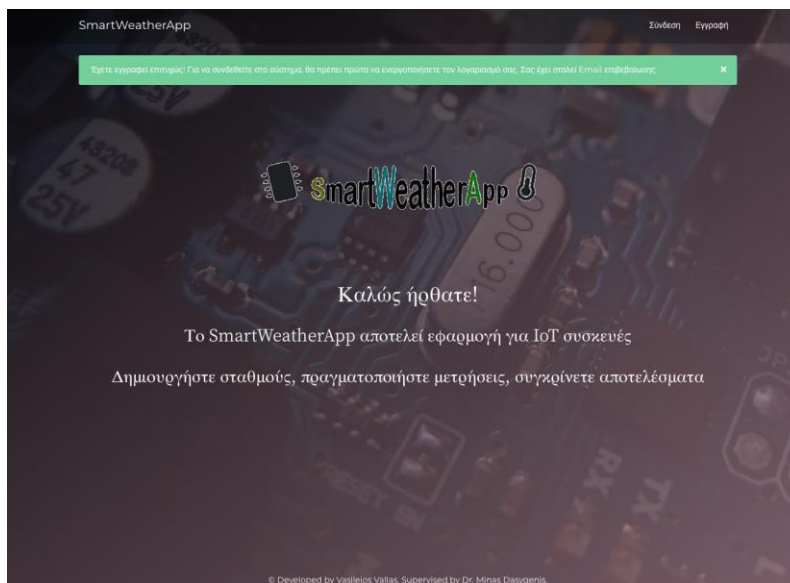
Εικόνα 47: Σελίδα εγγραφής

Η σελίδα εγγραφής περιέχει μία φόρμα, η οποία ζητάει να συμπληρωθούν τα εξής υποχρεωτικά πεδία: όνομα, επίθετο, διεύθυνση email, κωδικός πρόσβασης (password) και επιβεβαίωση κωδικού.

Με την υποβολή της φόρμας, αν κάποιο από τα πεδία δεν είναι συμπληρωμένο ή έχει συμπληρωθεί με διαφορετική μορφή από τη ζητούμενη (πχ λανθασμένη μορφή email), τότε ο χρήστης ενημερώνεται με κατάλληλα μηνύματα σφάλματος. Σε περίπτωση επιτυχημένης εγγραφής, ο χρήστης δέχεται ενημέρωση για επιτυχία εγγραφής και παράλληλα του αποστέλλεται email επιβεβαίωσης, στην διεύθυνση email που δήλωσε κατά την εγγραφή του. Η λειτουργία αυτή συναντάται αρκετά συχνά σε διαδικτυακές εφαρμογές καθώς επιβεβαιώνεται η ταυτοπροσωπία του χρήστη. Αν ο χρήστης προσπαθήσει να εισέλθει στην εφαρμογή προτού πραγματοποιήσει την επιβεβαίωση, τότε δέχεται κατάλληλο μήνυμα. Αξίζει να σημειωθεί πως, η οντότητα που πραγματοποιεί την εγγραφή της στο σύστημα, δηλώνεται με την ιδιότητα του απλού χρήστη.



Εικόνα 48: Μηνύματα σφάλματος αποτυχημένης προσπάθειας εγγραφής



Εικόνα 49: Μήνυμα επιτυχημένης εγγραφής

Ευχαριστούμε για την εγγραφή σας στην σελίδα **billyval.com!**

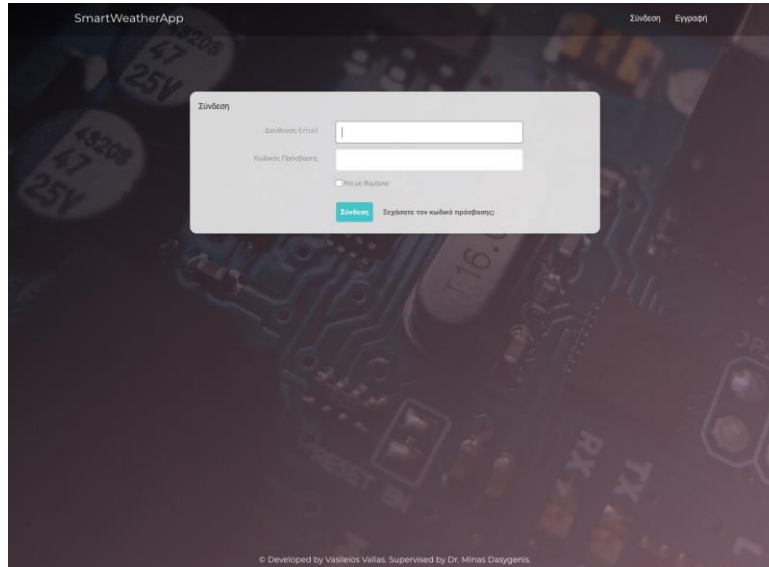
Παρακαλώ επιβεβαιώστε ότι το Email είναι σωστό για να προχωρήσετε. Πατήστε στον σύνδεσμο παρακάτω για να ξεκινήσετε.

[Επιβεβαίωση Διεύθυνσης Email](#)

Αυτό το Email στάλθηκε απο το [billyval.com](#)

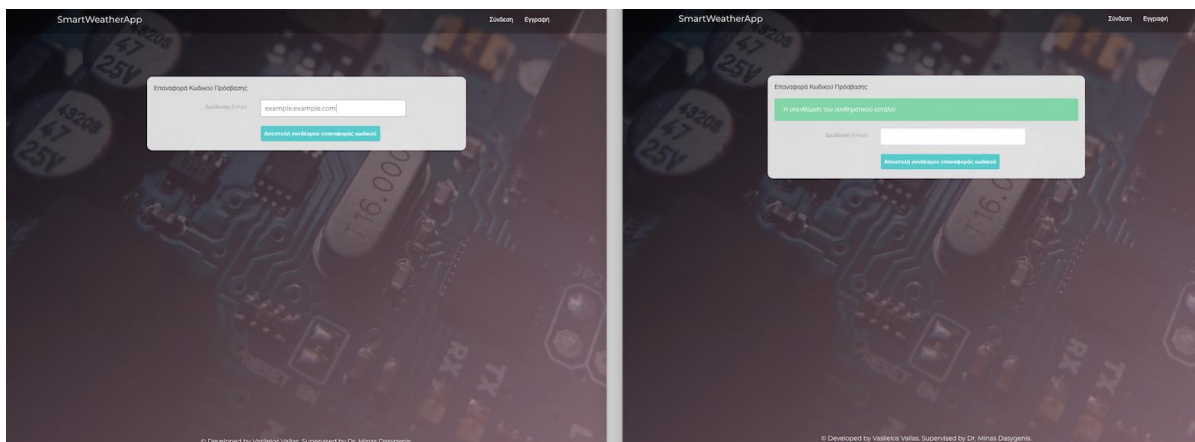
Εικόνα 50: Email επιβεβαίωσης

Με την επιλογή του συνδέσμου επιβεβαίωσης, ο χρήστης μεταφέρεται στην κεντρική σελίδα της εφαρμογής και του εμφανίζεται κατάλληλο μήνυμα επιτυχημένης επιβεβαίωσης. Έτσι είναι έτοιμος πλέον να συνδεθεί στην εφαρμογή. Στην σελίδα σύνδεσης, συμπληρώνει το email καθώς και τον κωδικό του. Σε περίπτωση επιτυχημένης εξακρίβωσης στοιχείων, οδηγείται σε κεντρικό περιβάλλον της εφαρμογής για εξουσιοδοτημένους χρήστες. Σε περίπτωση σφάλματος δέχεται κατάλληλο μήνυμα.



Εικόνα 51: Σελίδα σύνδεσης χρήστη

Επίσης παρέχεται η δυνατότητα στον χρήστη να ανακτήσει τον κωδικό πρόσβασης σε περίπτωση απώλειας του. Στην σελίδα σύνδεσης, υπάρχει σύνδεσμος, που οδηγεί στην σελίδα ανάκτησης κωδικού. Εκεί ο χρήστης συμπληρώνει σε φόρμα την διεύθυνση email, που έχει δηλώσει κατά την εγγραφή του και του αποστέλλεται email με σύνδεσμο επαναφοράς κωδικού. Με την επιλογή αυτού του συνδέσμου, ο χρήστης οδηγείται σε περιβάλλον της εφαρμογής, όπου του ζητείται να εισάγει τον καινούργιο κωδικό πρόσβασης καθώς και την επιβεβαίωση αυτού. Με τον επιτυχή έλεγχο των πεδίων, ο κωδικός πρόσβασης ανανεώνεται με παράλληλη ενημέρωση του χρήστη, αλλιώς σε περίπτωση σφάλματος εμφανίζεται κατάλληλο μήνυμα.



Εικόνα 52: Σελίδα ανάκτησης κωδικού μαζί με επιτυχημένη προσπάθεια

SmartWeatherApp

Χαίρεται!

Έχετε δεχτεί αυτό το Email, διότι μας ζητήθηκε επαναφορά του κωδικού πρόσβασης απο τον λογαριασμό σας.

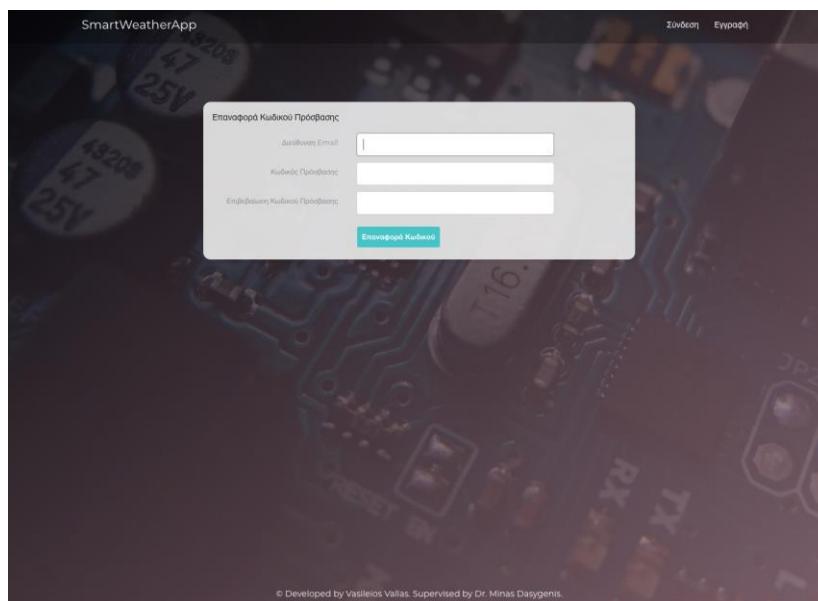
[Επαναφορά Κωδικού](#)

Εάν δεν έχετε ζητήσει επαναφορά κωδικού πρόσβασης, δεν απαιτείται καμία περαιτέρω ενέργεια.

Χαιρετισμοί, smartWeatherApp

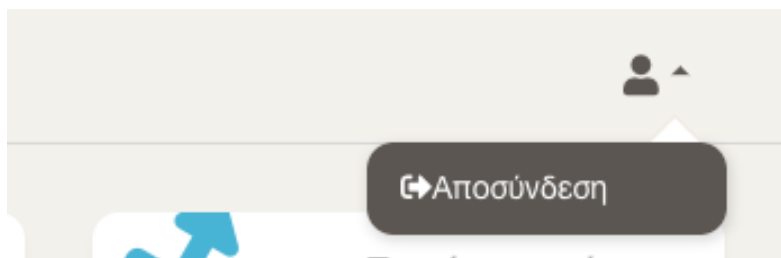
If you're having trouble clicking the "Επαναφορά Κωδικού" button, copy and paste the URL below into your web browser:
http://smart_test/password/reset/#4361d3ecb465130b8514bbdfdf1553463e9404433e651896f443aea7f2944

Εικόνα 53: Email συνδέσμου επαναφοράς κωδικού



Εικόνα 54: Περιβάλλον δημιουργίας καινούργιου κωδικού

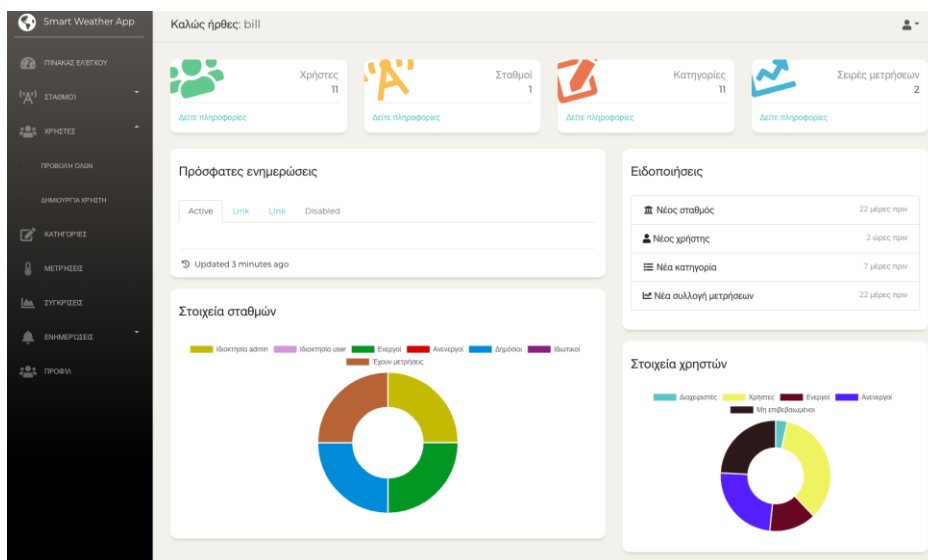
Τέλος για την αποσύνδεση του χρήστη από την εφαρμογή, αρκεί το πάτημα του συνδέσμου αποσύνδεσης στην πάνω δεξιά γωνία της εφαρμογής. Έπειτα ο χρήστης οδηγείται στην αρχική κεντρική σελίδα.



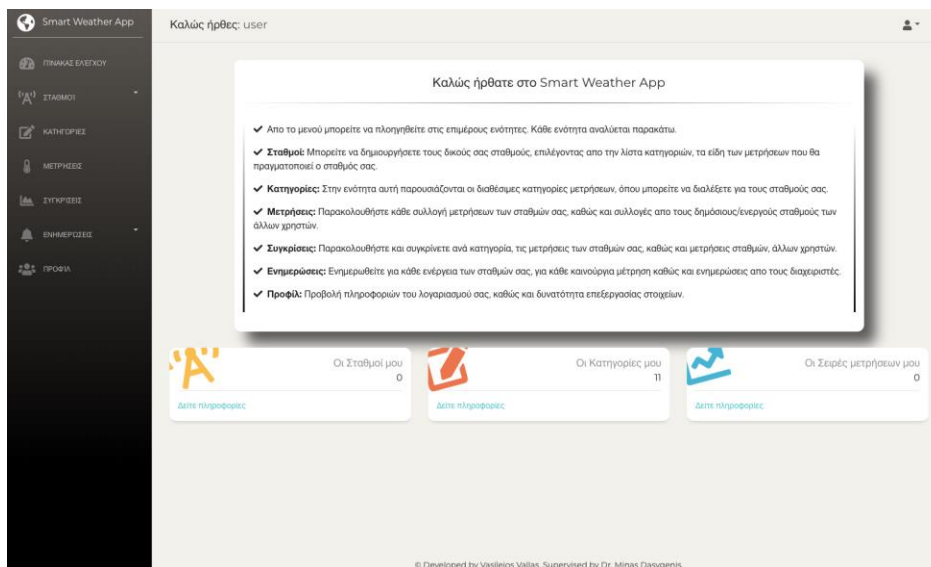
Εικόνα 55: Σύνδεσμος αποσύνδεσης χρήστη

5.1.3 Κεντρικό περιβάλλον εξουσιοδοτημένου χρήστη (διαχειριστής - απλός χρήστης)

Η κεντρική σελίδα του εξουσιοδοτημένου χρήστη, μέσω του οποίου μπορεί να περιηγηθεί σε όλες τις σελίδες του συστήματος και να εκτελέσει όλες τις λειτουργίες, φαίνεται στις παρακάτω εικόνες. Το περιβάλλον του διαχειριστή διαφέρει κατά πολύ με το αντίστοιχο του απλού χρήστη καθώς στον διαχειριστή εμφανίζονται αρκετές πληροφορίες και στατιστικά στοιχεία (και σε μορφή γραφημάτων) σχετικά με την εφαρμογή. Αντίστοιχα στον απλό χρήστη, εμφανίζονται βοηθητικές πληροφορίες για την καλύτερη κατανόηση των λειτουργιών της σελίδας, καθώς και ενδείξεις για την ευκολότερη πλοήγηση του σε αυτήν.



Εικόνα 56: Περιβάλλον διαχειριστή



Εικόνα 57: Περιβάλλον απλού χρήστη

Πιο συγκεκριμένα, ο διαχειριστής, μπορεί να πλοηγηθεί με την βοήθεια του πλάγιου μενού που βρίσκεται αριστερά σε σελίδες για:

- διαχείριση σταθμών
- διαχείριση χρηστών
- διαχείριση κατηγοριών μετρήσεων
- προβολή και επεξεργασία συλλογών μετρήσεων σταθμών
- προβολή και σύγκριση (γραφικά) συλλογών μετρήσεων σταθμών
- προβολή ενημερώσεων
- προβολή και επεξεργασία λογαριασμού χρήστη

Επίσης τα κεντρικά πάνελ, αποτελούν και αυτά σύνδεσμοι που οδηγούν, σε σελίδες που προαναφέρθηκαν παραπάνω.

Τέλος, ο απλός χρήστης, μπορεί και αυτός να πλοηγηθεί με την βοήθεια του πλάγιου μενού που βρίσκεται αριστερά σε σελίδες για:

- διαχείριση δικών του σταθμών
- προβολή κατηγοριών μετρήσεων

- προβολή και επεξεργασία συλλογών μετρήσεων δικών του σταθμών ή προβολή συλλογών μετρήσεων σταθμών άλλων χρηστών, που έχουν επιλεγεί ως δημόσιοι και ενεργοί
- προβολή και σύγκριση (γραφικά) συλλογών μετρήσεων δικών του σταθμών ή σταθμών άλλων χρηστών, που έχουν επιλεγεί ως δημόσιοι και ενεργοί
- προβολή ενημερώσεων
- προβολή και επεξεργασία λογαριασμού χρήστη

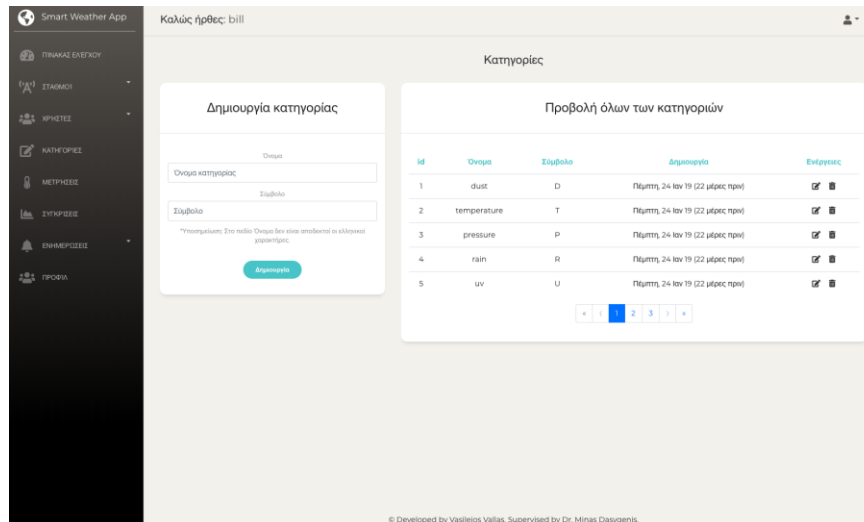
Επιπρόσθετα και στο περιβάλλον του απλού χρήστη, τα πάνελ λειτουργούν και ως σύνδεσμοι για ορισμένες σελίδες.

Αξίζει να αναφερθεί, πως για την χειροκίνητη μετάβαση στην κεντρική σελίδα πραγματοποιείται και μέσω του συνδέσμου που βρίσκεται στο αριστερό μενού, στην επιλογή "πίνακας ελέγχου".

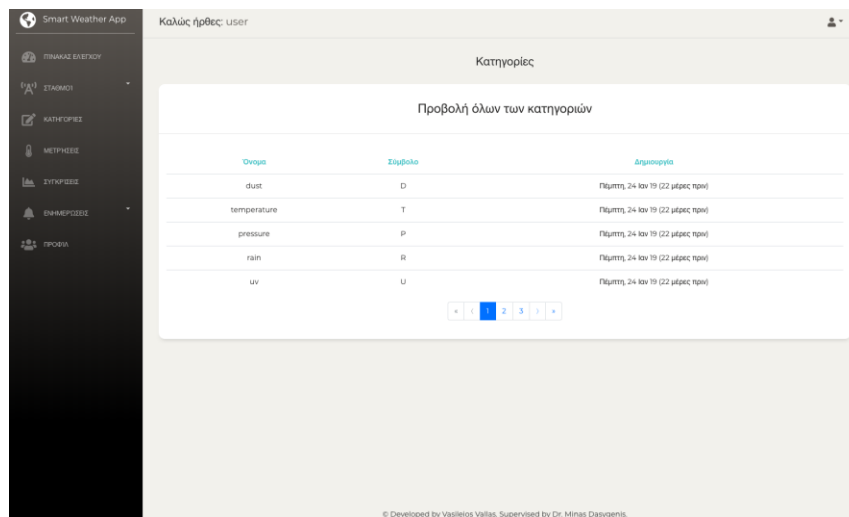
5.1.4 Σελίδα κατηγοριών (διαχειριστής - απλός χρήστης)

Η συγκεκριμένη σελίδα προορίζεται για την παρουσίαση και πληροφόρηση του εξουσιοδοτημένου χρήστη, ως προς τις διαθέσιμες κατηγορίες μετρήσεων, τις οποίες μπορεί να επιλέξει για τους σταθμούς του. Για παράδειγμα εάν ένας πραγματικός σταθμός έχει υλοποιηθεί προγραμματιστικά, στο να καταγράψει μετρήσεις για θερμοκρασία και υγρασία, αυτές οι δύο κατηγορίες θα πρέπει πρώτον να υπάρχουν στην λίστα και δεύτερον, να επιλεγούν κατά την δημιουργία του αντίστοιχου "εικονικού" σταθμού από τον ιδιοκτήτη, έτσι ώστε να υλοποιηθεί σωστά η διασύνδεση. Ο σύνδεσμος για την μετάβαση στην σελίδα κατηγοριών βρίσκεται στο αριστερό μενού, στην επιλογή "κατηγορίες". Είναι πολύ σημαντικό για την λειτουργία της εφαρμογής και συλλογικά για όλη την υλοποίηση να υπάρχει μία μεγάλη λίστα κατηγοριών. Οι υπεύθυνοι για την συμπλήρωση της λίστας είναι οι διαχειριστές μαζί με τον δημιουργό. Αυτοί έχουν και τον πλήρη έλεγχο δημιουργίας, επεξεργασίας και διαγραφής σε αντίθεση με τον απλό χρήστη ο οποίος έχει μόνο δικαίωμα προβολής. Η λίστα των κατηγοριών αποτελείται από έναν html πίνακα, ο οποίος σε κάθε γραμμή αναφέρει το όνομα, το σύμβολο και την ημερομηνία δημιουργίας κάθε κατηγορίας. Κάθε κατηγορία είναι μοναδική, που σημαίνει ότι δεν υπάρχει διπλότυπο

όνομα ή σύμβολο κατηγορίας. Έχει δοθεί ένα όριο πέντε κατηγοριών ανά σελίδα. Αν ξεπεραστεί ο αριθμός τότε δημιουργείται σελιδοποίηση (pagination).



Εικόνα 58: Σελίδα κατηγοριών διαχειριστή



Εικόνα 59: Σελίδα κατηγοριών απλού χρήστη

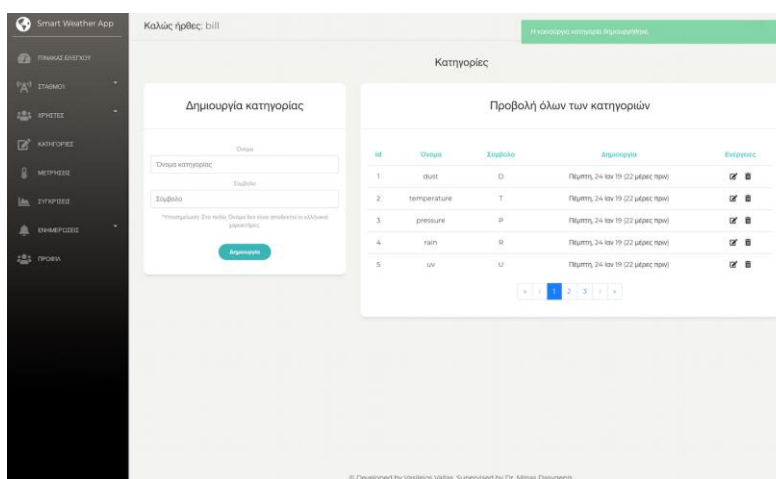
Δημιουργία κατηγορίας: Στο αριστερό κεντρικό πάνελ του διαχειριστή υπάρχει φόρμα για δημιουργία κατηγορίας. Εδώ συμπληρώνονται σε inputs τα εξής στοιχεία:

- **όνομα:** Όνομα κατηγορίας. Πρέπει να είναι μοναδικό στην εφαρμογή. Δεν είναι αποδεκτοί οι ελληνικοί χαρακτήρες, διότι θα προκύψει σοβαρό πρόβλημα


ταυτοποίησης των επιλεγμένων κατηγοριών των σταθμών, στην διαδικασία συλλογής μετρήσεων.

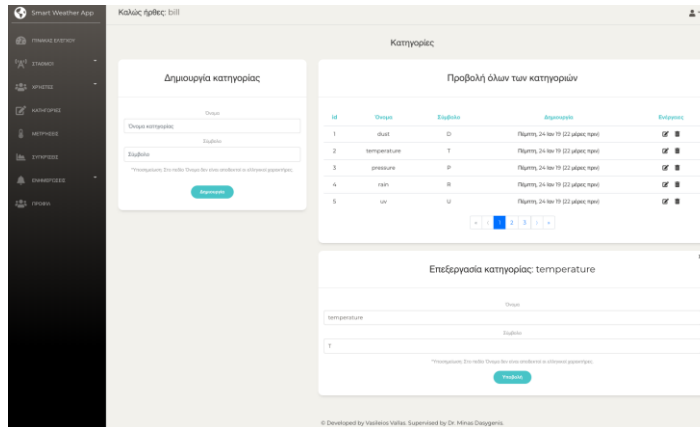
- **σύμβολο:** Σύμβολο κατηγορίας. Πρέπει να είναι μοναδικό στην εφαρμογή.

Με την συμπλήρωση των πεδίων και την επιλογή του κουμπιού "Δημιουργία", αφού ελεγχθεί η εγκυρότητα και μοναδικότητα των στοιχείων, δημιουργείται η καινούργια κατηγορία με παράλληλο μήνυμα ειδοποίησης ειδικού χρωματισμού, στο πάνω δεξί μέρος της εφαρμογής (tooltip). Σε περίπτωση σφάλματος εμφανίζεται κατάλληλο μήνυμα.




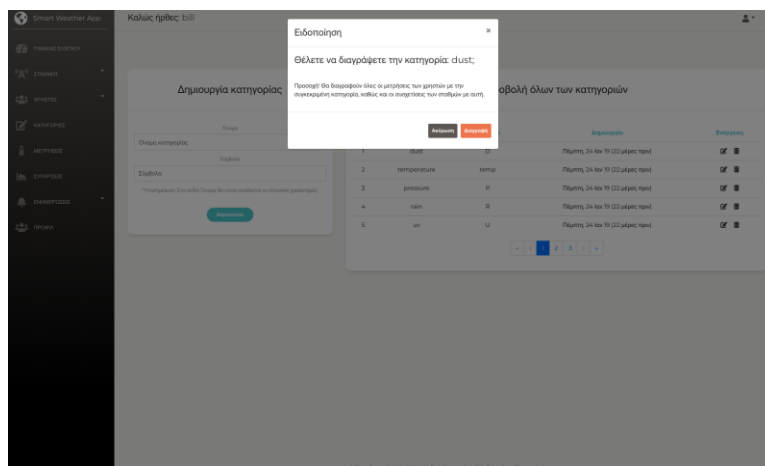
Εικόνα 60: Δημιουργία νέας κατηγορίας

Επεξεργασία κατηγορίας: Στο δεξί κεντρικό πάνελ του διαχειριστή, με την επιλογή του πρώτου εικονιδίου στην στήλη των ενεργειών , εμφανίζεται στο κάτω μέρος της σελίδας, ένα συμπληρωματικό πάνελ που χρησιμοποιείται για την ανανέωση της κατηγορίας. Εκεί εμφανίζονται οι υπάρχουσες πληροφορίες της κατηγορίας μέσα σε html inputs που επιδέχονται αλλαγές. Με το πάτημα του κουμπιού "Υποβολή" ελέγχονται τα δεδομένα που έχουν εισαχθεί και ανανεώνεται η κατηγορία, με την εμφάνιση κατάλληλου tooltip ή εμφανίζεται μήνυμα σφάλματος. Με την υποβολή των ίδιων στοιχείων με πριν εμφανίζεται κατάλληλη ειδοποίηση. Σημειώνεται πως, στις αλλαγές που συμβαίνουν σε πραγματικό χρόνο σε όλους τους ενεργούς χρήστες, είναι η ανανέωση της κατηγορίας σε όλες τις σελίδες στις οποίες εμφανίζεται.



Εικόνα 61: Πάνελ ανανέωσης κατηγορίας

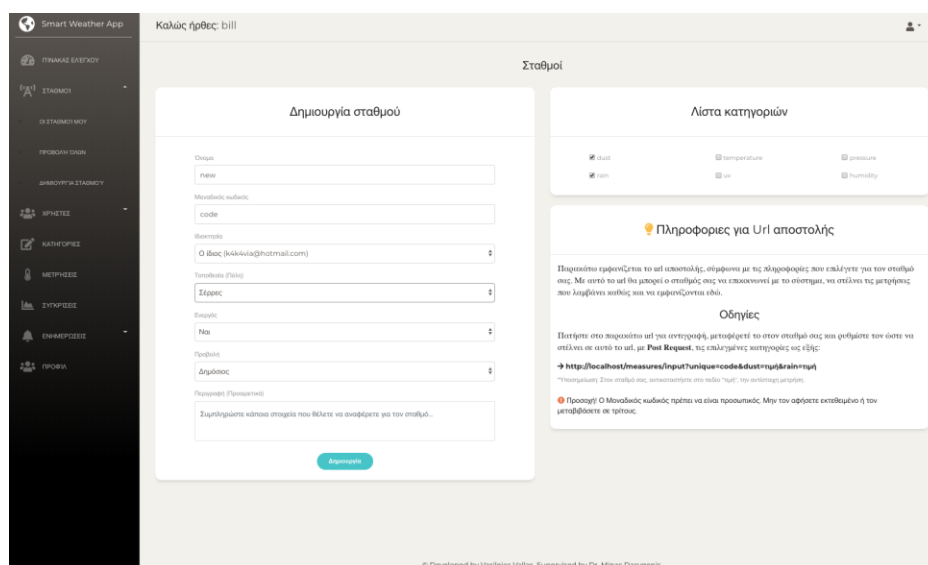
Διαγραφή κατηγορίας: Επίσης στο δεξί κεντρικό πάνελ του διαχειριστή, με την επιλογή του δεύτερου εικονιδίου στην στήλη των ενεργειών , εμφανίζεται πάνελ που υπερκαλύπτει την κεντρική οθόνη (modal), δίνοντας κάποιες έξτρα πληροφορίες και ζητάει την επιβεβαίωση της διαγραφής της κατηγορίας. Με το πάτημα του κουμπιού "Διαγραφή", διαγράφεται η κατηγορία, εμφανίζεται κατάλληλο tooltip και ειδοποιείται ο χρήστης για αναμενόμενη ανακατεύθυνση σελίδας. Το ίδιο συμβαίνει και στους ενεργούς χρήστες. Η ανακατεύθυνση σελίδας χρησιμεύει διότι, μία πιθανή διαγραφή κατηγορίας προκαλεί αλυσιδωτές αντιδράσεις στην βάση δεδομένων, με διαγραφές μετρήσεων και συσχετίσεων μεταξύ σταθμών. Έτσι προτιμήθηκε μία ανακατεύθυνση σελίδας από την παράλληλη χρήση πολλών events τα οποία και απαιτούν πολλές περιπτώσεις χρήσης. Τέλος, με το πάτημα του κουμπιού "Ακύρωση", ακυρώνεται η διαδικασία διαγραφής.



Εικόνα 62: Modal διαγραφής κατηγορίας

5.1.5 Σελίδα σταθμών (διαχειριστής - απλός χρήστης)

Η σελίδα αυτή αποτελεί ένα ενιαίο σύνολο υποσελίδων, που προορίζονται για το πιο σημαντικό στοιχείο της εφαρμογής που δεν είναι άλλο, από τους σταθμούς των χρηστών. Εδώ μπορεί ο εξουσιοδοτημένος χρήστης να πραγματοποιήσει όλες τις ενέργειες που σχετίζονται με τους σταθμούς του (δημιουργία, επεξεργασία, διαγραφή). Υπάρχει διαφορά στο σύνολο των υποσελίδων καθώς και στις δυνατότητες που προορίζονται, για τους διαχειριστές έναντι των απλών χρηστών. Ο απλός χρήστης έχει δυνατότητες δημιουργίας, επεξεργασίας και διαγραφής δικών του μόνο σταθμών. Σε αντίθεση ένας διαχειριστής εκτός από αυτές τις δυνατότητες που προαναφέρθηκαν, του παρέχονται ακόμα δυνατότητες για προβολή, επεξεργασία και διαγραφή όλων των σταθμών των χρηστών της εφαρμογής (απλών χρηστών και διαχειριστών), καθώς και δικαίωμα προβολής των σταθμών που ανήκουν στον δημιουργό. Σε αυτό το σημείο πρέπει να καταστεί σαφές, πως η επιλογή των διαχειριστών θα πρέπει να γίνει με σύνεση, αφενός από τον δημιουργό και αφετέρου, από τους υπόλοιπους διαχειριστές, καθώς ένας πιθανός διαχειριστής αποκτά σημαντική δύναμη έναντι των απλών χρηστών.



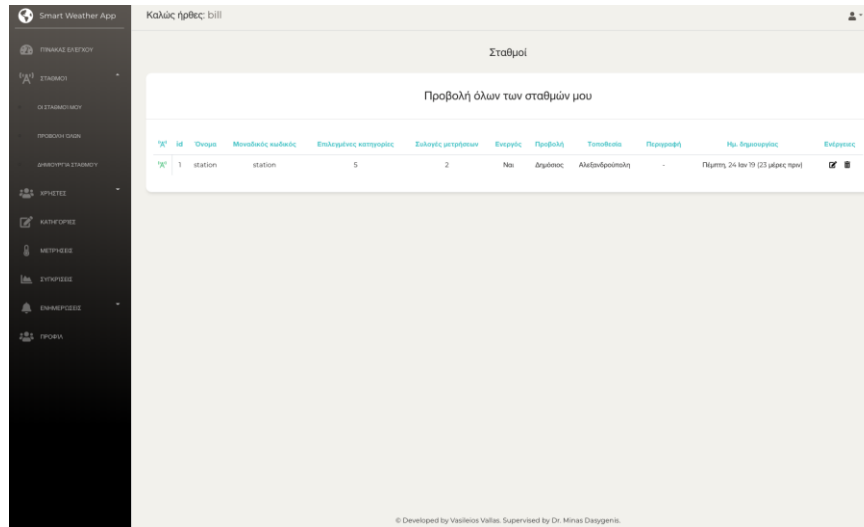
Εικόνα 63: Σελίδα δημιουργίας σταθμού

Δημιουργία σταθμού: Στην παραπάνω φωτογραφία εμφανίζεται η σελίδα δημιουργίας σταθμού. Το περιβάλλον είναι ίδιο και για τους διαχειριστές καθώς και για τους απλούς

χρήστες. Η μετάβαση στην συγκεκριμένη σελίδα πραγματοποιείται από το αριστερό μενού, με την επιλογή του συνδέσμου "δημιουργία σταθμού". Εδώ ο χρήστης συμπληρώνει στα inputs, τις εξής πληροφορίες για τον σταθμό:

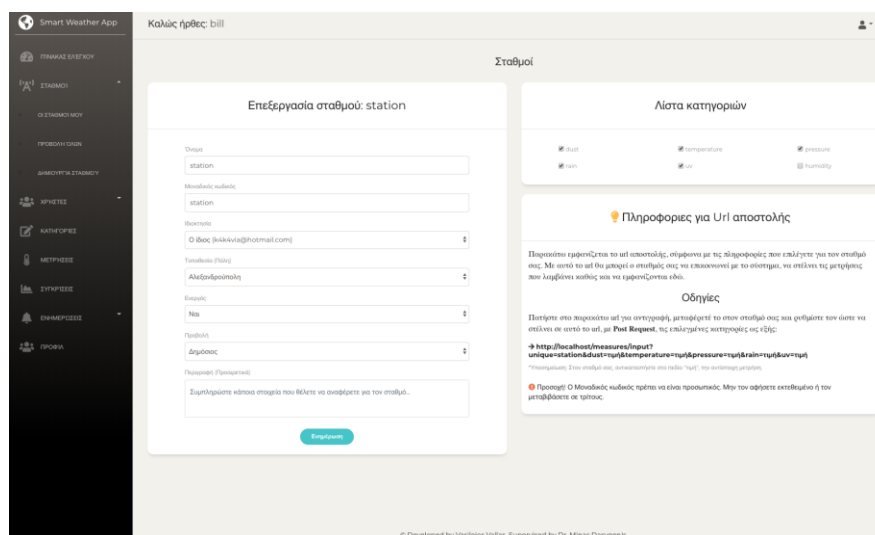
- **όνομα:** Ονομασία σταθμού. Πρέπει να είναι μοναδικό στην εφαρμογή.
- **μοναδικός κωδικός:** Κωδικός που χρησιμοποιείται στην αποστολή δεδομένων από τον πραγματικό σταθμό και βοηθάει στην ταυτοποίηση του με τον εικονικό της εφαρμογής. Πρέπει να είναι μοναδικός και δεν επιδέχεται ελληνικούς χαρακτήρες για τον ίδιο λόγο που αναφέρθηκε και στην σελίδα κατηγοριών.
- **τοποθεσία:** Επιλογή πόλης μεταξύ λίστας, που περιέχει όλες τις πρωτεύουσες της Ελλάδος.
- **ενεργός:** Επιλογή κατάστασης λειτουργίας σταθμού. Σε κατάσταση ανενεργή, ο σταθμός δεν λαμβάνει μετρήσεις, καθώς και δεν προβάλλονται συλλογές μετρήσεων του στους απλούς χρήστες.
- **προβολή:** Επιλογή ιδιωτικότητας σταθμού. Με την επιλογή προβολής ως ιδιωτικού, οι μετρήσεις του σταθμού δεν προβάλλονται προς τους απλούς χρήστες.
- **περιγραφή:** Προαιρετική περιγραφή σταθμού.

Στην συνέχεια ο χρήστης επιλέγει από το πάνω δεξί πάνελ, τις διαθέσιμες κατηγορίες μετρήσεων που θα καταγράψει ο σταθμός του. Στο κάτω δεξί πάνελ εμφανίζονται γενικές πληροφορίες που διευκολύνουν την διαδικασία καθώς επίσης παράγεται σε πραγματικό χρόνο το url αποστολής, όσο ο χρήστης συμπληρώνει τις πληροφορίες. Αυτό θα δοθεί εν τέλει στον χρήστη για να προγραμματίσει την πραγματική του συσκευή, ώστε η ίδια να εντοπίζεται από την εφαρμογή και να ταυτίζεται με την εικονική. Με το πάτημα του κουμπιού "Δημιουργία" και τον μετέπειτα έλεγχο των δεδομένων, δημιουργείται ο σταθμός με εμφάνιση κατάλληλου tooltip, ενώ σε περίπτωση σφάλματος ο χρήστης ειδοποιείται καταλλήλως. Αξίζει να σημειωθεί πως, στο περιβάλλον του διαχειριστή, δίνεται μία ακόμα επιλογή (html select) που έχει να κάνει με την επιλογή ιδιοκτησίας. Στην ουσία ο διαχειριστής έχει την δυνατότητα να επιλέξει τον ιδιοκτήτη του σταθμού.




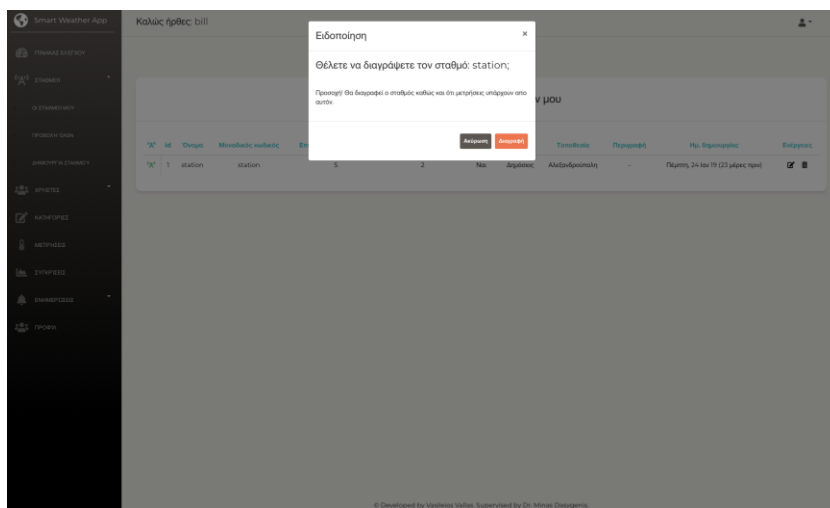
Εικόνα 64: Σελίδα προβολής προσωπικών σταθμών χρήστη

Προβολή προσωπικών σταθμών χρήστη: Στην παραπάνω φωτογραφία εμφανίζεται η σελίδα προβολής, των προσωπικών σταθμών κάθε χρήστη. Η μετάβαση στην συγκεκριμένη σελίδα πραγματοποιείται από το αριστερό μενού, με την επιλογή του συνδέσμου "οι σταθμοί μου". Είναι ίδια και για τους διαχειριστές καθώς και για τους απλούς χρήστες. Από εδώ, μπορεί ο χρήστης να παρακολουθήσει γενικά τους σταθμούς του, τις πληροφορίες που εμφανίζει ο καθένας, καθώς επίσης και να προβεί σε ενέργειες διαγραφής και επεξεργασίας. Έχει δοθεί ένα όριο οκτώ σταθμών ανά σελίδα. Αν ξεπεραστεί ο αριθμός τότε δημιουργείται σελιδοποίηση (pagination).




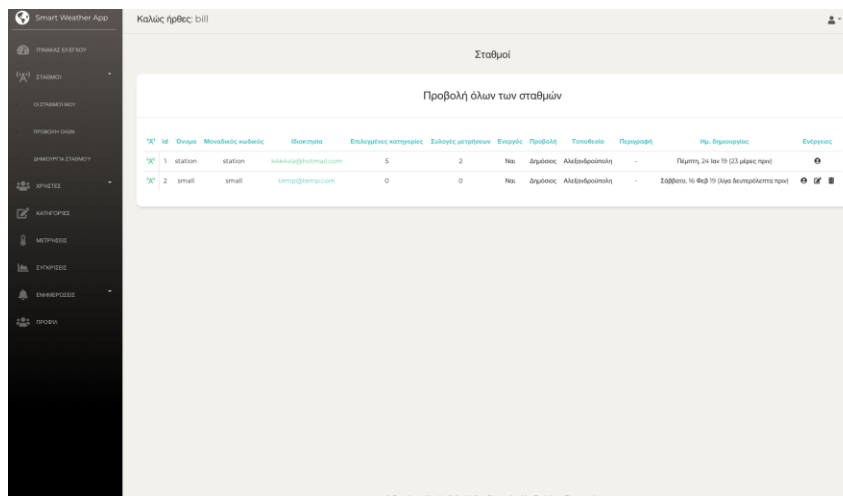
Εικόνα 65: Σελίδα επεξεργασίας σταθμού

Επεξεργασία σταθμού: Από την σελίδα προβολής προσωπικών σταθμών χρήστη, με την επιλογή του πρώτου εικονιδίου στην στήλη των ενεργειών , γίνεται μετάβαση στην σελίδα επεξεργασίας του σταθμού. Η συγκεκριμένη σελίδα διαθέτει ακριβώς τα ίδια panel, με την αντίστοιχη σελίδα δημιουργίας σταθμού, απλώς τα inputs εδώ είναι συμπληρωμένα με τα στοιχεία του σταθμού και παράλληλα επιδέχονται αλλαγές. Με το πάτημα του κουμπιού "Ενημέρωση" και τον μετέπειτα έλεγχο των δεδομένων, γίνεται ανανέωση του σταθμού, παράλληλα με την εμφάνιση κατάλληλου tooltip ή εμφανίζεται μήνυμα σφάλματος. Με την υποβολή των ίδιων στοιχείων με πριν εμφανίζεται κατάλληλη ειδοποίηση.





Εικόνα 66: Modal διαγραφής σταθμού

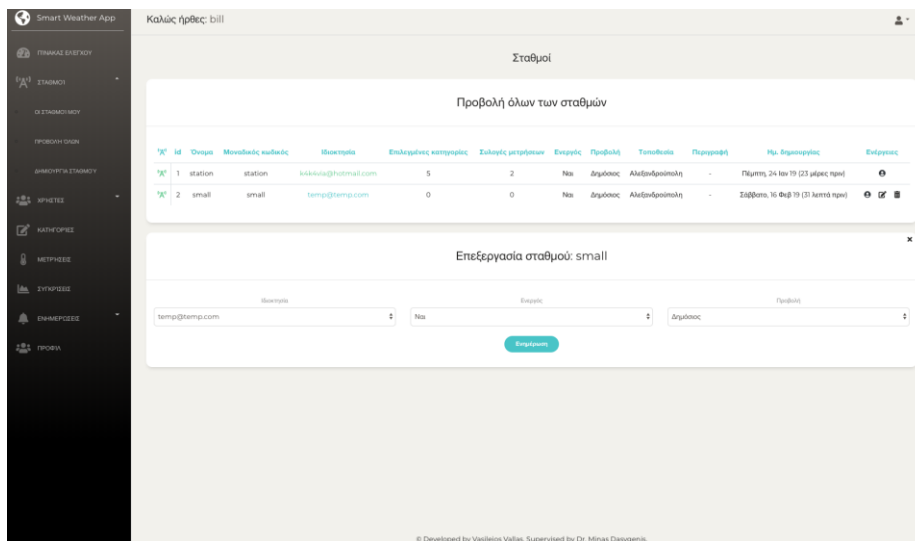
Διαγραφή σταθμού: Από την σελίδα προβολής προσωπικών σταθμών χρήστη, με την επιλογή του δεύτερου εικονιδίου στην στήλη των ενεργειών , εμφανίζεται πάνελ που υπερκαλύπτει την κεντρική οθόνη (modal), δίνοντας κάποιες έξτρα πληροφορίες και ζητάει την επιβεβαίωση της διαγραφής του σταθμού. Με το πάτημα του κουμπιού "Διαγραφή", διαγράφεται ο σταθμός και εμφανίζεται κατάλληλο tooltip. Με την συγκεκριμένη ενέργεια εκτός από την διαγραφή του σταθμού, διαγράφονται επίσης οι συλλογές μετρήσεων του σταθμού, καθώς και ότι συσχετίσεις είχε με κατηγορίες μετρήσεων. Τέλος, με το πάτημα του κουμπιού "Ακύρωση", ακυρώνεται η διαδικασία διαγραφής.




Εικόνα 67: Σελίδα προβολής όλων των σταθμών

Προβολή σταθμών όλων των χρηστών: Στην παραπάνω φωτογραφία εμφανίζεται η σελίδα προβολής όλων των σταθμών συνολικά, από όλους τους χρήστες της εφαρμογής. Η σελίδα αυτή είναι προσβάσιμη μόνο από τους διαχειριστές και τον δημιουργό. Η μετάβαση στην συγκεκριμένη σελίδα πραγματοποιείται από το αριστερό μενού, με την επιλογή του συνδέσμου "προβολή όλων". Είναι ακριβώς ίδια σε λειτουργίες και εμφάνιση, με την αντίστοιχη σελίδα προβολής προσωπικών σταθμών χρήστη. Οι αλλαγές εμφανίζονται, στην στήλη των ενεργειών όπου υπάρχει πλέον το εικονίδιο  σε κάθε σειρά. Αποτελεί σύνδεσμος προβολής σταθμού. Με το πάτημα του, εάν ο επιλεγμένος σταθμός ανήκει στον χρήστη, γίνεται μετάβαση στην σελίδα επεξεργασίας του συγκεκριμένου σταθμού. Εάν ο επιλεγμένος σταθμός έχει άλλον ιδιοκτήτη, πάλι γίνεται μετάβαση σε παρόμοια σελίδα επεξεργασίας του σταθμού, απλώς τα inputs είναι απενεργοποιημένα και δεν επιδέχονται αλλαγή (λειτουργεί καθαρά ως προβολή).

Διαγραφή σταθμού: Η λειτουργία της διαγραφής παραμένει η ίδια και πραγματοποιείται με το ίδιο εικονίδιο που αναφέρθηκε και προηγουμένως . Η μόνη διαφορά είναι, ότι στην παρούσα σελίδα δίνεται η δυνατότητα διαγραφής σταθμών, εκτός από εκείνους που ανήκουν στον χρήστη που πλοηγείται, καθώς και σταθμών που ανήκουν στον δημιουργό.



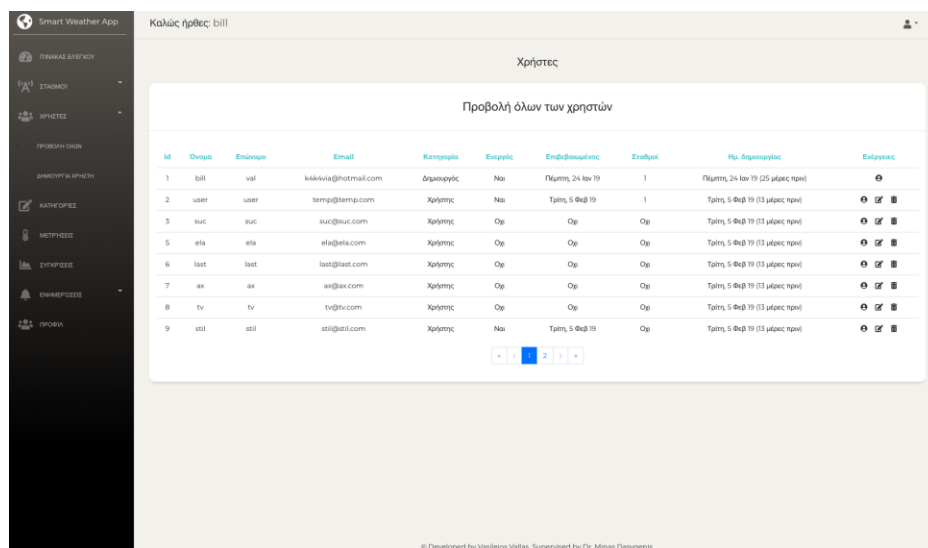
Εικόνα 68: Πάνελ επεξεργασίας σταθμού

Επεξεργασία σταθμού: Όσον αφορά την δυνατότητα επεξεργασίας σταθμού, αυτή επιτυγχάνεται με την επιλογή του εικονιδίου  από την στήλη των ενεργειών και εμφανίζεται στην παραπάνω εικόνα. Δικαίωμα επεξεργασίας επιδέχονται όλοι οι σταθμοί, εκτός από εκείνους που ανήκουν στον χρήστη που πλοηγείται, καθώς και σταθμών που ανήκουν στον δημιουργό. Με την επιλογή του εικονιδίου επεξεργασίας, εμφανίζεται στο κάτω μέρος της σελίδας, ειδικά διαμορφωμένο πάνελ με τις αντίστοιχες επιλογές. Ο χρήστης μπορεί να επεξεργαστεί από εδώ, την ιδιοκτησία του σταθμού, καθώς και καταστάσεις προβολής (ενεργός ή ανενεργός, δημόσιος ή ιδιωτικός). Με το πάτημα του κουμπιού "Ενημέρωση" και τον μετέπειτα έλεγχο των δεδομένων, γίνεται ανανέωση του σταθμού, παράλληλα με την εμφάνιση κατάλληλου tooltip ή εμφανίζεται μήνυμα σφάλματος. Με την υποβολή των ίδιων στοιχείων με πριν εμφανίζεται κατάλληλη ειδοποίηση. Αν γίνει αλλαγή ιδιοκτησίας, σαφώς ενημερώνονται για την ενέργεια, ο παλιός ιδιοκτήτης όπως και ο νέος.

5.1.6 Σελίδα χρηστών (διαχειριστής)

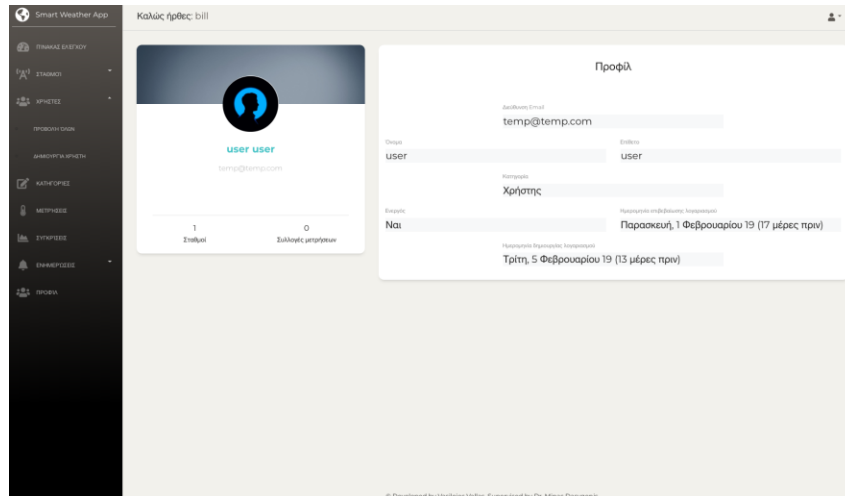
Η σελίδα αυτή αποτελεί ένα ενιαίο σύνολο υποσέλιδων, που προορίζονται για την ενημέρωση και πληροφόρηση των διαχειριστών, για όλους τους χρήστες της εφαρμογής.

Είναι προσβάσιμη μόνο από τους διαχειριστές. Στις υποσελίδες που περιέχονται, υπάρχουν δυνατότητες δημιουργίας, προβολής, επεξεργασίας και διαγραφής χρηστών και διαχειριστών.




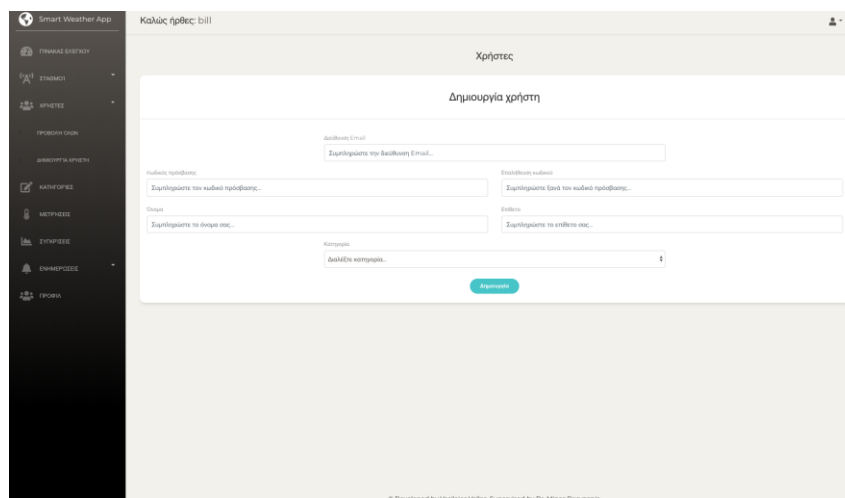
Εικόνα 69: Σελίδα προβολής χρηστών

Προβολή όλων των χρηστών: Στην παραπάνω φωτογραφία εμφανίζεται η σελίδα προβολής, όλων των χρηστών της εφαρμογής. Η μετάβαση στην συγκεκριμένη σελίδα πραγματοποιείται από το αριστερό μενού, με την επιλογή του συνδέσμου "προβολή όλων", που βρίσκεται εντός της κεφαλίδας με την ονομασία "χρήστες". Στο κεντρικό πάνελ της σελίδας, υπάρχει ένας html πίνακας, όπου εμφανίζονται σε αυτόν ανά σειρά, γενικές πληροφορίες ανά χρήστη (όνομα, επίθετο, email, ιδιότητα, κατάσταση, επιβεβαίωση, αριθμός σταθμών, ημερομηνία δημιουργίας), καθώς επίσης υπάρχει, στην δεξιά λωρίδα του πάνελ, η αντίστοιχη στήλη ενεργειών. Από εκεί πραγματοποιούνται ενέργειες προβολής λογαριασμού χρήστη, καθώς και ενέργειες επεξεργασίας και διαγραφής. Έχει δοθεί ένα όριο οκτώ χρηστών ανά σελίδα. Αν ξεπεραστεί ο αριθμός τότε δημιουργείται σελιδοποίηση (pagination).



Εικόνα 70: Σελίδα προβολής λογαριασμού χρήστη

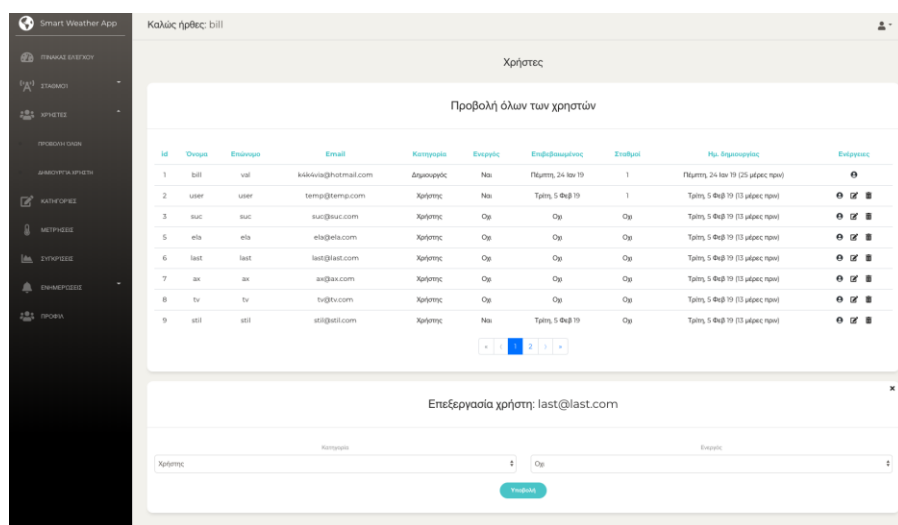
Προβολή λογαριασμού χρήστη: Με την επιλογή του εικονιδίου , από την στήλη ενεργειών, ο χρήστης μεταφέρεται σε σελίδα προβολής πληροφοριών λογαριασμού, για τον χρήστη που επέλεξε, η οποία παρουσιάζεται παραπάνω. Η συγκεκριμένη σελίδα, ουσιαστικά προβάλλει τις ίδιες πληροφορίες, με αυτές που περιέχει ο πίνακας, ωστόσο εδώ παρουσιάζονται τμηματοποιημένες και πιο ευανάγνωστες.



Εικόνα 71: Σελίδα δημιουργίας χρήστη

Δημιουργία χρήστη: Στην παραπάνω φωτογραφία εμφανίζεται η σελίδα δημιουργίας χρήστη. Η μετάβαση στην συγκεκριμένη σελίδα πραγματοποιείται από το αριστερό μενού, με την επιλογή του συνδέσμου "δημιουργία χρήστη". Εδώ δίνεται η δυνατότητα στον


δημιουργό καθώς και στους διαχειριστές, για δημιουργία λογαριασμού χρήστη ή διαχειριστή, παρακάμπτοντας στην ουσία την διαδικασία εγγραφής. Τα στοιχεία που ζητούνται είναι τα ίδια με αυτά της εγγραφής (email, όνομα, επώνυμο, κωδικός πρόσβασης). Επιπλέον δίνεται η δυνατότητα επιλογής της ιδιότητας χρήστη. Με το πάτημα του κουμπιού "Δημιουργία" και τον μετέπειτα έλεγχο των δεδομένων, γίνεται δημιουργία του λογαριασμού, παράλληλα με την εμφάνιση κατάλληλου tooltip ή εμφανίζεται μήνυμα σφάλματος. Αξίζει να σημειωθεί πως, ο λογαριασμός που δημιουργείται, παράλληλα τίθεται και ως επιβεβαιωμένος (δεν αποστέλλεται email επιβεβαίωσης).



Εικόνα 72: Πάνελ επεξεργασίας χρήστη


Επεξεργασία χρήστη: Όσον αφορά την δυνατότητα επεξεργασίας χρήστη, αυτή επιτυγχάνεται με την επιλογή του εικονιδίου από την στήλη των ενεργειών, στην σελίδα προβολής των χρηστών και εμφανίζεται στην παραπάνω εικόνα. Δικαίωμα επεξεργασίας επιδέχονται όλοι οι χρήστες, εκτός από εκείνον που πλοηγείται στην σελίδα καθώς και τον δημιουργό. Με την επιλογή του εικονιδίου επεξεργασίας, εμφανίζεται στο κάτω μέρος της σελίδας, ειδικά διαμορφωμένο πάνελ με τις αντίστοιχες επιλογές για επεξεργασία ιδιότητας (διαχειριστής ή απλός χρήστης) και κατάστασης (ενεργός ή όχι) χρήστη. Με το πάτημα του κουμπιού "Ενημέρωση" και τον μετέπειτα έλεγχο των δεδομένων, γίνεται ανανέωση του χρήστη, παράλληλα με την εμφάνιση κατάλληλου tooltip ή εμφανίζεται μήνυμα σφάλματος. Με την υποβολή των ίδιων στοιχείων με πριν

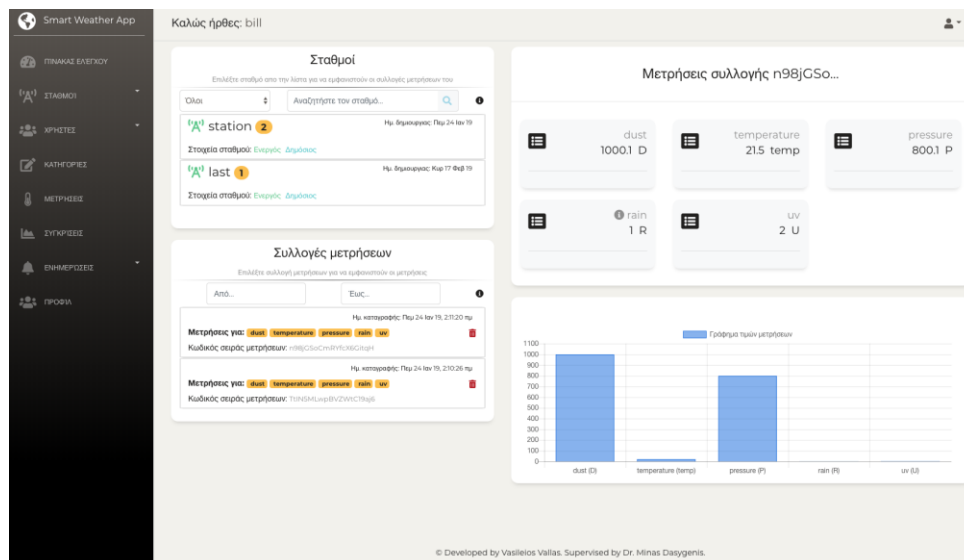
εμφανίζεται κατάλληλη ειδοποίηση. Αν γίνει υποβάθμιση ή αναβάθμιση κάποιου χρήστη, ο ίδιος ειδοποιείται με κατάλληλο tooltip και του γίνεται ανακατεύθυνση σελίδας για την πραγματοποίηση των αλλαγών. Σε περίπτωση αλλαγής κατάστασης του σε ανενεργό, του εμφανίζεται κατάλληλη ειδοποίηση, ο χρήστης οδηγείται στην αποσύνδεση και τέλος του αποστέλλεται ενημερωτικό email. Δεν δύναται να συνδεθεί στην εφαρμογή, μέχρι αλλαγής της κατάστασης του σε ενεργό, από κάποιον διαχειριστή.

Διαγραφή χρήστη: Η λειτουργία της διαγραφής επιτυγχάνεται με την επιλογή του εικονιδίου  από την στήλη των ενεργειών, στην σελίδα προβολής των χρηστών. Ο χρήστης που πλοηγείται, έχει δικαίωμα να διαγράψει όλους τους χρήστες και διαχειριστές, εκτός του ιδίου και του δημιουργού. Η λειτουργία είναι η ίδια που αναλύθηκε και στις άλλες σελίδες. Με την διαγραφή ενός χρήστη, διαγράφονται επίσης οι σταθμοί του καθώς και οι μετρήσεις αυτών.

5.1.7 Σελίδα εμφάνισης αποτελεσμάτων συλλογών μετρήσεων (διαχειριστής - απλός χρήστης)

Η σελίδα αυτή αποτελεί μία από τις πιο σημαντικές και συνάμα ενδιαφέρουσες της εφαρμογής καθώς εδώ, έχει την δυνατότητα ο χρήστης να παρακολουθήσει τα αποτελέσματα από τις συλλογές μετρήσεων των σταθμών. Ο σύνδεσμος για την μετάβαση στην σελίδα βρίσκεται στο αριστερό μενού, στην επιλογή "μετρήσεις". Το γραφικό περιβάλλον είναι το ίδιο και για τους απλούς χρήστες καθώς και για τους διαχειριστές, ωστόσο οι πληροφορίες που εμφανίζονται, διαφέρουν. Αποτελείται από τρία διαφορετικά πάνελ. Το πρώτο έχει τον τίτλο "σταθμοί" και βρίσκεται στο πάνω αριστερό άκρο της σελίδας. Είναι το μοναδικό πάνελ που συναντά ενεργοποιημένο ο χρήστης καθώς εισέρχεται. Στην ουσία αποτελεί μία λίστα (react scrollbar), που εμφανίζει όλους τους σταθμούς της εφαρμογής, οι οποίοι έχουν τουλάχιστον μία συλλογή μετρήσεων. Σε κάθε γραμμή εμφανίζεται το όνομα του σταθμού, ο αριθμός των συλλογών που έχει (κίτρινο χρώμα), η ημερομηνία δημιουργίας του σταθμού, καθώς και κάποιες γενικές πληροφορίες (κατάσταση και ιδιωτικότητα). Πάνω από την λίστα δίνονται επιλογές για λεκτική αναζήτηση σταθμού, όπως και διάφορες επιλογές ταξινόμησης. Στους διαχειριστές, η

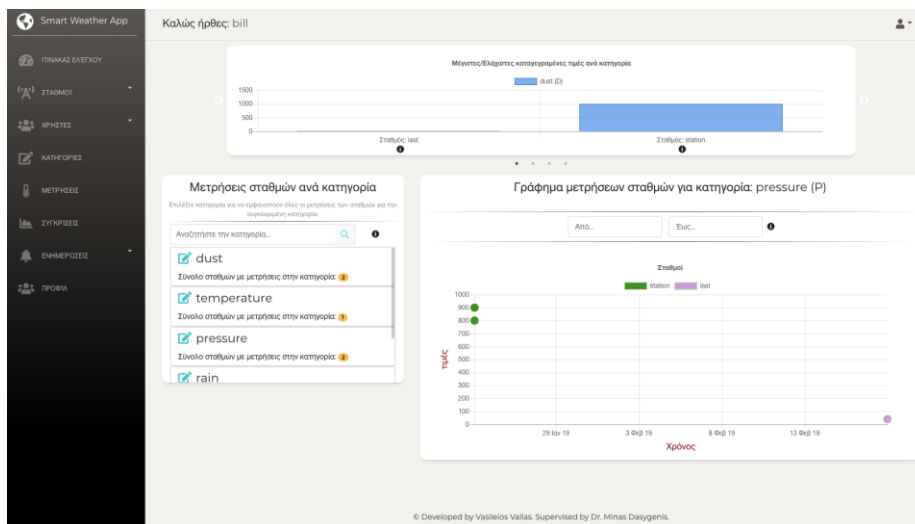
λίστα εμφανίζει όλους τους σταθμούς, ενώ στους απλούς χρήστες, εμφανίζονται μόνο σταθμοί του ιδιοκτήτη, καθώς και σταθμοί άλλων χρηστών, που έχουν επιλεγεί ως ενεργοί και δημόσιοι. Με το πάτημα σε γραμμή της λίστας για επιλογή σταθμού, ενεργοποιείται το δεύτερο πάνελ με ονομασία "συλλογές μετρήσεων" που βρίσκεται κάτω από το πρώτο. Επίσης και αυτό το πάνελ αποτελεί μία λίστα, όπου παρουσιάζονται οι συλλογές μετρήσεων του επιλεγμένου σταθμού. Σε κάθε γραμμή εμφανίζεται η ημερομηνία καταγραφής, ο κωδικός σειράς μετρήσεων, καθώς και οι κατηγορίες μετρήσεων της συλλογής. Επίσης δίνεται δυνατότητα διαγραφής της κάθε συλλογής με το πάτημα του κουμπιού  που βρίσκεται στην δεξιά πλευρά κάθε γραμμής. Δικαίωμα διαγραφής έχουν οι διαχειριστές και ο χρήστης του οποίου, οι συλλογές που παρουσιάζονται, έχουν καταγραφεί από σταθμό της ιδιοκτησίας του. Τέλος πάνω από την λίστα δίνεται δυνατότητα χρονικής οριοθέτησης, ώστε να εμφανίζονται συλλογές επιλεγμένου χρονικού διαστήματος. Με το πάτημα σε γραμμή της λίστας για επιλογή συλλογής, ενεργοποιείται και το τελευταίο πάνελ το οποίο φέρει τον τίτλο "μετρήσεις" και καταλαμβάνει το μισό πλάτος της οθόνης προς τα δεξιά. Εδώ παρουσιάζονται στο πάνω μέρος, σε μικρά πάνελ όλες οι κατηγορίες μετρήσεων από την επιλεγμένη συλλογή, με την τιμή της μέτρησης που είχε καταγραφεί και στο κάτω μέρος, παρουσιάζονται οι ίδιες πληροφορίες σε μορφή γραφήματος. Παράδειγμα της διεπαφής εμφανίζεται στην παρακάτω φωτογραφία.



Εικόνα 73: Σελίδα εμφάνισης αποτελεσμάτων

5.1.8 Σελίδα σύγκρισης αποτελεσμάτων συλλογών μετρήσεων (διαχειριστής - απλός χρήστης)

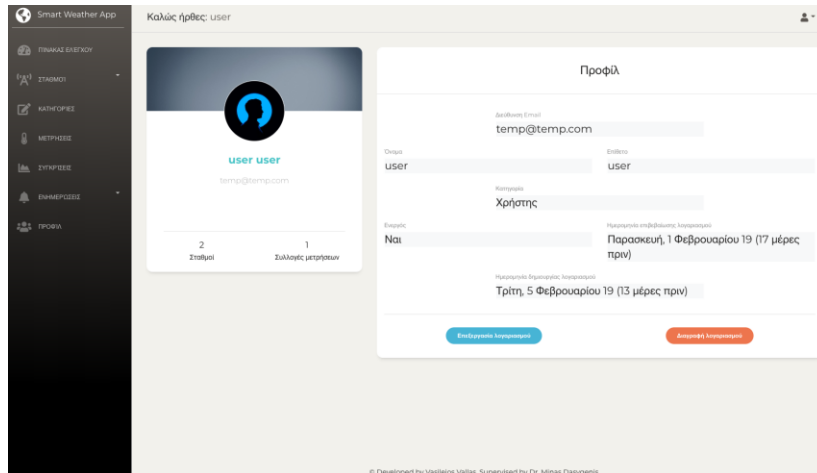
Η σελίδα αυτή αποτελεί έναν διαφορετικό τρόπο εμφάνισης αποτελεσμάτων, καθώς μπορεί εδώ ο χρήστης, να πραγματοποιήσει αρκετές συγκρίσεις και να λάβει σημαντικά αποτελέσματα και στοιχεία. Ο σύνδεσμος για την μετάβαση στην σελίδα βρίσκεται στο αριστερό μενού, στην επιλογή "συγκρίσεις". Το γραφικό περιβάλλον είναι ίδιο για διαχειριστές και απλούς χρήστες. Η διαφορά στις πληροφορίες που εμφανίζονται ανάμεσα στους διαχειριστές και στους απλούς χρήστες, εξακολουθεί να υπάρχει και ο λόγος παραμένει ο ίδιος και αναφέρεται στην σελίδα μετρήσεων. Αποτελείται από τρία διαφορετικά πάνελ. Το πρώτο πάνελ βρίσκεται στο πάνω κεντρικό μέρος της σελίδας και έχει τίτλο "μέγιστες/ελάχιστες καταγεγραμμένες τιμές ανά κατηγορία". Αποτελείται από μία σειρά γραφημάτων σε μορφή slide, που το καθένα από αυτά εμφανίζει, σε μορφή διπλού γραφήματος, την μέγιστη και ελάχιστη τιμή, που έχει καταγραφεί ανά κατηγορία, από όλους τους σταθμούς. Κάθε γράφημα παραμένει στο προσκήνιο για έξι δευτερόλεπτα και το επόμενο παίρνει την θέση του αυτόματα. Το δεύτερο πάνελ, το οποίο βρίσκεται στο αριστερό κεντρικό σημείο της σελίδας, έχει τίτλο "μετρήσεις σταθμών ανά κατηγορία". Εδώ εμφανίζονται σε λίστα όλες οι κατηγορίες μετρήσεων, οι οποίες έχουν καταγεγραμμένη έστω και μία μέτρηση. Σαν πρόσθετη πληροφορία εμφανίζεται το σύνολο των σταθμών, που έχουν καταγράψει μέτρηση στην συγκεκριμένη κατηγορία (κίτρινο χρώμα). Πάνω από την λίστα δίνονται επιλογές για λεκτική αναζήτηση κατηγορίας. Με το πάτημα σε γραμμή της λίστας για επιλογή κατηγορίας, ενεργοποιείται το τελευταίο πάνελ το οποίο φέρει τον τίτλο "γράφημα μετρήσεων σταθμών για κατηγορία" και καταλαμβάνει το μισό πλάτος της κεντρικής οθόνης προς τα δεξιά. Εδώ παρουσιάζονται σε μορφή γραφήματος με κυκλικές ενδείξεις, όλα τα αποτελέσματα μετρήσεων κάθε σταθμού, από την επιλεγμένη κατηγορία. Επίσης πάνω από την λίστα δίνεται δυνατότητα χρονικής οριοθέτησης, ώστε να εμφανίζονται αποτελέσματα επιλεγμένου χρονικού διαστήματος. Παράδειγμα της διεπαφής εμφανίζεται στην παρακάτω φωτογραφία.



Εικόνα 74: Σελίδα σύγκρισης αποτελεσμάτων

5.1.9 Σελίδα προβολής και επεξεργασίας λογαριασμού χρήστη (διαχειριστής - απλός χρήστης)

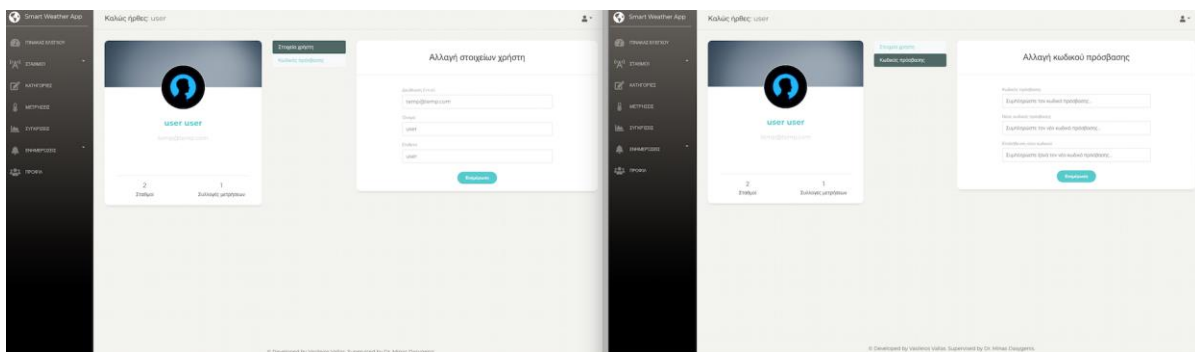
Η τελευταία σελίδα της εφαρμογής που θα παρουσιαστεί, είναι η σελίδα προβολής και επεξεργασίας του λογαριασμού. Απο εδώ, μπορεί ο χρήστης να δει τις πληροφορίες του λογαριασμού του, καθώς και να επεξεργαστεί διάφορα στοιχεία αυτού. Επίσης δίνεται η δυνατότητα σε όλους τους χρήστες και διαχειριστές, εκτός του δημιουργού, να διαγράψουν τον λογαριασμό τους. Ο σύνδεσμος για την μετάβαση στην σελίδα βρίσκεται στο αριστερό μενού, στην επιλογή "προφίλ". Το γραφικό περιβάλλον είναι ίδιο για διαχειριστές και απλούς χρήστες. Αρχικά, η σελίδα που αντικρίζει ο χρήστης δεν είναι τίποτε άλλο από δύο πάνελ, που παρουσιάζουν τα στοιχεία του χρήστη. Στο κάτω άκρο του δεξιού πάνελ βρίσκονται δύο σύνδεσμοι, ένας για επεξεργασία λογαριασμού και ένας για διαγραφή. Παράδειγμα της διεπαφής εμφανίζεται στην παρακάτω φωτογραφία.



Εικόνα 75: Σελίδα προβολής λογαριασμού

Διαγραφή λογαριασμού: Η δυνατότητα διαγραφής λογαριασμού επιτυγχάνεται με την επιλογή του συνδέσμου "διαγραφή λογαριασμού". Η λειτουργία της συγκεκριμένης ενέργειας είναι ίδια με όλες τις προηγούμενες ενέργειες διαγραφής που έχουν αναφερθεί και για αυτό δεν θα αναλυθεί περαιτέρω.

Επεξεργασία λογαριασμού: Με την επιλογή του συγκεκριμένου συνδέσμου, ο χρήστης μεταφέρεται σε υποσελίδα, που περιέχει δύο υποενότητες, μία για επεξεργασία στοιχείων χρήστη (email, όνομα, επώνυμο) καθώς και μία δεύτερη για επεξεργασία του κωδικού πρόσβασης. Οι λειτουργίες είναι παρόμοιες με προηγούμενες άλλων σελίδων και αποτελούνται από φόρμες που ο χρήστης τοποθετεί τα στοιχεία του, ή επεξεργάζεται τα ήδη υπάρχοντα. Παράδειγμα της υποσελίδας με τις δύο υποενότητες εμφανίζεται στην παρακάτω φωτογραφία.



Εικόνα 76: Υποσελίδα επεξεργασίας δεδομένων χρήστη

5.2 Ανάλυση λογισμικού Arduino

Στο τελευταίο κομμάτι της διπλωματικής εργασίας, πλην του epilόγου, γίνεται αναφορά στον προγραμματισμό του μικρο-επεξεργαστή Arduino. Σε πρώτο στάδιο, θα πραγματοποιηθεί η ανάλυση της λειτουργίας της συσκευής, ώστε να γίνει πιο εύκολη η κατανόηση της λογικής που αναπτύχθηκε. Στη συνέχεια θα παρουσιαστεί γραφικά το διάγραμμα ροής του προγράμματος, καθώς επίσης θα αναλυθούν περαιτέρω ορισμένα σημεία και λειτουργίες. Τέλος, θα παρουσιαστούν οι βιβλιοθήκες, καθώς και οι συναρτήσεις οι οποίες χρησιμοποιήθηκαν κατά την ανάπτυξη του προγράμματος του Arduino.

5.2.1 Παρουσίαση προγράμματος λειτουργίας

Σε αυτή την υποενότητα περιγράφεται ο κύκλος λειτουργίας του Arduino από την στιγμή που τοποθετηθεί σε κάποια πηγή ενέργειας, μέχρι την στιγμή που τίθεται σε κατάσταση μειωμένης ενεργειακής κατανάλωσης (θα αναφέρεται παρακάτω ως κατάσταση ύπνου). Τα σημεία που επιλέγονται με αστερίσκο (*), αποτελούν ξεχωριστές λειτουργίες και περιγράφονται αναλυτικά στην επόμενη υποενότητα, για μεγαλύτερη κατανόηση.

Αρχικά, με την εκκίνηση της λειτουργίας του Arduino, αρχικοποιούνται και δηλώνονται οι ακροδέκτες, που επιλέχθηκαν είτε για παροχή ενέργειας προς τα εξαρτήματα είτε για λήψη πληροφοριών από αυτά και ούτω καθεξής. Έπειτα ενεργοποιείται το RGB Led* με κατάλληλη χρωματική ένδειξη η οποία ενημερώνει τον χρήστη, για την εκκίνηση του σταθμού και αμέσως μετά, μπαίνει το Arduino στην κύρια συνάρτηση λειτουργίας. Αυτή αποτελείται από οκτώ καταστάσεις. Μερικές από αυτές είναι καταστάσεις ρουτίνας και πραγματοποιούνται πάντα, ενώ άλλες απλώς υποδηλώνουν διαφορετικές περιπτώσεις χρήσης και καλούνται αναλόγως. Στην αρχική κατάσταση, αφού δοθεί μία χρονική καθυστέρηση* ενός λεπτού, προγραμματίζεται η αφύπνιση* του σταθμού, για την επερχόμενη κατάσταση ύπνου που θα τεθεί στο τέλος του προγράμματος. Έπειτα πραγματοποιείται αναλυτικός έλεγχος* λειτουργίας του Ethernet, του WiFi καθώς

και του SD adapter. Στην συνέχεια, λαμβάνει χώρα η δεύτερη κατάσταση όπου αποτελεί, το κεντρικό σημείο αποφάσεων για τον σταθμό. Εδώ στην ουσία αναλύονται οι προηγούμενοι έλεγχοι και αποφασίζεται η πορεία της λειτουργίας. Υπάρχουν τέσσερα πιθανά αποτελέσματα:

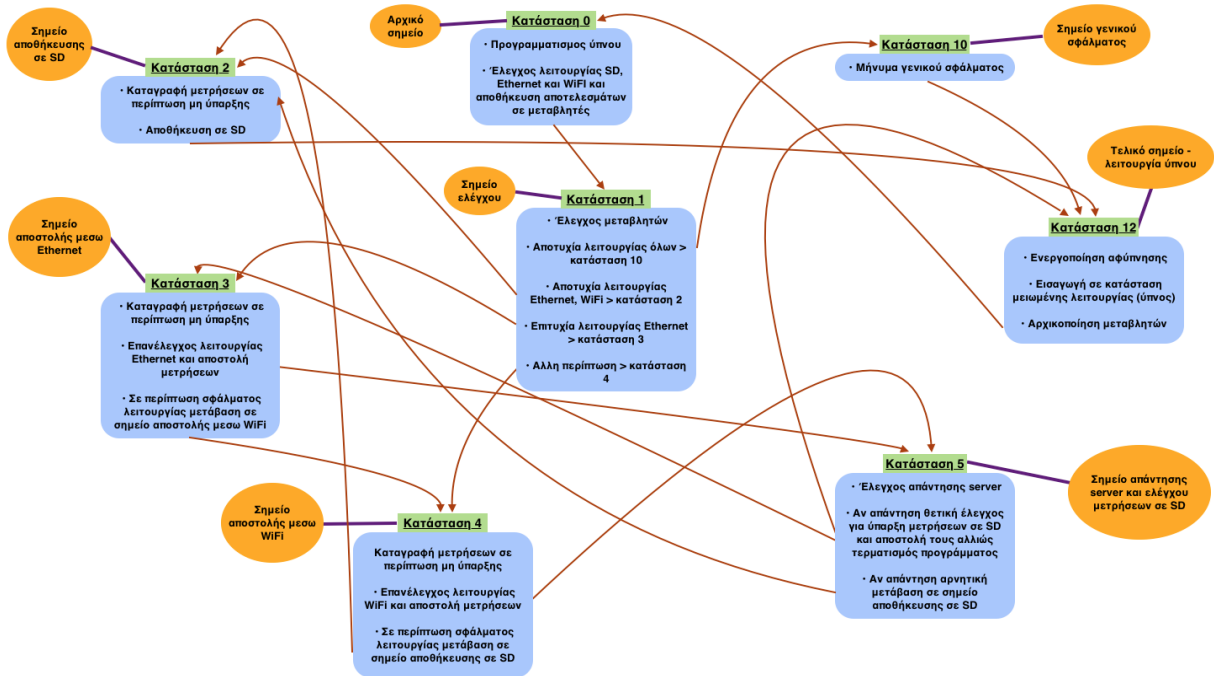
1. Πλήρης αποτυχία στον έλεγχο (Ethernet, WiFi, SD adapter)
2. Αποτυχία στο έλεγχο των στοιχείων αποστολής (Ethernet, WiFi)
3. Επιτυχία στον έλεγχο του Ethernet
4. Επιτυχία στον έλεγχο του WiFi

Η πρώτη περίπτωση, αποτελεί γενικό σφάλμα λειτουργίας του σταθμού, ενημερώνεται ο χρήστης με κατάλληλη χρωματική ένδειξη και οδηγείται στην τελευταία κατάσταση του προγράμματος, που είναι η κατάσταση επιβολής ύπνου (αναλύεται παρακάτω). Η δεύτερη κατάσταση οδηγεί στην καταγραφή μετρήσεων και αποθήκευση τους στην SD card. Η διαδικασία καταγραφής (αν δεν έχει πραγματοποιηθεί ήδη), ουσιαστικά γίνεται ανά εξάρτημα και το αποτέλεσμα ενώνεται (concatenate) με το προηγούμενο, με την χρήση συγκεκριμένου χαρακτήρα (&) σε έναν char array, δημιουργώντας στο τέλος μία παράσταση της μορφής &x=y&z=w (δημιουργείται το payload του POST request). Έπειτα, αφού προηγηθεί έλεγχος του RTC, καταγράφεται και ενώνεται στο τέλος και η χρονική στιγμή της μέτρησης (Unix Timestamp) και δίνεται εντολή ελέγχου του SD adapter. Σε θετική απάντηση ελέγχου, αποθηκεύεται η μέτρηση στην SD card, δίνεται κατάλληλη χρωματική ένδειξη και το πρόγραμμα οδηγείται στην κατάσταση ύπνου, αλλιώς σε αρνητική απάντηση, ακολουθεί η κατάσταση ύπνου με παράλληλη χρωματική ένδειξη σφάλματος. Στην τρίτη περίπτωση, υπάρχει επιτυχία ελέγχου του Ethernet. Έτσι ακολουθεί η καταγραφή μετρήσεων χωρίς την πρόσθεση του χρόνου και γίνεται επανέλεγχος λειτουργίας του Ethernet. Σε θετική απάντηση ελέγχου, πραγματοποιείται POST request σε endpoint της διαδικτυακής εφαρμογής, με "φορτίο" (payload) την παράσταση των μετρήσεων και ακολουθεί, η κατάσταση ελέγχου απάντησης του server. Σε αρνητική απάντηση ελέγχου του Ethernet, ακολουθεί η κατάσταση ελέγχου WiFi. Η συγκεκριμένη κατάσταση εκτελείται μόνο σε δύο συνθήκες, εάν υπάρξει εξ αρχής σφάλμα στον έλεγχο Ethernet ή δημιουργηθεί στην πορεία. Τα βήματα είναι ακριβώς τα ίδια που πραγματοποιούνται και στο Ethernet, με την διαφορά ότι εδώ δεν εκτελείται ξανά δεύτερη

καταγραφή μετρήσεων και χρησιμοποιείται η ίδια παράσταση, καθώς και στην περίπτωση αποτυχίας ελέγχου του WiFi, το πρόγραμμα οδηγείται στην κατάσταση αποθήκευσης της παράστασης μετρήσεων στην SD card. Στην κατάσταση ελέγχου απάντησης του server, ελέγχεται το response status που γύρισε πίσω σαν απάντηση. Σε περίπτωση αποτυχημένης απάντησης του server (HTTP response 300-500), το πρόγραμμα οδηγείται στην κατάσταση αποθήκευσης της παράστασης μετρήσεων στην SD card. Εάν είναι επιτυχές (HTTP response 200), ενημερώνεται ο χρήστης με χρωματική ένδειξη επιτυχίας και γίνεται έλεγχος για ύπαρξη μετρήσεων στην SD card. Εάν υπάρχουν, ακολουθείται η διαδικασία ανάκτησης της πρώτης σε σειρά παράστασης από την SD card και εν συνεχεία γίνεται απόπειρα αποστολής. Το πρότυπο* επιλογής τρόπου αποστολής που ακολουθείται είναι το ίδιο που αναλύθηκε προηγουμένως. Η διαδικασία αυτή εκτελείται μέχρι να σταλούν όλες οι αποθηκευμένες παραστάσεις μετρήσεων. Σε περίπτωση εμφάνισης σφάλματος, η λειτουργία σταματάει στην τρέχουσα παράσταση που ανακτήθηκε και το πρόγραμμα οδηγείται στην κατάσταση ύπνου. Εάν δεν υπάρχουν αποθηκευμένες μετρήσεις, το πρόγραμμα οδηγείται εξ αρχής εκεί. Τέλος στην κατάσταση ύπνου, ενημερώνεται ο χρήστης με χρωματική ένδειξη προειδοποίησης, κλείνει η παροχή ενέργειας προς τους ακροδέκτες και τέλος το Arduino εισέρχεται σε κατάσταση ύπνου. Όταν ενεργήσει η αφύπνιση, το Arduino επανέρχεται σε κανονική λειτουργία, ενεργοποιεί εξίσου τους ακροδέκτες, αρχικοποιεί τις μεταβλητές και ξεκινάει από την πρώτη κατάσταση.

5.2.2 Ροή προγράμματος και επεξήγηση λειτουργιών

Στην υποενότητα αυτή, θα παρουσιαστεί σχηματικά σε διάγραμμα ροής, το πρόγραμμα που αναλύθηκε προηγουμένως. Θα καταγραφούν σε αυτό περιληπτικά, οι πιο σημαντικές λειτουργίες κάθε κατάστασης όπου ενδέχεται να βρεθεί ο σταθμός, καθώς και θα αναλυθούν περαιτέρω, ορισμένες λειτουργίες, οι οποίες αναφέρθηκαν επιγραμματικά και στην προηγούμενη υποενότητα.



Εικόνα 77: Διάγραμμα ροής προγράμματος

Παρακάτω αναλύονται και περιγράφονται σημαντικές λειτουργίες του προγράμματος:

- Προγραμματισμός κατάστασης μειωμένης ενέργειας και ενεργοποίηση αφύπνισης:** Το Arduino με την χρήση της βιβλιοθήκης με ονομασία LowPower.h (αναφέρεται παρακάτω), έχει την δυνατότητα να τίθεται σε κατάσταση μειωμένης ενεργειακής κατανάλωσης (αναφέρθηκε προηγουμένως ως κατάσταση ύπνου), δεδομένο που ενισχύει την αρχική απαίτηση για δημιουργία μίας συσκευής χαμηλών ενεργειακών απαιτήσεων, για μεγάλη αυτονομία χρήσης. Η συγκεκριμένη βιβλιοθήκη διαθέτει επιλογές για χρονικό περιθώριο ύπνου μεταξύ λίγων δευτερολέπτων ή επ' αορίστου. Στην δεύτερη περίπτωση η οποία φαντάζει και ιδανική, ο τρόπος αφύπνισης του Arduino περιορίζεται μόνο από εξωγενείς παράγοντες, όπως η χρήση ενός χρονιστή RTC. Η λειτουργία του Arduino που καθιστά δυνατό αυτό το εγχείρημα, είναι η ιδιότητα ορισμένων ακροδεκτών που διαθέτει, να παραμένουν ενεργοί ακόμα και μετά την επιβολή κατάστασης ύπνου, καταναλώνοντας πολύ μικρό ποσοστό ενέργειας και παράλληλα "αναμένοντας" την ύπαρξη "κάποιας" διαφοράς δυναμικού στην υποδοχή τους. Αν συμβεί αυτό τότε δίνουν εντολή και αφυπνίζεται το Arduino. Έτσι,

στην παρούσα υλοποίηση, ο ακροδέκτης του RTC που είναι υπεύθυνος για την αφύπνιση τοποθετείται σε έναν τέτοιο ακροδέκτη του Arduino ειδικής λειτουργίας. Η βιβλιοθήκη δίνει και την δυνατότητα απενεργοποίησης ορισμένων ενσωματωμένων χρονιστών ή άλλων εξαρτημάτων του Arduino (Watchdog Timer, ADC ή Analog to Digital Converter κ.α.) ωστόσο δεν απενεργοποιεί τους ακροδέκτες παροχής ενέργειας οι οποίοι αποτελούν την κύρια αιτία κατανάλωσης. Έτσι αποφασίστηκε, πως όλοι οι θετικοί ακροδέκτες των εξαρτημάτων πλην του RTC, θα συνδέονται με ψηφιακούς ακροδέκτες που προγραμματίζονται (αναφέρεται στην προηγούμενη υποενότητα) και έτσι λίγο πριν την επιβολή κατάστασης ύπνου, θα απενεργοποιούνται και αυτοί. Το χρονικό διάστημα μεταξύ των αφυπνίσεων καθορίστηκε στα τριάντα λεπτά. Έτσι με απλά λόγια, το Arduino θα ξυπνάει, θα πραγματοποιεί τις ενέργειες που απαιτούνται και θα ξαναπαίρνει σε κατάσταση ύπνου για τα επόμενα τριάντα λεπτά και ούτω καθεξής. Τέλος αξίζει να αναφερθεί για να καταστεί σαφές, πως το Arduino δεν κάνει τίποτα παραπάνω από το να τεθεί σε κατάσταση ύπνου χάρις την βιβλιοθήκη που αναφέρθηκε και όλο το σημαντικό κομμάτι, το πραγματοποιεί το RTC διότι είναι αυτό που ρυθμίζει, κρατάει και προκαλεί την αφύπνιση στο Arduino.

- **Χρονικά διαστήματα καθυστέρησης και παύσεις:** Είτε το Arduino τεθεί σε λειτουργία για πρώτη φορά, είτε βρίσκεται σε ενεργή κατάσταση μετά από αφύπνιση, δίνεται στην αρχή, ένα χρονικό περιθώριο καθυστέρησης ενός λεπτού ώστε τα εξαρτήματα να μπου σε κατάσταση λειτουργίας. Επίσης το DSM501A (εξάρτημα μικροσωματιδίων και σκόνης) σύμφωνα με τον κατασκευαστή απαιτεί ένα λεπτό προθέρμανσης, προτού είναι σε θέση να λάβει έγκυρες μετρήσεις. Για το ίδιο εξάρτημα επίσης, κατά την διαδικασία καταγραφής, η διάρκεια μέτρησης που προτείνεται είναι ένα διάστημα μεταξύ τριάντα δευτερολέπτων. Τέλος, πριν από την διαδικασία ελέγχου όλων των εξαρτημάτων και ιδιαίτερα αυτών που είναι υπεύθυνα για την αποστολή των μετρήσεων, πάντα δίνεται μία παύση των ~500 μιλιδευτερολέπτων (milliseconds) ώστε να αποφευχθούν τυχόν βλάβες.
- **Έλεγχος σε Ethernet, WiFi και SD:** Πάντα, πριν την λειτουργία κάποιου από τα τρία στοιχεία που προαναφέρθηκαν πραγματοποιείται έλεγχος. Σε αρκετές καταστάσεις του

προγράμματος πραγματοποιείται και επανέλεγχος διότι μπορεί, κατά την διάρκεια της λειτουργίας να δημιουργηθεί κάποιο πρόβλημα (πρόβλημα σε δίκτυο WiFi, αφαίρεση καλωδίου Ethernet, αφαίρεση SD card κτλ.). Όσον αφορά τον έλεγχο που γίνεται στα στοιχεία που είναι υπεύθυνα για αποστολή, γίνονται έλεγχοι για την δήλωση της MAC διεύθυνσης στις συσκευές, έλεγχος για τον αν έχει δηλωθεί επιτυχώς διεύθυνση IP στην συσκευή, έλεγχος για ενημέρωση κατάστασης του server (success, timed_out, truncated κτλ.) όπως και για την απάντηση του (success, redirect, internal_server_error κτλ.) μετά την αποστολή μετρήσεων, καθώς και έλεγχος για πρόβλημα του υλικού. Στο WiFi επίσης γίνεται έλεγχος επιτυχημένης σύνδεσης στο δίκτυο που επιλέχτηκε. Για το στοιχείο SD πραγματοποιούνται έλεγχοι για ύπαρξη SD card, καθώς και για βλάβη του υλικού.

- **Λειτουργία Watchdog:** Στην λειτουργία ενός ενσωματωμένου συστήματος υπάρχει πάντα ο κίνδυνος, να "κολλήσει" η λειτουργία του προγράμματος σε κάποιο σημείο με συνέπεια, η συσκευή να μην ανταποκρίνεται και να απαιτεί επανεκκίνηση ώστε να ξαναδουλέψει. Ειδικότερα στην παρούσα υλοποίηση του σταθμού, θα ήταν καταστροφικό εάν τοποθετηθεί ο σταθμός σε μακρινό ή απρόσιτο σημείο για διάστημα ημερών και αυτός κολλήσει στα πρώτα λεπτά λειτουργίας. Η λειτουργία του Arduino που αποτρέπει τέτοιες καταστάσεις ονομάζεται χρονιστής Watchdog (Watchdog Timer) και στην ουσία αποτελεί έναν ενσωματωμένο χρονιστή του Arduino, που τοποθετείται σε "ύποπτα" σημεία κώδικα που υπάρχει κίνδυνος να κολλήσει η λειτουργία (συνήθως στον έλεγχο εξαρτημάτων). Στην ουσία, κλείνουμε μέσα στην λειτουργία του, ένα κομμάτι κώδικα και του δίνουμε το χρονικό διάστημα, ώστε να ξεκινήσει να μετράει. Εάν στον δοθέντα χρόνο κολλήσει η υλοποίηση και ο χρονιστής ξεπεράσει το χρονικό όριο που του δόθηκε, τότε αυτόματα δίνεται εντολή για επανεκκίνηση του Arduino. Αλλιώς, ο χρονιστής τερματίζεται ή απενεργοποιείται και το πρόγραμμα συνεχίζει κανονικά. Στο παρών sketch υπάρχει λειτουργία watchdog σε αρκετά σημεία, κυρίως σε αυτά που πραγματοποιείται έλεγχος λειτουργίας στοιχείων. Τέλος αξίζει να αναφερθεί, ότι ο χρονιστής αυτός είναι αδύνατο να κολλήσει.

- **Ζήτημα ιεραρχίας υλικών:** Έχει προαποφασιστεί ότι πάντα για την αποστολή μετρήσεων, θα επιλέγεται αρχικά ως μέσο αποστολής το Ethernet. Για να επιλεγεί η αποστολή μέσω WiFi σημαίνει ότι ο έλεγχος για Ethernet ήταν εσφαλμένος (μη ύπαρξη καλωδίου, σφάλμα πλακέτας κτλ.).
- **Αποστολή μετρήσεων από SD:** Ο έλεγχος για ύπαρξη μετρήσεων στην SD card πραγματοποιείται πάντα, αμέσως μετά την επιτυχή αποστολή μετρήσεων της συγκεκριμένης χρονικής στιγμής. Έπειτα, το σύστημα έχει προγραμματιστεί ώστε να στείλει, όλες τις αποθηκευμένες μετρήσεις, εάν υπάρχουν. Κάθε αποθηκευμένη μέτρηση που στέλνεται επιτυχώς σβήνεται και από την καταχώρηση του αρχείου. Τέλος, κατά την διαδικασία αποθήκευσης μέτρησης στην SD card, αποθηκεύεται μαζί και η χρονική στιγμή που καταγράφηκε ώστε όταν αποσταλεί, η διαδικτυακή εφαρμογή θα ελέγξει για ύπαρξη ημερομηνίας και θα την αποθηκεύσει αναλόγως.
- **Χρωματικές ενδείξεις RGB Led:** Επιλέχθηκαν 6 χρώματα για να υποδηλώνουν τις καταστάσεις του σταθμού και να ενημερώνουν τον χρήστη. Αυτά είναι:
 1. Λευκό: εκκίνηση λειτουργίας - ενδιάμεση κατάσταση λειτουργιών
 2. Ροζ: ένδειξη αποθήκευσης μετρήσεων σε SD card
 3. Πράσινο: επιτυχής αποστολή μετρήσεων
 4. μωβ: αποτυχία αποστολής μετρήσεων (αρνητική απάντηση από server)
 5. κόκκινο: ένδειξη γενικού σφάλματος (εσφαλμένος έλεγχος για Ethernet, WiFi και SD)
 6. μπλε: ένδειξη πριν την επιβολή κατάστασης ύπνου

5.2.3 Παρουσίαση βιβλιοθηκών και συναρτήσεων

Για τον προγραμματισμό της συσκευής χρησιμοποιήθηκαν οι παρακάτω έντεκα βιβλιοθήκες:

- SPI.h [57]
- Wire.h [58]
- Ethernet.h [59]
- WiFiEsp.h [60]
- SdFat.h [61]
- BME280I2C.h [62]
- LowPower.h [63]
- DS3232RTC.h [64]
- avr/wdt.h [65]
- stdlib.h [66]
- string.h [67]

SPI.h: Η συγκεκριμένη βιβλιοθήκη επιτρέπει την επικοινωνία με συσκευές που χρησιμοποιούν το πρωτόκολλο επικοινωνίας SPI, με το Arduino να λειτουργεί ως την κύρια συσκευή. Χρησιμοποιείται από μικροελεγκτές για την επικοινωνία με μία ή περισσότερες περιφερειακές συσκευές σε μικρές αποστάσεις. Μπορεί επίσης, να χρησιμοποιηθεί για την επικοινωνία μεταξύ δύο μικροεπεξεργαστών. Στην παρούσα συσκευή το στοιχείο που χρησιμοποιεί το συγκεκριμένο πρωτόκολλο επικοινωνίας είναι η πλακέτα Ethernet Shield (Ethernet chip και SD adapter).

Wire.h: Η συγκεκριμένη βιβλιοθήκη επιτρέπει την επικοινωνία με συσκευές που χρησιμοποιούν το πρωτόκολλο επικοινωνίας I2C, με το Arduino να λειτουργεί ως την κύρια συσκευή. Η διαφορά με το πρωτόκολλο επικοινωνίας SPI είναι ότι στο I2C δεν υπάρχει περιορισμός στην απόσταση, ωστόσο η επικοινωνία μεταξύ Arduino και συσκευών είναι μονόδρομη (half-duplex) ενώ στο SPI είναι αμφίδρομη (full-duplex). Στην παρούσα συσκευή δύο στοιχεία χρησιμοποιούν το συγκεκριμένο πρωτόκολλο και είναι το DS3231 RTC καθώς και το GY-BME280 (αισθητήρας θερμοκρασίας, υγρασίας και πίεσης).

Ethernet.h: Πρόκειται για την βιβλιοθήκη η οποία πραγματώνει τη σύνδεση με το διαδίκτυο ενσύρματα, μέσω του Ethernet shield. Υπάρχουν λειτουργίες της που θέτουν το

Arduino να λειτουργήσει είτε ως server, είτε ως client. Η βιβλιοθήκη υποστηρίζει έως και τέσσερις ταυτόχρονες συνδέσεις (εισερχόμενες ή και εξερχόμενες). Ακολουθεί συνοπτική περιγραφή των εντολών και των συναρτήσεων της βιβλιοθήκης οι οποίες χρησιμοποιήθηκαν στην παρούσα υλοποίηση.

1. **Ethernet.begin(mac, ip, dns, gateway, subnet):** Αρχικοποιεί τη βιβλιοθήκη Ethernet και τις ρυθμίσεις του δικτύου. Η βιβλιοθήκη Ethernet υποστηρίζει την δυνατότητα απόκτησης διεύθυνσης IP μέσω DHCP με την χρήση της κλήσης Ethernet.begin(mac). Αξίζει να σημειωθεί ότι, η χρήση του DHCP αυξάνει το μέγεθος του προγράμματος σημαντικά
2. **client():** Είναι η βασική κλάση για όλες τις Ethernet client συναρτήσεις. Δεν καλείται απευθείας, αλλά περιέχεται στην κλήση συνάρτησης που αναφέρονται σε αυτή.
3. **client.connect(IP, port) ή client.connect(URL, port):** Πραγματοποιεί σύνδεση με έναν διακομιστή στην διεύθυνση IP ή στην διεύθυνση URL και τη θύρα που καθορίζεται. Η τιμή επιστροφής δείχνει την επιτυχία ή την αποτυχία της σύνδεσης.
4. **client.stop():** Αποσύνδεση από το διακομιστή.
5. **client.print(data):** Παρέχει δεδομένα στο διακομιστή με τον οποίο έχει συνδεθεί.
6. **client.connected():** Ελέγχει εάν έχει επιτευχθεί η σύνδεση με το διακομιστή και επιστρέφει τιμές true ή false.

WiFiEsp.h: Πρόκειται για την βιβλιοθήκη η οποία πραγματώνει τη σύνδεση με το διαδίκτυο ασύρματα, μέσω του ESP8266. Έχει ακριβώς τις ίδιες συναρτήσεις με την βιβλιοθήκη Ethernet.h με μία διαφοροποίηση στην Ethernet.begin() και αρκετές προσθήκες. Ακολουθεί συνοπτική περιγραφή των εντολών και των συναρτήσεων της βιβλιοθήκης οι οποίες χρησιμοποιήθηκαν στην παρούσα υλοποίηση και διαφέρουν από την Ethernet.h.

1. **WiFi.status():** Πρόκειται για συνάρτηση γενικού ελέγχου του εξαρτήματος.
2. **WiFi.begin(ssid, pass):** Συνάρτηση που ελέγχει για επιτυχή σύνδεση του εξαρτήματος στο δίκτυο WiFi που επιλέχθηκε. Δίνονται σαν παράμετροι το όνομα του δικτύου, καθώς και ο κωδικός πρόσβασης.

SdFat.h: Πρόκειται για την βιβλιοθήκη η οποία επιτρέπει την αλληλεπίδραση του Arduino με την SD card (δημιουργία, επεξεργασία, διαγραφή). Χρησιμοποιείται ευρέως έναντι της αντίστοιχης βιβλιοθήκης του Arduino SD.h λόγω της αξιοπιστίας της. Ακολουθεί συνοπτική περιγραφή των εντολών και των συναρτήσεων της βιβλιοθήκης οι οποίες χρησιμοποιήθηκαν στην παρούσα υλοποίηση.

1. **SD.open(file_name, method):** Η συνάρτηση αυτή δίνει εντολή ανοίγματος ενός αρχείου txt στην SD card είτε για εγγραφή είτε για ανάγνωση. Επιστρέφει σαν απάντηση true ή false. Δίνονται σαν παράμετροι το όνομα του αρχείου, καθώς και η μέθοδος που θα κρίνει την ενέργεια (FILE_WRITE ή FILE_READ).
2. **SD.exists(file_name):** Η συνάρτηση αυτή ελέγχει για την ύπαρξη ενός αρχείου στην SD card. Επιστρέφει σαν απάντηση true ή false. Δίνεται σαν παράμετρος το όνομα του αρχείου.
3. **SD.remove(file_name):** Η συνάρτηση αυτή καλείται για να διαγράψει ένα αρχείο από την SD card. Δίνεται σαν παράμετρος το όνομα του αρχείου.

BME280I2C.h: Η βιβλιοθήκη αυτή χρησιμοποιείται από τον αισθητήρα GY-BME280 και περιέχει συναρτήσεις που επιτρέπουν την καταγραφή των μετρήσεων για τις οποίες προορίζεται. Ακολουθεί συνοπτική περιγραφή των εντολών και των συναρτήσεων της βιβλιοθήκης οι οποίες χρησιμοποιήθηκαν στην παρούσα υλοποίηση.

1. **bme:** Είναι η βασική κλάση για όλες τις bme συναρτήσεις. Δεν καλείται απευθείας, αλλά περιέχεται στην κλήση συνάρτησης που αναφέρονται σε αυτή.
2. **bme.begin():** Συνάρτηση για έλεγχο ύπαρξης του εξαρτήματος. Επιστρέφει σαν απάντηση true ή false.
3. **bme.read(pressure, temperature, humidity):** Η συνάρτηση αυτή καλείται για να επιστρέψει τις τιμές που κατέγραψε ο αισθητήρας για αυτές τις κατηγορίες. Η σειρά των παραμέτρων είναι προδιαγεγραμμένη από την βιβλιοθήκη.

LowPower.h: Η βιβλιοθήκη αυτή επιτρέπει την τοποθέτηση του Arduino σε κατάσταση μειωμένης ενεργειακής κατανάλωσης. Διαθέτει πολλές επιλογές ως προς το είδος της κατάστασης που μπορεί να τεθεί η συσκευή, τον χρόνο, καθώς και τα στοιχεία που μπορεί

να απενεργοποιήσει τα οποία είναι κυρίως χρονοιστές. Ακολουθεί συνοπτική περιγραφή της συνάρτησης της βιβλιοθήκης, η οποία χρησιμοποιήθηκε στην παρούσα υλοποίηση.

1. **LowPower.powerDown(duration, ...modules):** Είναι η συνάρτηση που θέτει το Arduino σε κατάσταση μειωμένης κατανάλωσης ενέργειας. Η πρώτη παράμετρος που παίρνει είναι ο χρόνος και μετά δηλώνονται όλα τα στοιχεία που θα απενεργοποιηθούν.

DS3232RTC.h: Είναι η βιβλιοθήκη που επιτρέπει την επικοινωνία με το DS3231 RTC. Περιέχει συναρτήσεις για την λήψη του χρόνου, καθώς και για την λειτουργία αφύπνισης. Ακολουθεί συνοπτική περιγραφή των εντολών και των συναρτήσεων της βιβλιοθήκης οι οποίες χρησιμοποιήθηκαν στην παρούσα υλοποίηση.

1. **RTC.get():** Η συνάρτηση αυτή επιστρέφει τον χρόνο σε Unix Time (δευτερόλεπτα από 1η Ιανουαρίου 1970).
2. **RTC.setAlarm(mode, ...data):** Συνάρτηση που θέτει αφύπνιση. Η πρώτη παράμετρος έχει να κάνει με την επιλογή χρονικού στοιχείου (δευτερόλεπτα, λεπτά, ώρα κτλ.) και μετά ακολουθούν στοιχεία που καθορίζουν την χρονική στιγμή.
3. **RTC.alarm(type_of_alarm):** Συνάρτηση αρχικοποίησης της αφύπνισης. Σαν παράμετρο παίρνει τον τύπο του ξυπνητηριού (η συνάρτηση επιτρέπει την λειτουργία δύο ξυπνητηριών).
4. **RTC.alarmInterrupt(type_of_alarm, bool):** Συνάρτηση ενεργοποίησης της αφύπνισης. Σαν πρώτη παράμετρο παίρνει τον τύπο του ξυπνητηριού. Η δεύτερη παράμετρος δηλώνει την ενεργοποίηση ή την απενεργοποίηση αντίστοιχα.
5. **attachInterrupt(pin, func, LOW) και detachInterrupt(pin):** Αποτελούν συναρτήσεις που θέτουν τον τρόπο αφύπνισης. Η πρώτη θέτει τον ακροδέκτη που είναι υπεύθυνος να αναμένει την διαφορά δυναμικού στην υποδοχή του. Αφού συμβεί αυτό καλείται αμέσως η συνάρτηση που ορίζεται σαν δεύτερη παράμετρος. Η δεύτερη συνάρτηση απενεργοποιεί την διαδικασία.

avr/wdt.h: Η συγκεκριμένη βιβλιοθήκη χρησιμοποιήθηκε για την υλοποίηση του watchdog του μηχανισμού, ο οποίος επανεκκινεί αυτόματα τη συσκευή εάν αυτή κολλήσει κατά την εκτέλεση του προγράμματος και τεθεί προσωρινά εκτός φυσιολογικής

λειτουργίας. Οι συναρτήσεις και οι εντολές της βιβλιοθήκης που χρησιμοποιήθηκαν ακολουθούν παρακάτω.

1. **wdt_enable(time):** Ενεργοποιεί το χρονόμετρο του watchdog, και το προγραμματίζει να λήξει μετά από μία ορισμένη χρονική στιγμή, όπου και θα πραγματοποιηθεί η επανεκκίνηση της συσκευής. Όπου time είναι μία σταθερά η οποία έχει οριστεί στη βιβλιοθήκη του watchdog. Στην παρούσα διπλωματική εργασία επιλέχθηκαν το 8s. Η αντίστοιχη σταθερά για αυτό το χρονικό διάστημα είναι η WDTO_8S.
2. **wdt_reset():** Επανεκκινεί το χρονόμετρο του watchdog. Αποτυχία εκτέλεσης της συνάρτησης αυτής πριν τη λήξη του χρονομέτρου σημαίνει επανεκκίνηση της συσκευής.

stdlib.h και string.h: Τέλος οι συγκεκριμένες βιβλιοθήκες αποτελούν βοηθητικές και παρέχουν συναρτήσεις για την αλληλεπίδραση μεταξύ πινάκων και strings. Μερικές συναρτήσεις που χρησιμοποιήθηκαν είναι οι: `strncat(char, &c, number)`, `strstr(char, "message")`, `string.length()`, `string.trim()` κ.α.

5.3 Μετρικές κώδικα

Στο κεφάλαιο αυτό αναλύθηκε το λογισμικό που αναπτύχθηκε στα πλαίσια του συστήματος τόσο για την διαδικτυακή εφαρμογή όσο και για την συσκευή Arduino. Κλείνοντας το, θα παρουσιαστούν οι μετρικές κώδικα των δύο αυτών λογισμικών σε επίπεδο γραμμών.

5.3.1 Διαδικτυακή εφαρμογή

Οι μετρικές για την εφαρμογή παρουσιάζονται στον πίνακα παρακάτω:

Ομάδα αρχείων	Αριθμός γραμμών
Controllers	1100
Models	210
JS - React	1600
Other	700

Πίνακας 14: Μετρικές διαδικτυακής εφαρμογής

5.3.2 Arduino

Οι μετρικές για το Arduino sketch παρουσιάζονται στον πίνακα παρακάτω:

Όνομα αρχείου	Αριθμός γραμμών
smartWeatherApp.ino	765

Πίνακας 15: Μετρικές Arduino sketch

Κεφάλαιο 6ο – Επίλογος

Το σύστημα το οποίο αναπτύχθηκε στη συγκεκριμένη διπλωματική εργασία εξυπηρετεί την αυτοματοποιημένη και απομακρυσμένη λειτουργία ενός ευφυούς περιβαλλοντικού σταθμού. Συνδυάζοντας τη χρήση συσκευών υλικού για τη λήψη δεδομένων, μέσω κατάλληλων αισθητήρων, και ενός λογισμικού για τη διαχείρισή τους καθώς και τη λήψη αποφάσεων, κατέστη εφικτός ο σχεδιασμός και η υλοποίηση του έργου αυτού. Η σύσταση αυτού του συστήματος μπορεί, βέβαια, να επεκταθεί με την ενσωμάτωση ολοένα και περισσότερων αισθητήρων και να αυτοματοποιηθεί ακόμα περισσότερο, έτσι ώστε να επιτευχθεί η μέγιστη εξοικονόμηση κάθε τύπου ενέργειας και χρόνου. Όπως ήταν λογικό ωστόσο, προέκυψαν ορισμένα προβλήματα κατά τη φάση του σχεδιασμού και της υλοποίησης τα οποία όμως ξεπεράστηκαν

6.1 Σύνοψη και συμπεράσματα

Συνοψίζοντας, καταλαβαίνει κανείς ότι το έργο αυτό αποτελεί μια καινοτόμα ιδέα, η οποία συνδυάζει διαφορετικές τεχνολογίες για τον έλεγχο και την αυτοματοποίηση ενός περιβαλλοντικού σταθμού. Επιπροσθέτως, συνδυάζει ομαλά την επικοινωνία των διαφόρων συσκευών με την εφαρμογή του συστήματος. Αποτέλεσμα αυτού, είναι η ανάγκη μίας κεντροκοποιημένης διαχειριστικής διαδικτυακής εφαρμογής για όλο το σύστημα και όχι πολλαπλών οπών η κάθε μία θα μπορεί να επικοινωνεί με μια συσκευή μόνο. Όσον αφορά το ενσωματωμένο σύστημα, δεν αποτελεί απλώς μια συνδεσμολογία πλακετών και αισθητήρων. Έχει τοποθετηθεί προγραμματιστικά ενός βαθμού νοημοσύνη, μετατρέποντας το σε ευφύες. Έτσι, λοιπόν, σκέφτεται και αποφασίζει αυτόνομα, για τις ενέργειες οι οποίες πρέπει να εκτελεστούν ή όχι σε μία δεδομένη χρονική στιγμή. Κατ' επέκταση, μπορεί να παρουσιαστεί σαν ένα μοντέλο "έξυπνης" συσκευής. Η δημιουργία τέτοιων έξυπνων συσκευών αποτελεί μονόδρομος στο μέλλον για τον άνθρωπο. Μπορούν πλέον, να μην αποτελούν μόνο άψυχα μηχανήματα πλήρως εξαρτημένα από αυτόν αλλά ευφυείς συσκευές που "μιλούν" και ανταλλάσσουν πληροφορίες με τον άνθρωπο ή ακόμα και μεταξύ τους. Χαρακτηριστικό παράδειγμα αποτελεί η υλοποίηση του

περιβατολογικού σταθμού που υλοποιήθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας, προς απόδειξη της λειτουργικότητας του όλου εγχειρήματος. Η επιλογή της αρχιτεκτονικής του Arduino για την ανάπτυξη του σταθμού έγινε με σκοπό την ταχύτερη δυνατή προτυποποίηση μίας υλικής κατασκευής για την απόδειξη ότι ο αρχικός σκοπός είναι εφικτός και παράλληλα οικονομικά προσιτός. Έτσι, συμπεραίνεται ότι η παρούσα διπλωματική εργασία αποτελεί ένα ισχυρό παράδειγμα το οποίο επιβεβαιώνει ότι η διασύνδεση των συσκευών γενικότερου ενδιαφέροντος μεταξύ διαδικτύου ή και μεταξύ τους, είναι απολύτως πραγματοποιήσιμη και ιδιαίτερα ανερχόμενη. Η διπλωματική αυτή εισάγει έναν πολύ σημαντικό κρίκο για το επόμενο βήμα στην εποχή των ‘έξυπνων’ συσκευών, με αποτέλεσμα να προσφέρεται η δυνατότητα για ακόμα μεγαλύτερη αυτοματοποίηση και χρήση ενοποιημένων συστημάτων στο κοντινό μέλλον.

6.2 Προβλήματα που προέκυψαν και η αντιμετώπιση τους

Σε κάθε καινοτόμο έργο, η πιθανότητα να ανακύψει κάποιο πρόβλημα είτε κατά τη φάση του σχεδιασμού είτε κατά τη διάρκεια της υλοποίησης του είναι μεγάλη, όπως συνέβη και με την παρούσα διπλωματική εργασία. Αρχικά το πρόβλημα που αντιμετωπίστηκε ήταν η ανάγκη για μικρο-επεξεργαστή υψηλών σχετικά επιδόσεων και αντοχής διότι οι διάφορες υλοποιήσεις που μελετήθηκαν πριν τον σχεδιασμό, δεν απαιτούσαν την παράλληλη χρήση του αριθμού των βοηθητικών πλακετών και αισθητήρων που αποφασίστηκε για την παρούσα υλοποίηση. Επίσης ένα δεύτερο στοιχείο που προέκυψε στην πορεία ήταν η ανάγκη για επαναπρογραμματισμό ορισμένων στοιχείων και εξαρτημάτων ώστε να λειτουργούν αρμονικά με το υπόλοιπο σύνολο. Η αντιμετώπιση και των δύο προβλημάτων ήρθε με την μελέτη και την αναζήτηση λύσεων στο διαδίκτυο, ωστόσο χρειάστηκαν αρκετές δοκιμές ώστε να έρθει το επιθυμητό αποτέλεσμα. Τέλος, ενιαίο πρόβλημα υπήρξε και αρχική απαίτηση, που ήταν η δημιουργία μίας διαδικτυακής εφαρμογής που αποτελείται από τεχνολογίες του σήμερα, οι οποίες χρησιμοποιούνται από διαδικτυακούς κολοσσούς. Με την χρησιμοποίηση τέτοιων τεχνολογιών στην παρούσα εργασία γίνεται σαφές ότι, έγινε μεγάλη προσπάθεια εναρμόνισης και συγχρονισμού της εφαρμογής με τα τεχνολογικά μέσα της σύγχρονης εποχής.

6.3 Μελλοντικές επεκτάσεις

Ως το βασικό πλεονέκτημα του συστήματος μπορεί να θεωρηθεί η δυνατότητα βελτίωσης του. Αυτό επιτυγχάνεται με την προσθήκη, νέων χαρακτηριστικών στην εφαρμογή του συστήματος καθώς και με τη δημιουργία νέων συσκευών υλικού για τον έλεγχο και την αυτοματοποίηση μεγαλύτερου εύρους λειτουργιών. Παρακάτω παρατίθενται ενδεικτικά μερικές από τις επεκτάσεις που μπορούν να βελτιώσουν σημαντικά το σύστημα.

- **Παράλληλη λειτουργία του ενσωματωμένου συστήματος και ως server:** Μια σημαντική διαφορά που θα αλλάξει κατά πολύ τα δεδομένα στο ενσωματωμένο σύστημα θα είναι η παράλληλη υλοποίηση και μετατροπή του σε server, με παράλληλη λειτουργία του client. Έτσι θα μπορεί να επικοινωνεί διαρκώς και σε νέα διάσταση με τον χρήστη αποφασίζοντας πότε θα πραγματοποιήσει καταγραφή μετρήσεων, από ποιους αισθητήρες, τι αποτελέσματα να κρατήσει και πότε να σταλούν.
- **Χρήση πρωτοκόλλου SSL για την επικοινωνία με την διαδικτυακή εφαρμογή:** Οι συσκευές που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία για την διασύνδεση με το διαδίκτυο, δεν επιτρέπουν την επικοινωνία μέσω κρυπτογραφημένου καναλιού λόγω περιορισμένων πόρων. Έτσι μια σημαντική αναβάθμιση στην ασφάλεια του συστήματος θα ήταν η χρήση κάποιας υλικής πλατφόρμας με περισσότερες δυνατότητες ώστε να μπορεί να ενσωματωθεί το πρωτόκολλο SSL.
- **Περαιτέρω ασύρματη δικτύωση:** Ο τρόπος ασύρματης δικτύωσης που θα ανέβαζε επίπεδο την υλοποίηση θα ήταν η χρήση του GSM, ώστε το Arduino να χρησιμοποιεί την επικοινωνία μέσω GPRS δικτύου για την αποστολή δεδομένων μέσω διαδικτύου. Παρόλο που αυτή η υλοποίηση θα πρόσθετε μηνιαία κάποιο έξτρα κόστος, θα μετέτρεπε την συσκευή σε μία εντελώς διαφορετική, σπάζοντας τα δεσμά της για εξάρτηση με WiFi και Ethernet, καθιστώντας την τελείως ασύρματη και διαθέσιμη για τοποθέτηση σε οποιοδήποτε γεωγραφικό μέρος, με μόνη απαίτηση την ύπαρξη κάλυψης τηλεφωνικού δικτύου.

Βιβλιογραφία

- [1] Αντίστοιχο σύστημα 1: Real time standalone data acquisition system for environmental data. Ιστότοπος: <https://ieeexplore.ieee.org/document/7853337>, Ιανουάριος 2019
- [2] Αντίστοιχο σύστημα 2: Weather Station Design Using IoT Platform On Arduino Mega. Ιστότοπος: <https://ieeexplore.ieee.org/document/8605456>, Ιανουάριος 2019
- [3] Αντίστοιχο σύστημα 3: Arduino Weather Station Project. Ιστότοπος: <https://ieeexplore.ieee.org/document/8079728>, Ιανουάριος 2019
- [4] Αντίστοιχο σύστημα 4: WiFi ESP8266 and DHT22 sensor. Ιστότοπος: <https://www.hackster.io/Hernanduino/wifi-esp8266-and-dht22-sensor-09d455>, Ιανουάριος 2019
- [5] Αντίστοιχο σύστημα 5: Arduino Weather Station Project. Ιστότοπος: <http://cactus.io/projects/weather/arduino-weather-station>, Ιανουάριος 2019
- [6] Αντίστοιχο σύστημα 6: MKRXero Weather Data Logger. Ιστότοπος: https://create.arduino.cc/projecthub/Arduino_Genuino/mkrzero-weather-data-logger-574190?ref=platform&ref_id=424_trending__&offset=4, Ιανουάριος 2019
- [7] Αντίστοιχο σύστημα 7: Measuring water and air temperature with Arduino GSM shield. Ιστότοπος: http://comoyo.github.io/blog/2013/08/01/m2m_adventures/, Ιανουάριος 2019
- [8] Προγραμματισμός διαδικτύου (Web Development). Ιστότοπος: https://en.wikipedia.org/wiki/Web_development, Ιανουάριος 2019

- [9] Υπολογιστής Εξυπηρετητής (Web Server). Ιστότοπος:
https://developer.mozilla.org/enUS/docs/Learn/Common_questions/What_is_a_web_server, Ιανουάριος 2019
- [10] Database Management System (DBMS). Ιστότοπος:
<https://en.wikipedia.org/wiki/Database>, Ιανουάριος 2019
- [11] HyperText Markup Language (HTML). Ιστότοπος:
https://www.w3schools.com/html/html_intro.asp, Ιανουάριος 2019
- [12] Cascading Style Sheets (CSS). Ιστότοπος: <https://www.w3schools.com/css/>, Ιανουάριος 2019
- [13] Javascript (JS). Ιστότοπος: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript, Ιανουάριος 2019
- [14] Mysql. Ιστότοπος: <https://searchoracle.techtarget.com/definition/MySQL>, Ιανουάριος 2019
- [15] PHP. Ιστότοπος: <http://php.net/docs.php>, Ιανουάριος 2019
- [16] Laravel. Ιστότοπος: <https://laravel.com/docs/5.7>, Ιανουάριος 2019
- [17] Framework. Ιστότοπος: https://en.wikipedia.org/wiki/Software_framework, Ιανουάριος 2019
- [18] MVC. Ιστότοπος: <https://en.wikipedia.org/wiki/Model-view-controller>, Ιανουάριος 2019
- [19] Bootstrap. Ιστότοπος:
https://www.w3schools.com/whatis/whatis_bootstrap.asp, Ιανουάριος 2019
- [20] jQuery. Ιστότοπος: <https://www.w3schools.com/jquery/default.asp>, Ιανουάριος 2019

- [21] React. Ιστότοπος: [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library)),
Ιανουάριος 2019
- [22] Single Page Application (SPA). Ιστότοπος: https://en.wikipedia.org/wiki/Single-page_application, Ιανουάριος 2019
- [23] Application Programming Interface (API). Ιστότοπος:
https://en.wikipedia.org/wiki/Application_programming_interface, Ιανουάριος 2019
- [24] Axios. Ιστότοπος: <https://github.com/axios/axios>, Ιανουάριος 2019
- [25] Redux. Ιστότοπος: <https://redux.js.org/introduction/getting-started>,
Ιανουάριος 2019
- [26] Webpack. Ιστότοπος: <https://webpack.js.org/concepts/>, Ιανουάριος 2019
- [27] Babel. Ιστότοπος: <https://babeljs.io/docs/en/>, Ιανουάριος 2019
- [28] Pusher API. Ιστότοπος: <https://pusher.com/docs>, Ιανουάριος 2019
- [29] Sass. Ιστότοπος: [https://sass-
lang.com/documentation/file.SASS_REFERENCE.html](https://sass-lang.com/documentation/file.SASS_REFERENCE.html), Ιανουάριος 2019
- [30] Arduino. Ιστότοπος: <https://www.arduino.cc/en/guide/introduction>, Ιανουάριος
2019
- [31] Micro-Controller. Ιστότοπος: <https://en.wikipedia.org/wiki/Microcontroller>,
Ιανουάριος 2019
- [32] Database Management System (DBMS). Ιστότοπος:
[https://www.techopedia.com/definition/24361/database-management-systems-
dbms](https://www.techopedia.com/definition/24361/database-management-systems-dbms), Ιανουάριος 2019
- [33] Content Management System (CMS). Ιστότοπος:
https://en.wikipedia.org/wiki/Content_management_system, Ιανουάριος 2019

- [34] PHP manual - CRYPT_BLOWFISH. Ιστότοπος:
<http://php.net/manual/en/function.crypt.php>, Ιανουάριος 2019
- [35] Επίθεση CSRF (Cross Site Request Forgery). Ιστότοπος:
https://en.wikipedia.org/wiki/Cross-site_request_forgery, Ιανουάριος 2019
- [36] Επίθεση XSS (Cross Site Scripting). Ιστότοπος:
https://en.wikipedia.org/wiki/Cross-site_scripting, Ιανουάριος 2019
- [37] Επίθεση SQL injection. Ιστότοπος:
https://en.wikipedia.org/wiki/SQL_injection, Ιανουάριος 2019
- [38] Λειτουργία PDO. Ιστότοπος: <http://php.net/manual/en/book.pdo.php>,
Ιανουάριος 2019
- [39] Arduino Mega. Ιστότοπος: <https://store.arduino.cc/arduino-mega-2560-rev3>,
Ιανουάριος 2019
- [40] Pulse Width Modulation (PWM). Ιστότοπος:
<https://www.arduino.cc/en/tutorial/PWM>, Ιανουάριος 2019
- [41] Ethernet Shield. Ιστότοπος:
<https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>, Ιανουάριος 2019
- [42] ESP8266 WiFi module. Ιστότοπος: <https://en.wikipedia.org/wiki/ESP8266>,
Ιανουάριος 2019
- [43] GY-BME280 - Αισθητήρας θερμοκρασίας, υγρασίας και ατμοσφαιρικής
πίεσης. Ιστότοπος: [http://www.mantech.co.za/datasheets/products/GY-BMP280-
3.3_BG.pdf](http://www.mantech.co.za/datasheets/products/GY-BMP280-3.3_BG.pdf), Ιανουάριος 2019
- [44] Πρωτόκολλο επικοινωνίας I2C. Ιστότοπος: <https://en.wikipedia.org/wiki/I%C2%82C>,
Ιανουάριος 2019

[45] Πρωτόκολλο επικοινωνίας SPI. Ιστότοπος:

<https://www.arduino.cc/en/reference/SPI>, Ιανουάριος 2019

[46] Πρωτόκολλο επικοινωνίας UART. Ιστότοπος:

https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter,

Ιανουάριος 2019

[47] DSM501A - Αισθητήρας καταγραφής συγκέντρωσης σκόνης και μικροσωματιδίων. Ιστότοπος:

https://publiclab.org/system/images/photos/000/003/726/original/tmp_DSM501A_Dust_Sensor630081629.pdf, Ιανουάριος 2019

[48] Low Pulse Occurance (LPO). Ιστότοπος:

<https://www.arduino.cc/reference/en/language/functions/advanced-io/pulsein/>,

Ιανουάριος 2019

[49] Αισθητήρας αίσθησης βροχής ονομασίας Raindrop Sensor. Ιστότοπος:

https://www.openhacks.com/uploadsproductos/rain_sensor_module.pdf,

Ιανουάριος 2019

[50] Αισθητήρας υπεριώδους ακτινοβολίας ονομασίας UV Sensor. Ιστότοπος:

https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/ML8511_3-8-13.pdf,

Ιανουάριος 2019

[51] Υπεριώδης ακτινοβολία. Ιστότοπος:

<https://en.wikipedia.org/wiki/Ultraviolet>, Ιανουάριος 2019

[52] DS3231 RTC - Εξάρτημα καταγραφής χρόνου. Ιστότοπος:

<https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>, Ιανουάριος 2019

- [53] LED (Light Emitting Diode). Ιστότοπος: https://en.wikipedia.org/wiki/Light-emitting_diode, Ιανουάριος 2019
- [54] Transistor. Ιστότοπος: <https://en.wikipedia.org/wiki/Transistor>, Ιανουάριος 2019
- [55] MOSFET. Ιστότοπος: <https://en.wikipedia.org/wiki/MOSFET>, Ιανουάριος 2019
- [56] IRF520. Ιστότοπος: <http://pdf1.alldatasheet.com/datasheet-pdf/view/22389/STMICROELECTRONICS/IRF520.html>, Ιανουάριος 2019
- [57] SPI.h. Ιστότοπος: <https://github.com/PaulStoffregen/SPI>, Ιανουάριος 2019
- [58] Wire.h. Ιστότοπος: <https://github.com/PaulStoffregen/Wire>, Ιανουάριος 2019
- [59] Ethernet.h. Ιστότοπος: <https://github.com/arduino-libraries/Ethernet>, Ιανουάριος 2019
- [60] WifiEsp.h. Ιστότοπος: <https://github.com/bportaluri/WiFiEsp>, Ιανουάριος 2019
- [61] SDFat.h. Ιστότοπος: <https://github.com/greiman/SdFat>, Ιανουάριος 2019
- [62] BME280I2C.h. Ιστότοπος: <https://github.com/finitespace/BME280>, Ιανουάριος 2019
- [63] LowPower.h. Ιστότοπος: <https://github.com/rocketscream/Low-Power>, Ιανουάριος 2019
- [64] DS3232RTC.h. Ιστότοπος: <https://github.com/JChristensen/DS3232RTC>, Ιανουάριος 2019
- [65] avr/wdt.h. Ιστότοπος: https://www.nongnu.org/avr-libc/user-manual/group_avr_watchdog.html, Ιανουάριος 2019

[66] stdlib.h. Ιστότοπος: <http://www.cplusplus.com/reference/cstdlib>, Ιανουάριος
2019

[67] string.h. Ιστότοπος: <http://www.cplusplus.com/reference/cstring>, Ιανουάριος
2019

Παράρτημα

Στο παρόν τμήμα, παρουσιάζονται οι βοηθητικές οδηγίες που πρέπει να ακολουθήσει κάποιος, προκειμένου να εγκαταστήσει την διαδικτυακή εφαρμογή στο δικό του σύστημα (localhost) και να την χρησιμοποιεί από εκεί ή την εγκατάσταση του σε shared hosting provider.

Οδηγίες εγκατάστασης σε τοπικό υπολογιστή (localhost)

Για να επιτευχθεί σωστή εγκατάσταση προτείνεται η χρήση του λογισμικού που αναφέρθηκε στο τεχνολογικό υπόβαθρο του δεύτερου κεφαλαίου. Προτείνεται η χρήση του Laravel Homestead με ταυτόχρονη χρήση του vagrant. Στην συνέχεια ακολουθούν τα βήματα:

1. Αντιγραφή της εφαρμογής στον τοπικό κατάλογο που συνδέεται και βλέπει ο server που έχει "σηκωθεί" από το vagrant.
2. Δήλωση του url της εφαρμογής στα αρχεία Homestead.yaml και .hosts καθώς και της local ip που θα τρέχει η εφαρμογή.
3. Στον κατάλογο της εφαρμογής, αλλαγή ονόματος του αρχείου από .env.example σε .env και συμπλήρωση όλων των πεδίων. Απαιτείται πάροχος για email. Μπορεί να χρησιμοποιηθεί το mailtrap api, ή άλλος πάροχος ή το log file που είναι η καταγραφή που κάνει το Laravel. Επίσης απαιτείται api key για την λειτουργία του pusher.
4. Με ssh στην εφαρμογή μέσω του καταλόγου vagrant (εντολές cli: vagrant up, vagrant provision και vagrant ssh) βρίσκουμε το path για την εφαρμογή μας.
5. Εντός της εφαρμογής δίνουμε από cli: composer install (php dependencies), npm install (node dependencies), php artisan migrate (δημιουργία βάσης δεδομένων).
6. Τέλος λόγω της χρήσης του browserSync με την κλίση της εντολής npm run watch, ανοίγει η εφαρμογή σε default browser και υπακούει σε οποιαδήποτε αλλαγή κώδικα.

Οδηγίες εγκατάστασης σε shared hosting provider

Για να επιτευχθεί σωστή εγκατάσταση, απαιτείται πρώτα η global εγκατάσταση στο hosting των composer και node. Ωστόσο επειδή η εγκατάσταση σε hosting provider είναι αρκετά πιο περίπλοκη και είναι σίγουρο ότι θα προκύψουν σφάλματα στην πορεία, παρακάτω θα περιγραφούν τα γενικά βήματα που ακολουθούνται πάντα:

1. Πραγματοποίηση βήματος 3 από προηγούμενη εγκατάσταση σε localhost.
2. Χωρισμός public φακέλου laravel και τοποθέτηση των εσωτερικών αρχείων στον κατάλογο public_html.
3. Επεξεργασία των αρχείων index.php και server.php ώστε να βρίσκουν τα κατάλληλα paths.
4. Εισαγωγή με cli στο hosting μέσω ssh και εισαγωγή εντολών από βήμα 5.