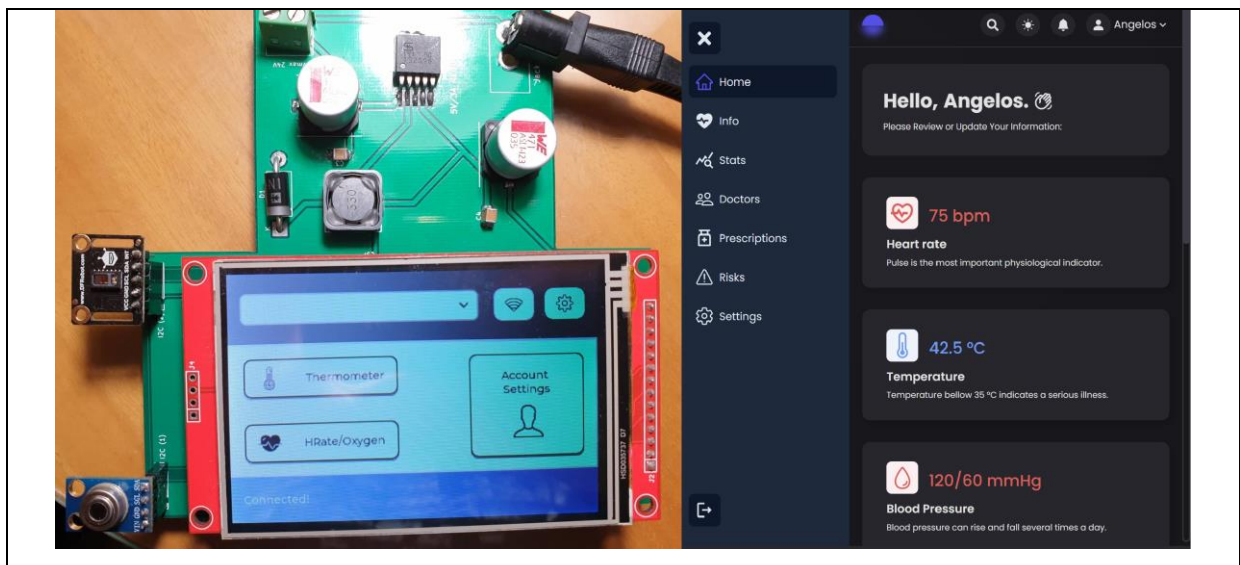


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σύστημα παρακολούθησης και αποθήκευσης
βιομετρικών δεδομένων ασθενών»



Των φοιτητών
Χοστελίδη Άγγελου και Στέλιου
Τράγου
Αρ. Μητρώου: 517155 και 517138

Επιβλέπων
Άγγελος Γιακουμής
Βαθμίδα: Λέκτορας

Ημερομηνία 20/06/2022

Τίτλος Δ.Ε Σύστημα παρακολούθησης και αποθήκευσης βιομετρικών δεδομένων ασθενών

Κωδικός Δ.Ε. 22161

Ονοματεπώνυμο φοιτητή/τών Άγγελος Χοστελίδης , Στέλιος Τράγος

Ονοματεπώνυμο εισηγητή Γιακουμής Άγγελος

Ημερομηνία ανάληψης Δ.Ε 16/03/2022

Ημερομηνία περάτωσης Δ.Ε. 20/06/2022

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία των φοιτητών Χοστελίδη Άγγελου και Στέλιου Τράγου που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Η παρούσα πτυχιακή εργασία εκπονήθηκε σύμφωνα με τις απαιτήσεις και τις εξελίξεις της σύγχρονης ζωής. Όπως σε κάθε τομέα της ζωής μας παρατηρούμε αλλαγές και εκσυγχρονίσεις, θεωρήσαμε σημαντικό να ασχοληθούμε με κάτι επίκαιρο αλλά και χρήσιμο για τον σύγχρονο άνθρωπο. Αφόρμηση της πτυχιακής μας αποτέλεσε η καθημερινή μας ζωή τα τελευταία χρόνια. Δεν μπορεί κανείς να αμφισβητήσει πως ο ιατρικός τομέας δέχεται μεγάλη πίεση στα χρόνια που ζούμε. Ο παραπάνω λόγος ήταν και αυτός που μας ώθησε να συνδέσουμε το αντικείμενο των σπουδών μας με την δημιουργία μιας εφαρμογής που αφορά το ζήτημα της υγείας. Είναι αδιαφισβήτητα σημαντική η σωστή και άμεση παρακολούθηση ενός ασθενή από τον γιατρό του. Η εφαρμογή που αναπτύξαμε επιτρέπει την άμεση αλληλεπίδραση γιατρού και ασθενή και μπορεί να χρησιμοποιηθεί και από τους δύο με διαφοροποιημένες λειτουργίες. Η εφαρμογή που θα παρουσιάσουμε δίνει δυνατότητες όπως, αυτή της άμεσης λήψης βιομετρικών δεδομένων από την πλευρά του ασθενή και άμεση συνταγογράφηση από την πλευρά του επαγγελματία υγείας. Οι παραπάνω αποτελούν λίγες από τις λειτουργίες της συγκεκριμένης εφαρμογής. Στόχος μας ήταν η κατασκευή μιας χρήσιμης και εύχρηστης εφαρμογής που προσφέρει ασφάλεια στους χρήστες που την χρησιμοποιούν. Καταληκτικά, οι απαιτήσεις της καθημερινότητας που ζούμε ήταν το έναυσμα για το περιεχόμενο της παρούσας πτυχιακής εργασίας.

Περίληψη

Στην παρούσα πτυχιακή εργασία σχεδιάστηκε και υλοποιήθηκε ένα σύστημα παρακολούθησης και αποθήκευσης βιομετρικών δεδομένων ασθενών. Πιο συγκεκριμένα, κατασκευάστηκε συσκευή που μετράει καρδιακούς παλμούς, θερμοκρασία και ποσοστό οξυγόνου στο αίμα. Μετά από κάποια μέτρηση, αφού ο χρήστης έχει συνδεθεί στο διαδίκτυο με την χρήση WiFi μέσω της οθόνης αφής της συσκευής, έχει την δυνατότητα να στείλει την μέτρηση αυτή στον προσωπικό του λογαριασμό στην διαδικτυακή εφαρμογή. Στην διαδικτυακή εφαρμογή υπάρχει δυνατότητα σύνδεσης, είτε σαν γιατρός, είτε σαν ασθενής. Ο ασθενής μπορεί να αποθηκεύσει επιπρόσθετα βιομετρικά δεδομένα, όπως την αρτηριακή του πίεση ή την θερμοκρασία του χωρίς την χρήση των αισθητήρων της συσκευής. Η εφαρμογή δίνει στον ασθενή την δυνατότητα επιλογής του γιατρού που επιθυμεί. Έπειτα, ο συγκεκριμένος γιατρός έχει πρόσβαση στα δεδομένα του ασθενή ανα πάσα στιγμή. Επιπλέον, είναι δυνατή η παρουσίαση των βιομετρικών δεδομένων σε γραφήματα σε σταθερές χρονικές περιόδους που ορίζονται μέσω των παρεχόμενων φίλτρων. Σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι να δοθεί δυνατότητα σε γιατρούς να αντλούν συνεχή ιατρικά δεδομένα ζωτικής σημασίας. Με αυτόν τον τρόπο θα μπορούν να κάνουν καίριες παρεμβάσεις και να βγάζουν ασφαλέστερα συμπεράσματα για την κατάσταση της υγείας των ασθενών τους, λόγω της συνεχούς δειγματοληψίας και της άμεσης πρόσβασης που τους παρέχεται.

«Monitoring and data storage system of patients' biometrics»

Chostelidis Angelos , Stelios Tragos

Abstract

In the present dissertation, a monitoring system of a patient's biometric data was designed and implemented. This monitoring system is capable of saving the above data and therefore, distribute them to the patient's desired doctor. More specifically, it consists of an external device that is capable of measuring patient's heart rate, temperature and oxygen level and an online webpage that allows the user to edit or view the data. The device connects to the internet via WiFi through a touch screen and sends the measurements to the cloud. In this particular web application, there is a possibility of connection, either as a doctor or as a patient. The patient can add additional data such as blood pressure, heart rate and temperature, without the need to use measuring instruments. The application gives the patient the opportunity to choose the doctor he wants. The doctor then has access to his patient data at any time. In addition, it is possible to present the biometric data obtained in graphs in fixed periods of time defined through filters provided. The purpose of this dissertation is to enable the physician selected by the patient to obtain continuous vital medical data, as a result he then, will be able to make key interventions and draw safer conclusions about the patient's state of health due to the continuous sampling and the immediate access provided to him.

Περιεχόμενα

Πρόλογος	3
Περίληψη	4
Abstract.....	5
Περιεχόμενα.....	6
Κατάλογος σχημάτων και εικόνων	10
Κεφάλαιο 1ο: Αρχιτεκτονική του συστήματος.....	13
1.1 Εισαγωγή	13
1.2 Διαδικτυακή εφαρμογή.....	13
1.3 User Interface της διαδικτυακής εφαρμογής.....	13
1.4 Αλληπίδραση με την βάση δεδομένων	14
1.5 Διαχωρισμός Λογισμικού	14
1.6 Η Συσκευή για τα βιομετρικά	14
1.7 Επίλογος.....	15
Κεφάλαιο 2ο: Διαθέσιμες τεχνολογίες διαδικτυακών εφαρμογών	16
2.1 Εισαγωγή	16
2.2 Frontend (Παρουσιαστικό της διαδικτυακής εφαρμογής)	16
2.2.1 Απαιτήσεις	16
2.2.2 Τεχνολογίες Frontend	16
2.2.3 Καταληκτική επιλογή τεχνολογίας για το Frontend	20
2.3 Backend (Επικοινωνία με τον server και την βάση δεδομένων).....	20
2.3.1 Απαιτήσεις	20
2.3.2 Τεχνολογίες Backend.....	21
2.3.3 Καταληκτική επιλογή τεχνολογίας για το Backend.....	24
2.4 Επίλογος.....	24
Κεφάλαιο 3ο: Χρήση λογισμικού πραγματικού χρόνου στην συσκευή	25
3.1 Εισαγωγή	25
3.2 RTOS (Real Time Operating System)	25
3.3 Διαθέσιμα λογισμικά RTOS	27
3.3.1 FreeRTOS	27

3.3.2 Zephyr RTOS.....	28
3.4 Επιλογή Λογισμικού RTOS.....	28
3.5 Επίλογος.....	28
Κεφάλαιο 4ο: Ανάπτυξη συσκευής για τις μετρήσεις	30
4.1 Εισαγωγή	30
4.2 Πλατφόρμες ανάπτυξης ηλεκτρονικής συσκευής με μικροελεγχτή	30
4.2.1 ESP32.....	30
4.2.2 STM32	31
4.2.3 ARDUINO	33
4.2.4 Καταληκτική επιλογή πλατφόρμας για την ανάπτυξη της συσκευής	34
4.3 Επιλογή Οθόνης.....	36
4.4 Αρχιτεκτονική λογισμικού της συσκευής.....	38
4.4.1 Finite State Machine.....	38
4.4.2 Ανανέωση των Pixels της οθόνης (Task 1).....	40
4.4.3 Η εφαρμογή της συσκευής.....	41
4.4.4 Πίνακας καταστάσεων και διαχείριση τους (Task 2).....	46
4.4.5 Διάγραμμα ροής λογισμικού.....	47
4.5 Φυσικά στοιχεία συσκευής και σύνδεση μεταξύ τους.....	48
4.5.1 Πρωτόκολλο SPI και οι επαφές του.....	48
4.5.2 Πρωτόκολλο I2C και οι επαφές του	49
4.5.3 Παροχή τροφοδοσίας	50
4.5.4 Λογισμικό σχεδιασμού PCB (KiCad).....	50
4.5.5 Το κύκλωμα και η λίστα των στοιχείων	51
4.6 Αισθητήρες και μετρήσεις	57
4.6.1 Οι Αισθητήρες.....	57
4.6.2 Χρήση συσκευής και μετρήσεις.....	58
4.7 Επίλογος.....	67
Κεφάλαιο 5ο: Η διαδικτυακή εφαρμογή.....	69
5.1 Εισαγωγή	69
5.2 Η εφαρμογή απο πλευρά ασθενή	70

5.2.1 Σελίδα σύνδεσης	70
5.2.2 Σελίδα εγγραφής	71
5.2.3 Αρχική σελίδα	72
5.2.4 Σελίδα μετρήσεων	76
5.2.5 Σελίδα γραφημάτων	82
5.2.6 Σελίδα αλληπίδρασης με γιατρούς.....	83
5.2.7 Σελίδα φαρμάκων.....	87
5.2.8 Σελίδα ρίσκων	89
5.2.9 Σελίδα ρυθμίσεων	90
5.3 Η εφαρμογή από πλευρά Γιατρού.....	91
5.3.1 Αρχική σελίδα	91
5.3.2 Σελίδα μετρήσεων ασθενών.....	92
5.3.3 Σελίδα γραφημάτων με μετρήσεις ασθενών	94
5.3.4 Σελίδα αλληλεπίδρασης με ασθενείς	95
5.3.5 Σελίδα φαρμάκων.....	97
5.3.6 Σελίδα ρίσκων	98
5.4 Restful API.....	100
5.5 Λογική HTTP αιτημάτων.....	104
5.6 Επίλογος.....	106
Κεφάλαιο 6ο: Ασφάλεια	107
6.1 Εισαγωγή	107
6.2 Διαδικασία αναγνώρισης χρήστη με cookies.....	107
6.3 Δικλείδες ασφαλείας των cookies.....	107
6.3.1 HTTP ONLY	108
6.3.2 SECURE	108
6.3.3 SAMESITE	108
6.4 HTTPS	108
6.5 Κρυπτογραφημένη αποθήκευση κωδικών πρόσβασης	108
6.6 Hashing και Salting.....	109

6.7 Επίλογος.....	109
Κεφάλαιο 7ο: Συμπεράσματα και προτάσεις βελτίωσης.....	110
Βιβλιογραφία	111
Οδηγός Λογισμικού	112

Κατάλογος σχημάτων και εικόνων

Σχήμα 3.1- Structure of RTOS [27]	26
Εικόνα 4.1– Το ESP-32	31
Εικόνα 4.2 - Το STM32	33
Εικόνα 4.3 - Το Arduino	33
Σχήμα 4.4 – Αρχιτεκτονική λογισμικού στο ESP32	35
Σχήμα 4.5 - Αρχιτεκτονική ESP32 [29]	36
Εικόνα 4.6 - Επαφές ESP32 [28]	36
Εικόνα 4.7– Η οθόνη από την μπροστινή πλευρά	37
Εικόνα 4.8– Η οθόνη από την πίσω πλευρά	38
Σχήμα 4.9 - Κατάσταση Finite State Machine	39
Σχήμα 4.10 - Καταστάσεις Finite State Machine	39
Σχήμα 4.11 Tasks (έργα) του λογισμικού FreeRtos.....	40
Σχήμα 4.12 - Αρχιτεκτονική λογισμικού οθόνης.....	41
Εικόνα 4.13 - Αρχική οθόνη	41
Εικόνα 4.14 - Οθόνη Ρυθμίσεων.....	42
Εικόνα 4.15 - Οθόνη σύνδεσης στον λογαριασμό	43
Εικόνα 4.16 - Οθόνη Μέτρησης θερμοκρασίας.....	44
Εικόνα 4.17 - Οθόνη Μέτρησης ποσοστό οξυγόνου / καρδιακών παλμών	45
Εικόνα 4.18 – Διάγραμματα ροής.....	47
Εικόνα 4.19 - Το κύκλωμα της συσκευής.....	52
Εικόνα 4.20 - Η σύνδεση των στοιχείων	53
Εικόνα 4.21 - Η σύνδεση των στοιχείων της μετατροπής τάσεως.....	53
Εικόνα 4.22 - Η πάνω πλευρά της συσκευής σε τρισδιάστατο μοντέλο.....	54
Εικόνα 4.23 - Η κάτω πλευρά της συσκευής σε τρισδιάστατο μοντέλο	54
Εικόνα 4.24- Η συσκευή απο την πάνω πλευρά	55
Εικόνα 4.25 - Η συσκευή απο την κάτω πλευρά	55
Εικόνα 4.26 – Η συσκευή με ανοιχτά τα δίκτυα.....	56
Εικόνα 4.27 – Αρχική οθόνη συσκευής.....	56
Εικόνα 4.28 – Η συσκευή χωρίς τους αισθητήρες και την οθόνη	57
Εικόνα 4.29 - Αισθητήρας θερμοκρασίας.....	58
Εικόνα 4.30 – Αισθητήρας Οξυγόνου / Καρδιακών παλμών	58
Εικόνα 4.31 – Τροφοδοσία συσκευής.....	59
Εικόνα 4.32 μέτρηση θερμοκρασίας.....	60
Εικόνα 4.33 – Χρήση αισθητήρα θερμοκρασίας	61
Εικόνα 4.34 – Χρήση αισθητήρα θερμοκρασίας 2	62
Εικόνα 4.35 – Χρήση αισθητήρα θερμοκρασίας 3	63
Εικόνα 4.36 – Μετρήσεις θερμοκρασίας	64
Εικόνα 4.37 – Μενού μέτρησης οξυγόνου/Παλμών	65
Εικόνα 4.38 – Διάγραμμα μετρήσεων καρδιακών παλμών	65
Εικόνα 4.39 – Διάγραμμα μετρήσεων ποσοστό οξυγόνου (Παλμική οξυμετρία)	66
Εικόνα 4.40 – Χρήση αισθητήρα οξυγόνου/καρδιακών παλμών	66
Εικόνα 4.41 - Χρήση αισθητήρα οξυγόνου/καρδιακών παλμών 2	67
Σχήμα 5.1 - Αρχιτεκτονική λογισμικού διαδικτυακής εφαρμογής	69
Εικόνα 5.2 - Στιγμιότυπο του λογισμικού Postman.....	70

Εικόνα 5.3 - Σελίδα σύνδεσης	70
Εικόνα 5.4 - Σφάλμα σύνδεσης.....	71
Εικόνα 5.5 - Σελίδα σύνδεσης	71
Εικόνα 5.6 - Σφάλματα εγγραφής	72
Εικόνα 5.7 - Αρχική σελίδα	72
Εικόνα 5.8 - Αρχική σελίδα με τις ειδοποιήσεις ανοιχτές	74
Εικόνα 5.9 - Αναζήτηση	74
Εικόνα 5.10 - Μπάρα αναζήτησης σε μικρή οθόνη.....	75
Εικόνα 5.11- Αρχική σελίδα με ανοιχτό του υποπλαισίου του προφίλ	76
Εικόνα 5.12 - Σελίδα με τις πρόσφατες μετρήσεις	77
Εικόνα 5.13 - Αλλαγή στοιχείων μέτρησης	79
Εικόνα 5.14 - Σφάλμα αλλαγής στοιχείων μέτρησης	80
Εικόνα 5.15 - Κουμπί για τις επόμενες μετρήσεις	80
Εικόνα 5.16 - Τέλος μετρήσεων	81
Εικόνα 5.17 - Σελίδα προσθήκης νέων μετρήσεων	81
Εικόνα 5.18 - Επιτυχή προσθήκη μέτρησης	82
Εικόνα 5.19 - Σελίδες προσθήκης νέων μετρήσεων	82
Εικόνα 5.20 - Σελίδα γραφημάτων	83
Εικόνα 5.21 - Σελίδα αιτημάτων.....	84
Εικόνα 5.22 - Σελίδα αιτημάτων που έχουν σταλθεί.....	85
Εικόνα 5.23 - Σελίδα εγκεκριμένων γιατρών.....	86
Εικόνα 5.24 - Σελίδα εύρεσης νέων γιατρών.....	87
Εικόνα 5.25 - Επιτυχή αποστολή αιτήματος.....	87
Εικόνα 5.26 - Σελίδα φαρμάκων.....	88
Εικόνα 5.27 - Δημιουργία αρχείου PDF	89
Εικόνα 5.28 - Σελίδα ρίσκων	89
Εικόνα 5.29 – Πληροφορίες ρίσκων	90
Εικόνα 5.30 - Σελίδα ρυθμίσεων	90
Εικόνα 5.31 - Ερώτημα αλλαγής ρυθμίσεων	91
Εικόνα 5.32 - Αρχική σελίδα ιατρού	92
Εικόνα 5.33 - Σελίδα πληροφοριών ασθενών	93
Εικόνα 5.34 - Επιλογή κατηγορίας πληροφοριών	93
Εικόνα 5.35 - Φάρμακα ασθενή.....	94
Εικόνα 5.36 - Σελίδα γραφημάτων	95
Εικόνα 5.37 - Σελίδα γραφημάτων	96
Εικόνα 5.38 – Ασθενείς και εύρεση ασθενών.....	96
Εικόνα 5.39 – Σελίδα προσθήκης νέου φαρμάκου	97
Εικόνα 5.40 - Επιτυχία αίτησης για νέο φάρμακο	98
Εικόνα 5.41- Σφάλμα αιτήματος για νέο φάρμακο.....	98
Εικόνα 5.42 - Λίστα συνταγογραφούμενων φαρμάκων	98
Εικόνα 5.43 - Λίστα ασθενών	99
Εικόνα 5.44 - Φόρμα ρίσκων	99
Εικόνα 5.45 - Λίστα ρίσκων	100
Εικόνα 5.46 - μέθοδοι HTTP	101
Εικόνα 5.47 - Απάντηση αιτήματος HTTP.....	101
Εικόνα 5.48 - Ειδοποιήσεις.....	102

Εικόνα 5.49 - HTTP αίτημα για τις ειδοποιήσεις	103
Εικόνα 5.50 - Μεταβλητή HTTP αιτήματος.....	103
Εικόνα 5.51- Απάντηση DELETE HTTP αιτήματος.....	103
Εικόνα 5.52 - Απάντηση GET HTTP αιτήματος	104
Εικόνα 5.53 - Νέα λίστα ειδοποιήσεων	104
Εικόνα 5.54 - Αλληλεπίδραση ελεγκτών με εσωτερικές υπηρεσίες	105
Εικόνα 5.55 - HTTP αίτημα για φάρμακα	105
Εικόνα 5.56 - Αλληλεπίδραση του ελεγκτή φαρμάκων με εσωτερικές υπηρεσίες.....	106
Εικόνα 6.1 - Αποκωδικοποιημένες πληροφορίες Cookie	107
Εικόνα 6.2 - Κωδικοί πρόσβασης στην βάση δεδομένων	109

Κεφάλαιο 1ο: Αρχιτεκτονική του συστήματος

1.1 Εισαγωγή

Στο κεφάλαιο αυτό θα ανεφερθούμε στην αρχιτεκτονική του συστήματος, δηλαδή από τι αποτελείται η πτυχιακή εργασία με την χρήση παραδειγμάτων. Όπως θα αναφερθεί και παρακάτω τα κύρια στοιχεία είναι η διαδικτυακή εφαρμογή και η εξωτερική ηλεκτρονική συσκευή που λαμβάνει και αποστέλλει τις μετρήσεις στο διαδίκτυο. Επίσης, θα γίνει αναφορά στις λειτουργίες και την χρήση τους.

1.2 Διαδικτυακή εφαρμογή

Αρχικά για να δημιουργηθεί το σύστημα το οποίο υλοποιήσαμε έπρεπε αρχικά να προγραμματιστεί μία διαδικτυακή σελίδα, η οποία θα έμοιαζε πολύ με εφαρμογή κινητού, όπως για παράδειγμα της γνωστής εφαρμογής Facebook. Η διαδικτυακή εφαρμογή, επομένως, μοιάζει με μία εφαρμογή κοινωνικής δικτύωσης αφού οι χρήστες μπορούν να επεξεργαστούν τα δεδομένα τους σε πραγματικό χρόνο. Ένας χρήστης, για παράδειγμα, με ρόλο γιατρού μπορεί να επεξεργαστεί δεδομένα ενός ασθενή, δηλαδή δεδομένα ενός άλλου χρήστη, όμως για να επιτευχθεί αυτό πρέπει προηγουμένως να στείλει ένα αίτημα σε αυτόν τον χρήστη. Αν ο χρήστης δεχτεί το αίτημα η επεξεργασία των δεδομένων ολοκληρώνεται, αν όμως ο χρήστης δεν αποδεχτεί το αίτημα η επεξεργασία απορρίπτεται και η αλλαγή στα δεδομένα δεν πραγματοποιείται. Το παραπάνω θυμίζει την λειτουργία των αιτημάτων φιλίας στην εφαρμογή Facebook. Αν ο χρήστης που λάβει το αίτημα φιλίας δεν το αποδεχτεί, ουσιαστικά δεν θα αλλάξει η λίστα των διαδικτυικών του φίλων, άρα δεν θα ολοκληρωθεί η επεξεργασία των δεδομένων, δηλαδή δεν θα προσθεθεί καινούριος φίλος στην λίστα. Ο χρήστης μπορεί να επισκεφτεί την εφαρμογή στην σελίδα <https://ptuxiakiangstel.herokuapp.com/>. Στον οδηγό λογισμικού παρέχονται και στοιχεία για την σύνδεση σε δοκιμαστικούς λογαριασμούς.

Η παραπάνω αρχιτεκτονική λογισμικού χρησιμοποιήθηκε έτσι ώστε να κάνει τον κάθε χρήστη να ξέρει και να νιώθει ότι έχει τον πλήρη έλεγχο στα προσωπικά του δεδομένα. Βέβαια, ο χρήστης έχει την δυνατότητα και μπορεί να επεξεργαστεί τα δικά του δεδομένα οποιαδήποτε χρονική στιγμή δίχως την παραχώρηση άδειας ή την αποδοχή αιτημάτων από κανέναν.

1.3 User Interface της διαδικτυακής εφαρμογής

Δόθηκε μεγάλη βάση στην εμφάνιση της διαδικτυακής εφαρμογής. Χρησιμοποιήθηκε αγγλική ορολογία στο U.I ή αλλιώς User Interface. Είναι σημαντικό η εφαρμογή να είναι εύκολη στην πλοήγηση, απλή, όμορφη και εύκολη στην χρήση. Πιο συγκεκριμένα, για παράδειγμα, η εφαρμογή θα πρέπει να προσαρμόζεται στην ανάλυση της συσκευής που χρησιμοποιείται κάθε φορά, έτσι ώστε ο χρήστης είτε έχει κυριολεκτικά μεγάλη ή μικρή οθόνη, να μην αναγκάζεται να κάνει μεγέθυνση για να διαβάσει ή να επιλέξει κάτι. Αν ο χρήστης πρέπει να κάνει μεγέθυνση συνέχεια δεν θα μπορεί να πλοηγηθεί εύκολα και ως απότοκο αυτού μπορεί να ενοχληθεί και επομένως να μην ξαναχρησιμοποιήσει την εφαρμογή. Ακόμη, κρίνεται σημαντικό η χρήση των χρωμάτων να είναι συνεπής, δηλαδή να χρησιμοποιηθεί συγκεκριμένη παλέτα χρωμάτων με συγκεκριμένες αποχρώσεις, έτσι ώστε να υπάρχει κατάλληλη αντίθεση χωρίς να κουράζονται τα μάτια του χρήστη. Έχει δοθεί επιλογή να γίνεται εναλλαγή χρωμάτων με την ρύθμιση Light Mode και Dark Mode. Στην πρώτη είναι κυρίως ανοιχτόχρωμα ενώ στην δεύτερη βαθύχρωμα. Οι παραπάνω επιλογές είναι διαθέσιμες στις περισσότερες εφαρμογές που κυκλοφορούν και ο χρήστης είναι οικείος με αυτές.

1.4 Αλληλεπίδραση με την βάση δεδομένων

Για την αποθήκευση όλων των δεδομένων απαραίτητη είναι η χρήση κάποιου είδους βάσης δεδομένων, με σκοπό να αποθηκεύονται και να μην χάνονται. Η αλληλεπίδραση με την βάση δεδομένων για λόγους ασφάλειας δεν πρέπει να γίνει απευθείας, δηλαδή από την εφαρμογή που βλέπει ο χρήστης στην οθόνη του. Για τον λόγο αυτό έγινε προγραμματισμός λογισμικού του οποίου η λειτουργία ήταν να εγκρίνει ή να απορρίπτει την αλληλεπίδραση αυτή. Για παράδειγμα, το λογισμικό αυτό αποφασίζει ότι ο χρήστης μπορεί να επεξεργαστεί τα δικά του δεδομένα μόνο εάν έχει συνδεθεί στην εφαρμογή με τους κατάλληλους κωδικούς, αλλιώς θα γίνει απόρριψη οποιασδήποτε προσπάθειας αλλαγής και το λογισμικό θα απαντάει ότι ο χρήστης πρέπει να συνδεθεί με τα κατάλληλα στοιχεία πρόσβασης, δηλαδή τους προσωπικούς κωδικούς του. Το παραπάνω λογισμικό ονομάζεται backend και είναι λογισμικό το οποίο τρέχει στον server. Επομένως, όταν ο χρήστης χρησιμοποιεί την εφαρμογή και θέλει να κάνει οποιαδήποτε αλλαγή σε δεδομένα, αυτή η ενέργεια φιλτράρεται πρώτα από τον server και μόνο αν ο server αποφασίσει ότι ο χρήστης αυτός έχει δικαίωμα να κάνει αυτή την ενέργεια γίνεται η αλληλεπίδραση με την βάση δεδομένων. Έτσι, ουσιαστικά δημιουργούνται και οι ρόλοι των χρηστών (γιατρός, ασθενής), δηλαδή ένας ασθενής δεν μπορεί να γράψει φάρμακο, οπότε αν το επιχειρήσει ο server θα απαντήσει ότι δεν έχει το δικαίωμα να το κάνει, γιατί δεν είναι γιατρός.

1.5 Διαχωρισμός Λογισμικού

Η λογική του backend είναι ξεχωριστή από την λογική του frontend, δηλαδή από την εφαρμογή που βλέπει ο χρήστης στην οθόνη του. Ο τρόπος που συνδέονται είναι ο εξής: Το frontend στέλνει HTTP αιτήματα στο backend και αυτό απαντάει ανάλογα, έτσι η εφαρμογή ανάλογα με την απάντηση του backend απεικονίζει τα κατάλληλα στοιχεία. Αφού οι λογικές είναι ξεχωριστές δίνεται η δυνατότητα να σταλθούν αιτήματα στον server και εκτός εφαρμογής. Αυτό ήταν σημαντικό να γίνει, διότι σκοπός την πτυχιακής ήταν να φτιαχτεί και μία συσκευή για την μέτρηση διαφόρων βιομετρικών δεδομένων και την αποστολή τους, εν τέλει στην βάση δεδομένων. Καθώς, όμως η συσκευή αυτή δεν έχει την δυνατότητα να κάνει χρήση προγράμματος για την πλοήγηση στο διαδίκτυο, δηλαδή ένα πρόγραμμα σαν το google chrome, λόγω τεχνικών χαρακτηριστικών, έπρεπε να μπορεί να επικοινωνεί με την βάση δεδομένων και να στέλνει αιτήματα στον server χωρίς την χρήση κάποιου προγράμματος. Το μεγαλύτερο πλεονέκτημα αυτής της αρχιτεκτονικής η οποία ονομάζεται headless είναι ότι οποιοσδήποτε μπορεί να στείλει με ευκολία αιτήματα στον server, άρα ο δημιουργός μπορεί να φτιάξει διάφορα προϊόντα που να μιλάν με τον ίδιο σέρβερ με ευκολία, όπως για παράδειγμα μια εφαρμογή κινητού ή όπως στην παρούσα πτυχιακή εργασία μια συσκευή. Έτσι όλο αυτό το σύστημα καθιστά την χρήση οθόνης σε συσκευές ακόμη και προαιρετική, καθώς μπορεί για παράδειγμα να προγραμματιστεί το πάτημα κάποιου κουμπιού να στέλνει ένα αίτημα στον server και αν η απάντηση του server είναι θετική να ανάβει ένα led πράσινο ή αντίστοιχα αν είναι αρνητική να ανάβει ένα κόκκινο. Με αυτόν τον τρόπο ο χρήστης της συσκευής μπορεί να επικοινωνεί με τον ίδιο server που επικοινωνεί και μία συσκευή 500 ευρώ ή και ακριβότερη. Με αυτόν τον τρόπο καταλήξαμε να φτιάξουμε μια συσκευή μικρής αξίας που να στέλνει τα δεδομένα που θέλουμε. Στην συσκευή που φτιάχτηκε προστέθηκε οθόνη αφής, αλλά όπως αναφέρθηκε παραπάνω θα μπορούσε να σχεδιαστεί έτσι ώστε να μην είναι απαραίτητη.

1.6 Η Συσκευή για τα βιομετρικά

Η συσκευή η οποία φτιάχτηκε χρησιμοποιεί επεξεργαστή με ελάχιστους πόρους, δηλαδή με πόρους όπως ένας υπολογιστής 15 χρονών. Αυτό την καθιστά φθηνή και εύκολα υλοποιήσιμη. Το γεγονός

αυτό βέβαια έχει και κάποια αρνητικά, όπως ότι πρέπει να γίνει έξυπνος προγραμματισμός λογισμικού έτσι ώστε να χώραει στην μνήμη του επεξεργαστή ή να γίνει αποφυγή αποθήκευση φωτογραφιών για τον ίδιο λόγο.

Το πιο σημαντικό κομμάτι στην δημιουργία της συσκευής ήταν το λογισμικό. Έπρεπε να χρησιμοποιηθεί λογισμικό το οποίο να είχε την δυνατότητα να επιτρέψει τον χρήστη να μπορεί να κάνει πολλές λειτουργίες ταυτόχρονα. Για παράδειγμα, καθώς η συσκευή ψάχνει για συσκευές στις οποίες μπορεί να συνδεθεί στο διαδίκτυο, να έχει τη δυνατότητα να ο χρήστης να κάνει κι άλλα πράγματα, όπως να αλλάζει κάποιες ρυθμίσεις ή γενικά να πλοηγείται στο μενού της εφαρμογής. Αν αυτό δεν γινόταν η συσκευή θα «πάγωνε», καθώς έψαχνε για συσκευές. Αν ο χρήστης προσπαθούσε να κάνει κάτι άλλο η συσκευή δεν θα αποκρινόταν. Έτσι με την χρήση κατάλληλου λογισμικού ο χρήστης νιώθει ότι χρησιμοποιεί μια συσκευή σαν το κινητό του τηλέφωνο. Με αυτόν τον τρόπο εκπληρώνονται οι προσδοκίες που έχει και κάποιος από μία συσκευή, δηλαδή δεν φαίνεται να κολλάει, ανταποκρίνεται σε αλλαγές, στην πλοήγηση και γενικότερα στην κάθε ενέργεια του χρήστη χωρίς καθυστέρηση. Συνήθως, το λογισμικό που χρησιμοποιείται σε τέτοιους επεξεργαστές με λίγους πόρους ονομάζεται RTOS (Real Time Operation System), διότι, κατά την διάρκεια της χρήσης του, η απόκριση είναι πραγματικού χρόνου.

Με την χρήση αυτής της συσκευής και την χρήση RTOS μπορούμε με ευκολία να στείλουμε στον server αιτήματα σε πραγματικό χρόνο και στο τέλος να γίνει αλληλεπίδραση με την βάση δεδομένων. Όπως προαναφέρθηκε παραπάνω η συσκευή και η εφαρμογή αλληλεπιδρούν με την ίδια βάση δεδομένων άρα και με τον ίδιο server. Όποια αλλαγή γίνει από την συσκευή θα αντικατοπτρίζεται και στην εφαρμογή που θα είναι συνδεδεμένος ο χρήστης στον προσωπικό του υπολογιστή.

1.7 Επίλογος

Η αρχιτεκτονική του συστήματος χωρίζει όλες τις τεχνολογίες σε κομμάτια. Η κάθε μία τεχνολογία που χρησιμοποιείται είναι ανεξάρτητη από την άλλη, έτσι υπάρχει μεγάλη ευελιξία στο σχεδιασμό και στην περαιτέρω βελτίωση. Κάθε κομμάτι της αρχιτεκτονικής έχει δικά του χαρακτηριστικά και κανόνες. Η διαδικτυακή εφαρμογή αποτελείται από το παρουσιαστικό κομμάτι, δηλαδή το frontend, όπου ο ασθενής ή ο γιατρός μπορεί να δει μετρήσεις, αιτήματα για φάρμακα ή ρίσκα κτλπ, και το λογισμικό που βρίσκεται στον server, ο οποίος είναι υπεύθυνος για την αποστολή των πληροφοριών στις εφαρμογές. Το frontend πρέπει να είναι απλό και εύχρηστο έτσι ώστε να χρησιμοποιείται από τον χρήστη με ευκολία ενώ το backend πρέπει να στέλνει τις σωστές πληροφορίες στον χρήστη και να αποτρέπει την διαρροή πληροφοριών σε χρήστες που δεν πρέπει να έχουν πρόσβαση σε αυτές. Η συσκευή για την μέτρηση των βιομετρικών δεδομένων μετράει και στέλνει τα δεδομένα στην διαδίκτυο και επικοινωνεί με τον παραπάνω server. Χρησιμοποιείται ειδικό λογισμικό που ονομάζεται RTOS και την επιτρέπει να κάνει πολλές λειτουργίες ταυτόχρονα όπως να ανακαλύπτει νέες συσκευές για σύνδεση στο διαδίκτυο, ενώ ταυτόχρονα ο χρήστης να αλλάζει ρυθμίσεις. Έτσι, ο χρήστης νιώθει ότι χρησιμοποιεί μία συσκευή που αποκρίνεται γρήγορα, σαν το κινητό του τηλέφωνο.

Κεφάλαιο 2ο: Διαθέσιμες τεχνολογίες διαδικτυακών εφαρμογών

2.1 Εισαγωγή

Λαμβάνοντας υπόψιν την αρχιτεκτονική του συστήματος χρειάζεται να ψάξουμε τεχνολογίες που είναι διαθέσιμες για την επίτευξη των στόχων μας. Έχοντας διαφορετικές απαιτήσεις για κάθε λειτουργία που θέλουμε να δημιουργήσουμε πρέπει να γίνει ξεχωριστή έρευνα για κάθε τομέα. Πιο συγκεκριμένα διαφορετική έρευνα για το frontend, backend κτλ. Ευνοϊκό είναι το γεγονός ότι η λογική κάθε λειτουργίας είναι ανεξάρτητη από την άλλη. Όπως αναφέρθηκε επειδή η λογική της εφαρμογής δεν συνδέεται άμεσα με το λογισμό του server βρισκόμαστε να έχουμε πολλές και διαφορετικές επιλογές εργαλείων για την επίλυση των προβλημάτων μας.

2.2 Frontend (Παρουσιαστικό της διαδικτυακής εφαρμογής)

2.2.1 Απαιτήσεις

Συγκεκριμένα για την δημιουργία της εφαρμογής μας με την οποία αλληλεπιδρά ο χρήστης, δηλαδή το κομμάτι του frontend μας κέντρισαν το ενδιαφέρον 4 τεχνολογίες οι οποίες συνάδουν με τις απαιτήσεις τις οποίες είχαμε. Οι απαιτήσεις αυτές ήταν :

- Ευκολία στην σύνταξη του κώδικα για το «γράψιμο» της εφαρμογής
- Ευκολία για την αλληλεπίδραση της εφαρμογής με τον server
- Όσο τον δυνατό μεγαλύτερη υποστήριξη για την συγκεκριμένη τεχνολογία στο διαδίκτυο, είτε μέσω εφαρμογών κοινωνικής δικτύωσης είτε από βίντεο κτλ.
- Να είναι κώδικας με ανοιχτή πηγή (Open source).
- Να συντηρείται και να βελτιώνεται ακόμη και σήμερα.
- Να μην γίνεται ανανέωση της ιστοσελίδας καθώς ο χρήστης αλλάζει σελίδες μέσα στην εφαρμογή, αλλά να γίνεται αντικατάσταση μόνο του απαραίτητου κομματιού .
- Να υπάρχει ελευθερία στην αρχιτεκτονική της εφαρμογής δηλαδή όσο το λιγότεροι περιορισμοί γίνεται για το πως θα φτιάξουμε τους φακέλους για να βάλουμε τα αρχεία με τον κώδικα μέσα ή ακόμη και να δίνεται ελευθερία για το ποιά έκδοση της γλώσσας προγραμματισμού θέλουμε να χρησιμοποιήσουμε λόγω της σύνταξης της κτλ.

Οι τεχνολογίες που καταλήξαμε είναι οι εξής:

- React.js
- Angular.js
- Vue.js
- jQuery

2.2.2 Τεχνολογίες Frontend

2.2.2.1 React.js

Η React είναι μια βιβλιοθήκη, ανοιχτού κώδικα, της JavaScript με την οποία μπορούμε να φτιάξουμε μία διαδικτυακή εφαρμογή. Αποτελεί την πιο διαδεδομένη βιβλιοθήκη για το front, καθώς φτιάχτηκε και χρησιμοποιείται από το Facebook. Η React.js παραχωρεί την πλήρη ελευθερία στον προγραμματιστή να οργανώσει την εφαρμογή με όποιον τρόπο θέλει. Το πιο μεγάλο πλεονέκτημα της React είναι ότι επιτρέπει να γράφεις κώδικα JavaScript μέσα σε κώδικα html. Λόγω αυτού καθιστά

Κεφάλαιο 2

την ανάπτυξη εφαρμογών πάρα πολύ γρήγορη. Επιπλέον επειδή δίνει στον δημιουργό μεγάλη ελευθερία του επιτρέπει να λύσει ένα πρόβλημα με πολλούς διαφορετικούς τρόπους και δεν τον περιορίζει σε συγκεκριμένη αρχιτεκτονική κώδικα. Επίσης είναι μια βιβλιοθήκη που συντηρείται ακόμη και σήμερα και χρησιμοποιεί μοντέρνες τεχνολογίες και τεχνικές για να πετύχει την ανάπτυξη εφαρμογών. Χρησιμοποιείται γιατί είναι εύκολη στην χρήση και παρέχει υψηλή απόδοση. Υπάρχει πολύ μεγάλη υποστήριξη στο διαδίκτυο και πάρα πολλές βιβλιοθήκες βασισμένες πάνω στην react που έχουν γραφτεί και συντηρούνται από τρίτους δημιουργούς. Έτσι με αυτόν τον τρόπο ο δημιουργός μπορεί να διαλέξει ένα από τα πολλά πακέτα βιβλιοθηκών, για παράδειγμα για να πετύχει εύκολη, γρήγορη και ασφαλή επικοινωνία με server, κάτι το οποίο αποτελεί σημαντικό για την περίπτωση της πτυχιακής εργασίας καθώς ταίριαζε με το προφίλ της αρχιτεκτονικής που επιλέξαμε να ακολουθήσουμε.

Χαρακτηριστικά της ReactJs [10]:

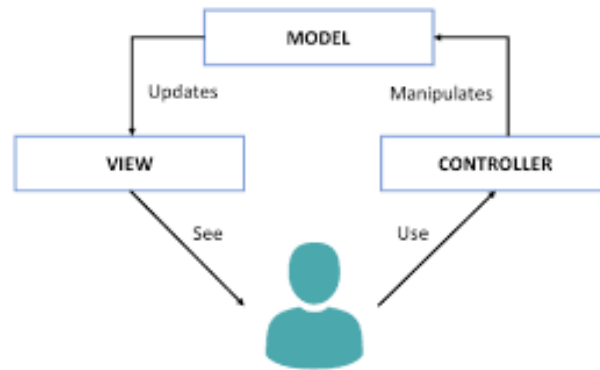
- Όταν αλλάζει ή ενημερώνεται κάτι στην εφαρμογή αυτόματα γίνεται ανανέωση και αλλαγή στο τι βλέπει ο χρήστης και σε αυτό βοηθάει το εικονικό (virtual) DOM.
- Επιτρέπεται η χρήση της JavaScript μέσα σε κώδικα HTML (αρχεία .jsx).
- Υπάρχει η δυνατότητα χρήσης της react native για την ανάπτυξη εφαρμογών σε Android και iOS.
- Η ανάπτυξη της εφαρμογής γίνεται σε μικρά κομμάτια, τα λεγόμενα components .
- Συνήθως χρησιμοποιείται για την ανάπτυξη δυαδικτικών εφαρμογών μιας μόνο σελίδας (SPA , Single Page Application).
- Η εφαρμογή «φτιάχνεται» στην πλευρά του χρήστη, έτσι ώστε να μην επιβαρύνεται ο server.

Πλεονεκτήματα της ReactJs [10]:

- Είναι εύκολη η εκμάθηση της.
- Δίνει την δυνατότητα ανάπτυξης καλαίσθητων και ισχυρών δυνατοτήτων εφαρμογών.
- Επιτρέπει την δημιουργία στοιχείων τα οποία είναι προσαρμοσμένα στον δημιουργό.
- Βοηθάει στην παραγωγικότητα των δημιουργών.
- Υπάρχουν εργαλεία όπως βιβλιοθήκες τα οποία διευκολύνουν αρκετά τον δημιουργό.
- Μεγάλη κοινότητα για την παροχή βοήθειας

2.2.2.2 Angular

Η Angular είναι μία πλατφόρμα που χρησιμοποιείται για την ανάπτυξη διαδικτυακών εφαρμογών χρησιμοποιώντας html και JavaScript. Είναι πλατφόρμα ανοιχτού κώδικα και είναι βασισμένη στην MVC(MODEL – VIEW – CONTROLLER) αρχιτεκτονική. Η MVC αρχιτεκτονική είναι μια αρχιτεκτονική λογισμικού που χρησιμοποιούμε για να φτιάξουμε χώρους επικοινωνίας μεταξύ των χρηστών. Στην αρχιτεκτονική αυτή η εφαρμογή χωρίζεται σε τρία μέρη που συνδέονται μεταξύ τους με σκοπό να κάνει πιο εύκολο το να ελεγχθεί, να αλλάξει και να διαμοιραστεί η πληροφορία που επιθυμούμε να βλέπει ο χρήστης, ανάλογα με τις λειτουργίες που έχει επιλέξει ή γενικότερα ανάλογα με το τι κάνει στην εφαρμογή που χρησιμοποιεί εκείνη την ώρα. Τα τρία μέρη είναι το model, το view και ο controller. Το model είναι υπεύθυνο για την απολαβή και διαχείριση της πληροφορίας στο σύστημα. Το view είναι υπεύθυνο για τον τρόπο που ο χρήστης θα δει την πληροφορία, δηλαδή το κομμάτι που αναφέρουμε ως frontend .Και τέλος ο controller παίρνει δεδομένα από τον χρήστη και τα προωθεί σαν εντολές στο model και στο view.



Σχήμα 2.2 - Αρχιτεκτονική Model-View-Controller [27]

Η Angular μπορεί εύκολα και με διαφορετικούς τρόπους να αλληλεπιδράσει με ένα λογισμικό ενός server, δηλαδή στην περίπτωση μας το backend και συνεπώς και με την βάση δεδομένων. Επίσης, η angular λόγω του ότι προωθεί στον προγραμματιστή να «χτίσει» την εφαρμογή του με έναν συγκεκριμένο τρόπο, χρησιμοποιήθηκε και από μεγάλες εταιρίες για πολύ μεγάλες διαδικτυακές εφαρμογές. Συγκεκριμένα, αυτή η αρχιτεκτονική βοηθάει και συντονίζει πολλά άτομα να γράφουν τον κατάλληλο κώδικα με συγκεκριμένο τρόπο και με συγκεκριμένη σύνταξη. Με τον τρόπο αυτό ασχέτως που ο κώδικας γράφεται από πολλά διαφορετικά άτομα, είναι σταθερός και συνεπής επειδή έχει να ακολουθήσει κάποιος συγκεκριμένους κανόνες. Η επιλογή της angular έμοιαζε αρκετά ενδιαφέρον παρόλο που δεν εκπληρούσε την απαίτηση μας να έχουμε την πλήρη ελευθερία στην διαμόρφωση και στην σύνταξη του κώδικα της εφαρμογής μας. Θα έκανε την συνεργασία πάνω σε αυτήν πιο εύκολη και θα υπήρχε μεγαλύτερη συνενόηση ως προς την κατεύθυνση που θα θέλαμε να πάρει. Επιπλέον στην angular μπορεί κάποιος να γράψει κώδικα JavaScript σε αρχείο html πράγμα ακριβώς αντίθετο από την react στην οποία γράφεις html κώδικα σε αρχείο JavaScript.

Χαρακτηριστικά της Angular [11]:

- Σε οποιαδήποτε αλλαγή που γίνεται στην σελίδα θα ανανεωθούν όλα και όχι μόνο το στοιχείο που άλλαξε.
- Υπάρχει δυνατότητα να χρησιμοποιηθεί η TypeScript.
- Παρέχεται δοκιμαστικό περιβάλλον για την βοήθεια των δημιουργών.

Πλεονεκτήματα της Angular [11]:

- Επιτρέπει στον δημιουργό να φτιάχνει δικά του στοιχεία, όπως τα θέλει, εύκολα
- Επιτρέπει στον δημιουργό να διαμορφώσει την εφαρμογή του με κάποιες υπηρεσίες που του παρέχονται έτοιμες και όταν δεν τις χρειάζεται να τις απορρίψει.
- Είναι ένα ολοκληρωμένο framework.
- Είναι συμβατή με όλους τους περιηγητές και όλα τα λειτουργικά συστήματα.

2.2.2.3 Vue.js

Η Vue είναι ένα framework (πλαίσιο ή πακέτο απο βιβλιοθήκες) το οποίο μας βοηθάει να υλοποιήσουμε διαδικτυακές εφαρμογές. Είναι ένα πλαίσιο ανοιχτού κώδικα κατά το οποίο χρησιμοποιείται η γλώσσα προγραμματισμού JavaScript και είναι αρκετά διαδεδομένο, καθώς συντηρείται μέχρι και σήμερα. Η vue είναι πολύ ευέλικτη ως προς την σύνταξη και την διαμόρφωση του κώδικα της διαδικτυακής εφαρμογής. Η vue είναι αρκετά διαδεδομένη και πολύ μοντέρνα, καθώς

Κεφάλαιο 2

το εγχειρίδιο της είναι πολύ κατανοητό και καποιός μπορεί να μάθει σχετικά γρήγορα πως να χρησιμοποιεί το συγκεκριμένο framework. Μοιάζει σε αρκετά πράγματα την react, όμως η vue είναι ένα αρκετά ολοκληρωμένο framework, ενώ η react είναι μία βιβλιοθήκη. Αυτό σημαίνει ότι η vue παρέχει τον προγραμματιστή περισσότερες λειτουργίες που έχουν φτιαχτεί από την κύρια ομάδα που συντηρεί και αναβαθμίζει την vue και στηρίζεται λιγότερο σε πακέτα και σε βιβλιοθήκες από τρίτους προγραμματιστές. Ο τρόπος προγραμματισμού με vue επιταχύνει αρκετά το «χτίσιμο» της εφαρμογής και επίσης η αλληλεπίδραση με βάσεις δεδομένων και server είναι αρκετά εύκολη.

Χαρακτηριστικά της Vue.JS [12]:

- Όταν αλλάζει ή ενημερώνεται κάτι στην εφαρμογή αυτόματα γίνεται ανανέωση και αλλαγή στο τι βλέπει ο χρήστης και σε αυτό βοηθάει το virtual DOM.
- Κάνει την σύνταξή της JavaScript όπως αυτήν της HTML.
- Επιτρέπει την δημιουργία στοιχείων τα οποία είναι προσαρμοσμένα στον δημιουργό.***
- Παρέχεται η δυνατότητα στον χρήστη να εισάγει δεδομένα και έπειτα το σύστημα κάνει αυτόματους υπολογισμούς.
- Έχει κάποιες λέξεις κλειδιά που μπορούν να κάνουν πολλές ενέργειες στο frontend.
- Για τις αλλαγές που γίνονται στα δεδομένα εισόδου δεν χρειάζεται αλλαγή στον κώδικα.
- Χρησιμοποιώντας το vue-router γίνεται η μεταφορά μεταξύ των σελίδων.
- Είναι πολύ ελαφριά και πολύ γρήγορη.
- Με την χρήση του vue-cli γίνεται εύκολο το «χτίσιμο» και η μετατροπή του πρότζεκτ.

Πλεονεκτήματα της Vue.JS [12]:

- Μικρό μέγεθος.
- Όταν αλλάζει ή ενημερώνεται κάτι στην εφαρμογή αυτόματα γίνεται ανανέωση και αλλαγή στο τι βλέπει ο χρήστης και σε αυτό βοηθάει το virtual DOM.
- Επιτρέπει την αμφίδρομη κίνηση των δεδομένων .
- Έχει την δυνατότητα να ενσωματώνει άλλες εφαρμογές που ήδη υπάρχουν.
- Παρέχει αρκετά εργαλεία.
- Εύκολη εκμάθηση.
- Μεγάλη κοινότητα για την παροχή βοήθειας.
- Η χρήση της αρχιτεκτονικής «headless» είναι εύκολη

2.2.3.4 jQuery

Η jQuery είναι μια βιβλιοθήκη της JavaScript που αποτελεί παλαιότερη τεχνολογία, παρόλο που χρησιμοποιείται ακόμη και σε καινούριες διαδικτυακές εφαρμογές. Η jQuery παρέχει τις λιγότερες λειτουργίες σε σχέση με τις προαναφερθείσες τεχνολογίες. Όμως, η jQuery παρέχει ένα ευκολότερο και πιο κατανοητό τρόπο να γράψεις κώδικα JavaScript. Το αποτέλεσμα είναι η ταχύτερη ανάπτυξη και η πιο αξιόπληστη σύνταξη των εφαρμογών. Η jQuery έχει επίσης χρησιμοποιηθεί σε χίλιαδες πρότζεκτ σε όλο τον κόσμο και βρίσκεται σε πάρα πολλές βιβλιοθήκες που χρησιμοποιούνται ακόμη και στις παραπάνω τεχνολογίες που αναφέρθηκαν. Παρέχει έναν εύκολο τρόπο στους δημιουργούς να γράφουν βιβλιοθήκες πάνω στην jQuery και γι αυτό τον λόγο η υιοθέτηση της είναι και πολύ μεγάλη. Είναι πάρα πολύ διαδεδομένη, λόγω τούτου υπάρχει πάρα πολύ μεγάλη υποστήριξη σε forum, σε μέσα κοινωνικής δικτύωσης, καθώς υπάρχουν πληροφορίες σχετικά με την λειτουργία αυτών σε βιβλία, εγχειρίδια και σε διαδικτυακά έντυπα.

Κεφάλαιο 2

Χαρακτηριστικά της jQuery:

- Δυνατότητα επαναφοράς στοιχείων DOM βάση διαφορετικών χαρακτηριστικών όπως όνομα κτλ.
- Δυνατότητα χειρισμού στοιχείων DOM χρησιμοποιώντας συναρτήσεις της.
- Εφαρμογή εφέ σε στοιχεία DOM.
- Δυνατότητα ανάκτησης πληροφοριών από έναν server δίχως να απαιτείται η φόρτωση όλης της σελίδας, μέσω των συναρτήσεων Ajax.

Πλεονεκτήματα της jQuery :

- Εύκολη εκμάθηση.
- Φιλοσοφία στην οποία γράφοντας λιγότερα μπορείς να κάνεις περισσότερα.
- Επικοινωνία εφαρμογών μεταξύ τους.
- Διαχωρισμός html και jQuery κώδικα.

2.2.3 Καταληκτική επιλογή τεχνολογίας για το Frontend

Η react.js τελικά επιλέχθηκε για την υλοποίηση του frontend στην διαδικτυακή εφαρμογή της παρούσας πτυχιακής εργασίας, καθώς έχει τρία βασικά χαρακτηριστικά τα οποία μας ενδιαφέρουν και περισσότερο. Πρώτον είναι εύκολη στην εκμάθηση και στην χρησιμοποίησή της. Δεύτερον είναι πολύ γρήγορη και επιτρέπει να γράφεις κώδικα JavaScript μέσα σε κώδικα html και τέλος πληροί όλες τις προϋποθέσεις που θέλαμε να έχει. Πιο συγκεκριμένα, η χρήση της JavaScript μέσα σε κώδικα html κάνει ταχύτερη την ανάπτυξη της εφαρμογής, επειδή η σύνταξη της είναι εύκολη. Επιπλέον, η react δίνει την μεγαλύτερη ελευθερία σχετικά με την διαμόρφωση της εφαρμογής. Υπάρχει πολύ εύκολη πρόσβαση σε χιλιάδες πακέτα βιβλιοθηκών που είναι γραμμένες για την react, έτσι ώστε να εκμεταλεύονται την ταχύτητα και την αποτελεσματικότητα της. Επιπρόσθετα, ο κώδικας είναι ανοιχτής πηγής, δηλαδή είναι ελεύθερα διαθέσιμος στο διαδίκτυο και μπορεί να τον δει καθώς και να συμβάλει οποιοσδήποτε στην περαιτέρω ανάπτυξη του λογισμικού της. Συντηρείται και βελτιώνεται με καινούριες εκδόσεις και επιπρόσθετες λειτουργίες πολύ συχνά. Ακόμη, εφόσον χρησιμοποιείται για εφαρμογές μίας σελίδας (SPA , SINGLE PAGE APPLICATION), όταν ο χρήστης πλοηγείται σε διαφορετικά μενού η σελίδα δεν ανανεώνεται, αλλά αντικαθίσταται μόνο τα σημεία που επιθυμεί με νέα δεδομένα. Τέλος, η υποστήριξη για την συγκεκριμένη βιβλιοθήκη είναι τεράστια καθώς, είναι η πιο διαδεδομένη γι αυτό ουσιαστικά προαπαιτείται ως ικανότητα και γνώση για δουλειές που αφορούν ανάπτυξη λογισμικού σε όλο τον κόσμο.

2.3 Backend (Επικοινωνία με τον server και την βάση δεδομένων)

2.3.1 Απαιτήσεις

Αφού πλέον έχει επιλεγεί μία τεχνολογία για την ανάπτυξη της εφαρμογής , παρακάτω ,πρέπει να επιλεγεί και μία διαφορετική τεχνολογία για το λογισμικό του server δηλαδή όπως έχει ήδη αναφερθεί το backend.Οι απαιτήσεις που είχαμε για την τεχνολογία που θέλουμε είναι οι εξής:

- Όσο το δυνατόν μεγαλύτερη ελευθερία στην αρχιτεκτονική και στην γενικότερη διαμόρφωση του λογισμικού.
- Εύκολη δρομολόγηση των endpoints
- Η πηγή του κώδικα να είναι ανοιχτή και η τεχνολογία να βελτιώνεται και να συντηρείται ακόμη.
- Γρήγορη ανάπτυξη του λογισμικού

Κεφάλαιο 2

- Η σύνταξη της γλώσσας προγραμματισμού που θα χρησιμοποιηθεί να συμβάλει στην παραγωγικότητα μας.
- Να υπάρχει όσο μεγαλύτερη υποστήριξη και όσες περισσότερες πληροφορίες γίνεται στο διαδίκτυο, σε βιβλία, σε μέσα κοινωνικής δικτύωσης και σε φόρουμ από προγραμματιστές και ακόμη και σε άτομα που ασχολούνται ερασιτεχνικά.

Έχοντας κάνει εκτεταμένη έρευνα για τις επιλογές που μας παρουσιάζονται στο διαδίκτυο καταλήξαμε σε 3 οι οποίες είναι :

- Express.js
- Flask
- Spring Boot

2.3.2 Τεχνολογίες Backend

2.3.2.1 Express.js

Η Express.js είναι ένα framework, δηλαδή ένα πλαίσιο ανάπτυξης λογισμικού, το οποίο έχει αναπτυχθεί για την πλατφόρμα NODEJS. Η Express.js χρησιμοποιείται για να «φτιαχτεί» ένα γρήγορο και αποτελεσματικό λογισμικό διαδικτυακής εφαρμογής το οποίο θα μπορεί να δέχεται πολλά και διαφορετικά http αιτήματα σε ένα ή περισσότερα endpoint. Χρησιμοποιείται σε εφαρμογές πραγματικού χρόνου, σε περιπτώσεις επικοινωνίας μεταξύ server και browser ή ακόμη και σε δημιουργία ολοκληρωμένης διαδικτυακής εφαρμογής, καθώς είναι πολύ εύκολο να χρησιμοποιηθεί η αρχιτεκτονική MVC. Έτσι με την βοήθεια κάποιου περιγράμματος HTML (HTML TEMPLATE) μπορεί να δημιουργηθεί το «VIEW» κομμάτι της αρχιτεκτονικής MVC και να απεικονίζεται στον χρήστη η εφαρμογή με σωστό τρόπο. Η χρήση της αρχιτεκτονικής MVC δεν είναι απαραίτητη καθώς όπως προαναφέρθηκε η express μπορεί να χρησιμοποιηθεί ως αυτόνομος server για να δεχτεί HTTP αιτήματα από εφαρμογές, συσκευές κτλ. Είναι ένα πάρα πολυ εύελικο framework και αποτελεί η πρώτη επιλογή για πάρα πολλές διαδικτυακές εφαρμογές. Για παράδειγμα χρησιμοποιείται στην ιστοσελίδα της γνωστής εταιρείας Nike για να δώσει δεδομένα στο frontend. Είναι μία αρκετά ώριμη τεχνολογία, γι' αυτό τον λόγο η βελτίωση της είναι πλέον αργή, αλλά όχι μηδαμινή. Η ευελιξία που παρέχει τον προγραμματιστή να συντάξει την εφαρμογή του όπως θέλει, είναι πολύ σημαντική διότι αυξάνει την παραγωγικότητα του. Η Express δεν παρέχει πάρα πολλές λειτουργίες όπως ένα εντελώς ολοκληρωμένο framework γιατί στοχεύει στην απλοϊκότητα και δίνει την επιλογή στον δημιουργό της εφαρμογής να επιλέξει ένα ή μία από τα/τις χιλιάδες πακέτα/βιβλιοθήκες που είναι διαθέσιμα από τρίτους δημιουργούς για κάθε λειτουργία που φαντάζεται ο καθένας ή ακόμη να φτιάξει και τα δικά του, για τις δικές του ανάγκες. Το framework αυτό χρησιμοποιεί την γλώσσα προγραμματισμού JavaScript.

Χαρακτηριστικά της express js [13] :

- Γρήγορο «χτίσιμο» του server.
- Χρήση ενδιάμεσων λογισμικών (Middlewares)
- Δρομολόγηση (Routing)
- Μηχανισμός που ανιχνεύει και δείχνει σε ποιο σημείο υπάρχει λάθος στην εφαρμογή.

Πλεονεκτήματα της express js [13] :

- Εύκολη διαμόρφωση και προσαρμογή.

Κεφάλαιο 2

- Δυνατότητα ορισμού διαδρομών της εφαρμογής με βάση HTTP μεθόδους και URLs.
- Περιέχει διάφορες μονάδες ενδιαμέσου λογισμικού τις οποίες μπορείς να χρησιμοποιήσεις για να αναδείξεις εξτρά εργασίες στα αιτήματα και της απαντήσεις.
- Καθορισμός σφαλμάτων χειρισμού ενδιαμέσου λογισμικού.
- Εύκολη σύνταξη για την δημιουργία και χρήση ενδιάμεσων λογισμικών
- Εύκολη δημιουργία REST API server.
- Μεγάλη υποστήριξη κοινότητας
- Είναι ευρέως διαδεδομένη
- Δίνεται η επιλογή στον προγραμματιστή να συντάξει την εφαρμογή του με όποιον τρόπο θέλει.
- Θετική η χρήση της JavaScript διότι τα περισσότερα framework του frontend χρησιμοποιούν την ίδια γλώσσα, οπότε ο προγραμματιστής δεν χρειάζεται να μάθει επιπλέον γλώσσες προγραμματισμού
- Ευκολία σύνδεσης με βάσεις δεδομένων όπως η Mongo DB, η Redis, η MySQL με πακέτα από τρίτους δημιουργούς.

2.3.2.2 Spring Boot

Το Spring Boot είναι ένα framework ανοιχτής πηγής κώδικα το οποίο βασίζεται στην Java. Το Spring Boot χρησιμοποιείται πολύ από την κοινότητα των προγραμματιστών και γενικότερα από τις εταιρίες που ασχολούνται με διαδικτυακά προϊόντα, ιστοσελίδες κλπ. Τα κύρια θετικά του συγκεκριμένου πλαισίου προγραμματισμού είναι ότι βοηθάει την πολύ γρήγορη ανάπτυξη της ιδέας. Αυτό επιτυγχάνεται διότι το framework για να λειτουργήσει δεν χρειάζεται πολλές ρυθμίσεις από τον δημιουργό, αλλά είναι σε πολύ μεγάλο βαθμό αυτορυθμιζόμενο. Βέβαια, ταυτόχρονα δίνεται και η δυνατότητα αλλαγής ρυθμίσεων που θέλει ο δημιουργός. Δηλαδή εκτός από την βοήθεια της αυτορύθμισης, το framework σου δίνει και ένα μεγάλο βαθμό ελευθερίας να ορίσεις σε πολλές λειτουργίες τον τρόπο λειτουργίας τους. Το spring boot είναι αρκετά έξυπνο στο να μπορεί να αποφασίσει ποια πακέτα ή ποιες βιβλιοθήκες πρέπει να εγκαταστήσει αυτόματα ανάλογα με τις ρυθμίσεις ή λειτουργίες χρησιμοποιεί ο προγραμματιστής της εφαρμογής. Επιπρόσθετα δίνεται και η δυνατότητα να «φτιαχτεί» μια ολόκληρη εφαρμογή, δηλαδή όπως και η express, το spring boot δεν είναι υποχρεωτικό να χρησιμοποιηθεί ως μόνο ξεχωριστό server. Το spring boot είναι πολύ διάσημο, ως εκ τούτου, υπάρχει πολύ μεγάλη διαδικτυακή κοινότητα για υποστήριξη και βοήθεια για οποιοδήποτε πρόβλημα προκύψει κατά την ανάπτυξη εφαρμογής. Ακόμη, επειδή χρησιμοποιείται η γλώσσα προγραμματισμού java, παρέχονται λειτουργίες της java που καθιστούν την αλληλεπίδραση με βάσεις δεδομένων πολύ εύκολη. Για αυτόν τον λόγο πολλές μεγάλες εταιρίες έχουν δημιουργήσει cloud, βάσεις δεδομένων που αφορούν την spring boot και αυτές μπορούν να χρησιμοποιηθούν από νέες ή παλιές εφαρμογές κατά την βελτίωση τους ή για την τελική έκδοσή τους. Δυστυχώς όμως, ο τρόπος που πρέπει να γράφεται ο κώδικας, η δομή της εφαρμογής και οι κανόνες που πρέπει να ακολουθούνται δεν είναι πολύ εύκολοι ως προς την εκμάθησή τους και χρειάζεται να αφιερωθεί κάποιος χρόνος έτσι ώστε να κατανοηθούν πλήρως. Όμως, όταν συμβεί το παραπάνω το spring boot αποτελεί μία εξαιρετική επιλογή και δεν γίνεται να αγνοηθεί για την ανάπτυξη οποιασδήποτε εφαρμογής.

Χαρακτηριστικά της java spring boot [14]:

- Δρομολόγηση (Routing)

Κεφάλαιο 2

- Δυνατότητα τηλεχειρισμού της εφαρμογής.(admin features)
- Ασφάλεια.

Πλεονεκτήματα της java spring boot [14]:

- Ευκολία στο να χτιστεί ένας αυτόνομος HTTP server.
- Δημιουργία εφαρμογών χωρίς πολύ κόπο.
- Γρήγορη ανάπτυξη των εφαρμογών.
- Δουλεύει πολύ καλά με ενσωματωμένες και με in-memory βάσεις δεδομένων.

2.3.2.3 Flask

Το flask είναι ένα «ελαφρύ» και μικρό διαδικτυακό πλαίσιο της python το οποίο δίνει εργαλεία και χαρακτηριστικά που είναι χρήσιμα για την δημιουργία μιάς διαδικτυακής εφαρμογής με εύκολο τρόπο. Παρέχει στους δημιουργούς την δυνατότητα να ελιχθούν και είναι κατάλληλη για νέους προγραμματιστές, καθώς μπορεί κάποιος να χτίσει μια διαδικτυακή εφαρμογή εύκολα και σε σύντομο χρονικό διάστημα, χρησιμοποιώντας μόνο ένα αρχείο python. Δίνεται η δυνατότα να χρησιμοποιηθούν επιπρόσθετα πακέτα έτσι ώστε η διαδικτυακή εφάρμογη να βρίσκεται σε μία κατάσταση που την καθιστά εύκολα να συντηριθεί ή ακόμα και να εξελιχθεί. Συνήθως το flask δεν φαίνεται να χρησιμοποιείται σε εφαρμογές με πάρα πολύ μεγάλο όγκο επισκεψιμότητας. Όμως ένα τα πολύ μεγάλα πλεονεκτήματα που έχει είναι οτι χρησιμοποιεί την γλώσσα προγραμματισμού python. Η python είναι μία γλώσσα προγραμματισμού που μπορεί ακόμη και ένας αρχάριος να καταλάβει σε σχετικά σύντομο χρονικό διάστημα. Αυτό συμβαίνει διότι η σύνταξη της είναι εύκολη και κατανοητή. Αυτό έχει ως αποτέλεσμα η ταχύτητα ανάπτυξης της διαδικτυακής εφαρμογής να είναι πολύ γρήγορη. Όπως και στις παραπάνω τεχνολογίες η υποστήριξη που μπορεί να βρει κάποιος στο διαδίκτυο, βιβλία κτλπ είναι πάρα πολύ μεγάλη. Στο Flask επίσης η ρύθμιση λειτουργιών είναι εύκολη και γίνεται πολύ γρήγορα. Με την χρήση του, έρχονται μαζί του και δύο σημαντικά χαρακτηριστικά/λειτουργίες. Αρχικά, υπάρχει προεγκατεστημένο δοκιμαστικό περιβάλλον για την ανάπτυξη της εφαρμογής, έτσι ο δημιουργός δεν χρειάζεται να εγκαταστήσει πακέτα από τρίτες πηγές για να δημιουργήσει το δικό του δοκιμαστικό περιβάλλον και στη συνέχεια παρέχονται εύκολα περιγράμματα HTML για δημιουργηθούν ολοκληρώμενες εφαρμογές. Η αλληλεπίδραση με βάσεις δεδομένων πρέπει να γίνει με επιπρόσθετα πακέτα που πρέπει να εγκαταστήσει ο χρήστης, δηλαδή όπως την expressjs. Αυτό συμβαίνει γιατί το Flask σχεδιάστηκε να είναι όσο πιο ελαφρύ γίνεται και έτσι οι λειτουργίες που παρέχονται από την κύρια εγκατάσταση του framework δεν είναι πάρα πολλές. Με τον τρόπο αυτό ο δημιουργός έχει την δυνατότητα να επέλεξει τα επιπλέον εργαλεία που θέλει να χρησιμοποιήσει ελεύθερα, χωρίς κανένα περιορισμό.

Χαρακτηριστικά της flask python [15]:

- Εσωτερικός server και γρήγορη ανίχνευση λαθών.
- Πλήρης βοήθεια για τον έλεγχο λειτουργικότητας.
- Στηρίζεται στο Unicode.
- Δρομολόγηση (Routing)

Πλεονεκτήματα της flask python [15]:

- Ευκολία στο να χτιστεί ένας αυτόνομος HTTP server.
- Συμβατότητα με τις νέες τεχνολογίες

Κεφάλαιο 2

- Μπορεί να κάνει πολλά πράγματα με διαφορετικούς τρόπους.
- Δυνατότητα εξέλιξης απλών εφαρμογών.
- Ευκολία στην χρήση.

2.3.3 Καταληκτική επιλογή τεχνολογίας για το Backend

Αφού έγινε μελέτη των παραπάνω επιλογών καταλήξαμε να χρησιμοποιήσουμε την Express.js που είναι βασισμένη στην JavaScript. Το πιο ισχυρό πλεονέκτημα για εμάς, ήταν η γλώσσα προγραμματισμού που έπρεπε να χρησιμοποιήσουμε. Δηλαδή η χρήση της JavaScript έμοιαζε να ήταν η πιο λογική επιλογή διότι χρησιμοποιείται η ίδια γλώσσα και στην ανάπτυξη του frontend κομματιού της εφαρμογής. Έτσι θα μπορούσαμε ουσιαστικά να χρησιμοποιήσουμε την ίδια σύνταξη, να μην ασχοληθούμε με νέα γλώσσα προγραμματισμού και να χρησιμοποιήσουμε όλες τις λειτουργίες που παρέχει η JavaScript και που ξέρουμε ήδη. Ακόμη λόγω της ελευθερίας που παρέχει στον δημιουργό αυτό το framework μπορέσαμε να βρούμε βιβλιοθήκη και πακέτα από βιβλιοθήκες για ό,τι λειτουργίες θέλαμε να προσθέσουμε στην εφαρμογή. Το μόνο αρνητικό, ήταν ότι από μόνη της η express.js δεν παρείχε διεπαφή για την σύνδεση του λογισμικού με την βάση δεδομένων. Όμως αυτό λύθηκε εύκολα διότι μπορέσαμε να βρούμε με μεγάλη ευκολία μία τέτοια βιβλιοθήκη που έχει φτιαχτεί και συντηρείται από τρίτους δημιουργούς. Αυτή ονομάζεται Prisma και μας παρέχει την δυνατότητα να συνδεθούμε με την βάση δεδομένων και να γράψουμε λειτουργίες γι' αυτήν με πολύ εύκολο τρόπο. Εκτός από αυτό η δημιουργία του λογισμικού δηλαδή η δρομολόγηση, οι λειτουργίες του RESTFUL server, όπως οι απαντήσεις που δίνει το λογισμικό σε HTTP αιτήματα κτλπ, ήταν ομαλή και γρήγορη.

2.4 Επίλογος

Λόγω των πληθώραν τεχνολογιών για την δημιουργία μιας διαδικτυακής εφαρμογής έγινε εκτεταμένη έρευνα για την επιλογή αυτών που ταίριαζαν περισσότερα στις δικές μας απαιτήσεις. Τελικά, επιλέχθηκε να χρησιμοποιηθεί η React.js για το παρουσιαστικό της εφαρμογής και η Express.js για την δημιουργία λογισμικού του server. Τα πιο σημαντικά χαρακτηριστικά τους είναι ότι χρησιμοποιούν την γλώσσα προγραμματισμού JavaScript και έτσι δεν χρειάστηκε να γράψουμε σε δύο διαφορετικές γλώσσες. Είναι εύκολες στην εκμάθηση τους και υπάρχουν πολλές διαθέσιμες λύσεις και γενικότερα πληροφορίες στο διαδίκτυο.

Κεφάλαιο 3ο: Χρήση λογισμικού πραγματικού χρόνου στην συσκευή

3.1 Εισαγωγή

Τελειώνοντας πλέον, με τις επιλογές των τεχνολογιών και γενικά των βοηθητικών εργαλείων για την ανάπτυξη την διαδικτυακής εφαρμογής και της ανάπτυξη του λογισμικού του HTTP RESTFUL server, έπρεπε πλέον να γίνει έρευνα και να αποφασιστεί ποια πλατφόρμα λογισμικού θα χρησιμοποιούσαμε για να φτιάξουμε την συσκευή μέτρησης των βιομετρικών. Η ιδέα και ο στόχος μας είναι η συσκευή να ανταποκρίνεται όσο πιο γρήγορα γίνεται στις ενέργειες του χρήστη της, το περιβάλλον της εφαρμογής να είναι εύκολο στην πλοήγηση και γενικότερα ο χρήστης να νιώθει ήδη από την πρώτη στιγμή οικεία με το λογισμικό. Άρα το λογισμικό έπρεπε να περιέχει χαρακτηριστικά όπως αυτά των εφαρμογών ενός κινητού τηλεφώνου και ταυτόχρονα να είναι όσο πιο φθηνή γίνεται έτσι ώστε να φτιαχτεί εύκολα. Έτσι η στροφή προς τα εργαλεία με ανοιχτή, ελεύθερη πηγή του κώδικα ήταν η λογική κίνηση.

3.2 RTOS (Real Time Operating System)

Όπως αναφέρθηκε, στην αρχιτεκτονική του συστήματος διαλέξαμε να χρησιμοποιήσουμε λογισμικό RTOS (Real Time Operation System). Υπάρχουν διάφορα τέτοια λογισμικά που μπορεί να επιλέξει ένας δημιουργός να αναπτύξει την εφαρμογή του. Το RTOS προσφέρει πολλές και διαφορετικές λύσεις σε προβλήματα που προκύπτουν σε περίπλοκες εφαρμογές. Επειδή η συσκευή πρέπει να μπορεί να εκτελέσει πολλές και διαφορετικές λειτουργίες ταυτόχρονα, η πολυπλοκότητα αυξάνεται, έτσι είναι λογικό να έρθουμε αντιμέτωποι με αυτά τα προβλήματα. Κάποια από αυτά τα προβλήματα είναι:

- Η ταυτόχρονη λειτουργία πολλών υπηρεσιών που προσφέρει η συσκευή.
- Συνδυασμός των υπηρεσιών.
- Διαμόρφωση του κώδικα με σωστή δομή.

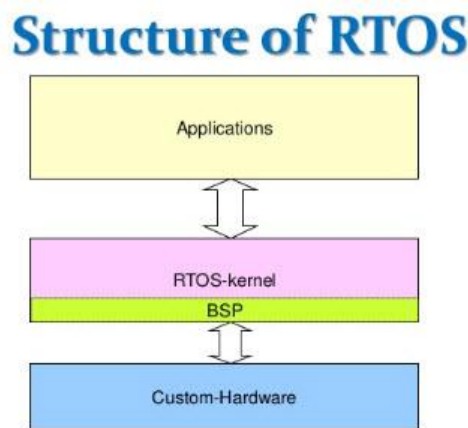
Υπάρχουν RTOS μικρού μεγέθους που προσφέρουν τα απολύτως απαραίτητα εργαλεία στον προγραμματιστή για να προχωρήσει στην ανάπτυξη λογισμικού με γενικότερα ταχύτερο ρυθμό από αυτόν χωρίς την χρήση του. Υπάρχουν RTOS που κατέχουν πιστοποίηση και εγγυώνται κάποιο βαθμό ασφάλειας στα εργαλεία που περιέχονται σε αυτό, τέτοια RTOS χρησιμοποιούνται για παράδειγμα σε μηχανήματα για ιατρική χρήση όπου η συσκευή πρέπει να λειτουργήσει οποσδήποτε σώστα έτσι ώστε να μην προκαλέσει κανένα πρόβλημα σε κάποιον ασθενή. Ακόμη υπάρχουν RTOS που χρησιμοποιούνται σε συσκευές όπως τα κινητά τηλέφωνα, που προσφέρουν παραπάνω λειτουργίες, δηλαδή εργαλεία για εύκολο προγραμματισμό ολοκληρωμένων SOM (system on module) ή SOC (system on chip) για σύνδεση σε δίκτυα ή σε άλλες συσκευές, όπως για παράδειγμα με την χρήση Bluetooth ή BLE (Bluetooth Low Energy) αντίστοιχα για συσκευές όπου η μικρή κατανάλωση ενέργειας είναι σημαντική.

Ουσιαστικά το RTOS είναι ένα λειτουργικό σύστημα πραγματικού χρόνου και δουλειά του είναι να μπορεί να εκτελεί πολλές εργασίες εντός πλαισίου μιας καθυστέρησης που θεωρείται αποδεκτή. Το αποτύπωμα που αφήνει στην μνήμη ενός επεξεργαστή/μικροελεγχτή δηλαδή στην μνήμη τους, είναι πολύ μικρότερο από αυτό ενός ολοκληρωμένου συστήματος, όπως για παράδειγμα αυτών των Linux ή των Windows. Συνήθως ένα ολοκληρωμένο λειτουργικό σύστημα OS (Operating System) περιέχει οδηγούς για συσκευές όπως ένα πληκτρολόγιο ή ένα ποντίκι, καθώς και πολλών ακόμη συσκευών ή

Κεφάλαιο 3

υπηρεσιών του λογισμικού όπως το file system, γι αυτό τον λόγο το αποτύπωμα στην μνήμη είναι και πολλαπλά μεγαλύτερο από αυτό του RTOS. Έτσι, το RTOS είναι η κατάλληλη λύση για χρήση σε μικροελεγχτές με λίγους πόρους, δηλαδή για παράδειγμα σε μνήμες μερικών δεκάδων mb (megabytes) ή ακόμη και kb (kilobytes), αντί χιλιάδων mb όπως σε έναν ολοκληρωμένο λειτουργικό υπολογιστή οικιακής χρήσης. Πάραυτα αυτά γίνεται εκτεταμένη χρήση και σε συσκευές με πάρα πολλούς πόρους, γιατί κανένα ολοκληρωμένο λειτουργικό σύστημα δεν δίνει κάποια εγγύηση για το πλαίσιο που θα κυμαίνεται η χρονική καθυστέρηση κάποιας λειτουργίας. Αυτό είναι και το πιο σημαντικό πλεονέκτημα του RTOS, γι αυτό κιόλας γίνεται χρήση του σε εκατομμύριες συσκευές και προϊόντα, από αυτοκίνητα και αεροπλάνα μέχρι και κάτι απλό όπως μια συσκευή οικιακής χρήσης. ένα πλυντήριο ή έναν φούρνο μικροκυμάτων.

Αν δει κάποιος όσο πιο απλά γίνεται από τι αποτελείται ένα απλό σύστημα που χρησιμοποιεί RTOS θα καταλάβει ότι ουσιαστικά χρησιμοποιείται ανάμεσα στην εφαρμογή και το hardware που υπάρχει μέσα στην συσκευή. Δηλαδή είναι μία διεπαφή που συνδέει τα παραπάνω. Με τον ίδιο τρόπο ακριβώς χρησιμοποιήθηκε και στην ανάπτυξη της συσκευής για την μέτρηση το βιομετρικών.



Σχήμα 3.1- Structure of RTOS [28]

Οπώς φαίνεται στο σχήμα 3.1, στην περίπτωση μας το «custom hardware» είναι η ίδια η συσκευή που σχεδιάσαμε, το «RTOS-kernel» είναι το RTOS που χρησιμοποιήσαμε και το «Applications» είναι η εφαρμογή που αναπτύξαμε για να χρησιμοποιείται στην συσκευή. Ένα απλό παράδειγμα για να κατανοηθεί καλύτερα είναι ο τρόπος που επικοινωνεί το πληκτρολόγιο με τον υπολογιστή: το «Custom Hardware» είναι το πληκτρολόγιο, το «kernel» τα windows που είναι εγκατεστημένα στον υπολογιστή και τέλος το «Applications», οι εφαρμογές ή τα προγράμματα που τρέχουν πάνω στον λειτουργικό σύστημα. Επομένως, καταλήγουμε στο συμπέρασμα ότι το «kernel» είναι η πλατφόρμα στην οποία γίνεται η ανάπτυξη των εφαρμογών. Αυτή η πλατφόρμα μας παρέχει ορισμένες λειτουργίες που παίρνουμε δεδομένες, για παράδειγμα το κουμπί που πατάμε για να κλείσει ένα πρόγραμμα στα windows ή συγκεκριμένα για τα RTOS, η εγγύηση τους ότι μία λειτουργία δεν θα αργήσει να τρέξει πάνω από κάποιο χρονικό όριο.

Έχοντας λάβει υπόψιν όλα τα παραπάνω ψάξαμε να βρούμε όσες περισσότερες διαθέσιμες επιλογές είχαμε σχετικά με το ποιο RTOS θα μπορούσαμε να χρησιμοποιήσουμε. Η επιλογή του σωστού RTOS είχε μεγάλη σημασία γιατί διαδραματίζει καθοριστικό ρόλο στην διαμόρφωση ολόκληρης την εφαρμογής που τρέχει στην συσκευή που αργότερα φτιάξαμε. Τα RTOS που μας τράβηξαν περισσότερο την προσοχή ήταν δύο:

- FreeRTOS
- Zephyr RTOS

3.3 Διαθέσιμα λογισμικά RTOS

3.3.1 FreeRTOS

Το FreeRTOS είναι ένα πολύ διάσημο RTOS στην κοινότητα των προγραμματιστών που ασχολούνται κυρίως με ανάπτυξη λογισμικού πιο κοντά στα στοιχεία που αποτελούν μια συσκευή, όπως ένας μικροελεγχτής. Το συγκεκριμένο RTOS έχει αρκετά χαμηλό αποτύπωμα στην μνήμη. Η πηγή του κώδικα είναι ανοιχτή και μπορεί να χρησιμοποιηθεί στην ανάπτυξη προϊόντων δωρεάν, δηλαδή δεν υπάρχει κάποια άδεια που προαπαιτείται να αγοραστεί. Είναι γραμμένο στην γλώσσα προγραμματισμού C όπως τα περισσότερα, διότι η συγκεκριμένη γλώσσα είναι κυρίαρχη στον κλάδο αυτό επειδή έχει δοκιμαστεί σε εκατομύρια εφαρμογές, συσκευές και γενικότερα στην ανάπτυξη λογισμικού. Η ταχύτητα που προσφέρει είναι σχεδόν μη συγκρίσιμη με υπόλοιπες γλώσσες προγραμματισμού. Επίσης το FreeRTOS παρέχει πολλές ρυθμίσεις στον προγραμματιστή του λογισμικού έτσι ώστε να έχει μεγάλο έλεγχο στην δομή της εφαρμογής. Τα εργαλεία που προσφέρει είναι τα απολύτως βασικά που αποτελούν ένα λειτουργικό σύστημα πραγματικού χρόνου. Αυτό σημαίνει ότι από μόνο του δεν παρέχει λειτουργίες για δημιουργία λογισμικού που αφορά για παράδειγμα την σύνδεση μιας συσκευής σε κάποιο δίκτυο και την μεταφορά πακέτων με δεδομένα μέσω της σύνδεσης αυτής. Όμως δίνεται η δυνατότητα να προσθεθούν τέτοιες λειτουργίες με την εγκατάσταση του RTOS+, όπου ουσιαστικά είναι το βασικό κομμάτι του λογισμικού με επιπλέον λειτουργίες. Αυτό είναι πολύ θετικό διότι αν κριθεί ότι τέτοιες λειτουργίες δεν χρειάζονται μπορεί να μην γίνει η εγκατάσταση τους και έτσι να μην επιβαρυνθούν χωρίς λόγο παραπάνω οι πόροι που έχει ο μικροελεγχτής. Ακόμη το FreeRTOS παρέχει εργαλεία που χρησιμοποιούνται κατά τη διάρκεια της ανάπτυξης της εφαρμογής. Για παράδειγμα έχει εργαλεία που εντοπίζουν αν κάποια λειτουργία έχει κολλήσει λόγω λάθος δομής του κώδικα ή ειδοποιεί αν υπάρξει υπερχειλίση της προσωρινής μνήμης για τον παραπάνω λόγο. Όλα αυτά είναι πολύ σημαντικά για τον προγραμματιστή, διότι πολύ απλά διευκολύνεται η δουλειά του.

Χαρακτηριστικά του FreeRTOS [16]:

- Δυνατότητα πολλών ρυθμίσεων για το λειτουργικό σύστημα ανάλογα με τις απαιτήσεις της εφαρμογής.
- Χρήσιμα API για την σωστή επεξεργασία των εργασιών.
- Mutex και Semaphore για συγχρονισμό.
- Χρονοδιακόπτες λογισμικού.
- Διαχείριση μνήμης για διατήρηση της μνήμης Heap .

Πλεονεκτήματα του FreeRTOS [16]:

- Αξιοπιστία.
- Συνεχής εξέλιξη.
- Προσφέρει πολλές λειτουργίες και πολλά εργαλεία.
- Είναι δωρεάν για όλους ακόμα και για εμπορικούς σκοπούς.
- Μέσω του Safe RTOS παρέχεται ασφάλεια στις εφαρμογές που χρειάζεται.
- Χρησιμοποιείται από πολλούς ανθρώπους.
- Διαθέτει ένα φόρουμ που ο κάθε χρήστης μπορεί να ζητήσει βοήθεια.

Κεφάλαιο 3

- Παρέχει αρκετές δυνατότητες επέκτασης και δεν είναι δύσχρηστο.

3.3.2 Zephyr RTOS

Το Zephyr αποτελεί κι αυτό ένα λειτουργικό σύστημα πραγματικού χρόνου (RTOS). Το Zephyr όπως και το FreeRTOS είναι γραμμένο στην γλώσσα προγραμματισμού C για τους λόγους που αναφέρθηκαν παραπάνω. Ανήκει στην κατηγορία των RTOS που προσφέρουν παραπάνω από τα βασικά συνηθισμένα εργαλεία που αποτελείται ένα RTOS. Η πηγή του κώδικα του είναι ανοιχτή και γι αυτόν τον λόγο τα τελευταία χρόνια η υιοθέτησή του έχει μεγαλώσει πολύ και έτσι η συνεισφορά στην ανάπτυξη του από προγραμματιστές είναι εξίσου μεγάλη. Το Zephyr στοχεύει να γίνει ένα από τα πιο ασφαλή real time operating system. Επομένως, φυσικά περιέχει πολλές λειτουργίες που αφορούν την ασφάλεια. Ακόμη, όπως στο FreeRTOS, η ρύθμιση του για την προσαρμογή του στο λογισμικό είναι πολύ ευέλικτη. Το Zephyr ουσιαστικά παρέχει ό,τι λειτουργίες παρέχει και το FreeRTOS και το FreeRTOS+ μαζί. Έτσι δεν είναι και το πιο ελαφρύ RTOS που υπάρχει. Όμως η επιλογή του εξαρτάται από τι είδους εφαρμογή θα αναπτυχθεί από τους προγραμματιστές. Αν χρειάζονται λειτουργίες όπως σύστημα για αρχεία ή λογισμικό για Bluetooth, η ασφάλεια του λογισμικού είναι κρίσιμη και το Zephyr αποτελεί σοβαρή επιλογή.

Χαρακτηριστικά του Zephyr [17] :

- Προσφέρει πολλές υπηρεσίες ανάπτυξης.
- Η εφαρμογή χρησιμοποιεί μόνο τα απαραίτητα στοιχεία για να λειτουργήσει.
- Μπορεί να χρησιμοποιηθεί από πολλές πλατφόρμες.
- Προστασία της μνήμης.
- Υποστηρίζει την έκδοση Bluetooth 5.0.

Πλεονεκτήματα του Zephyr [17]:

- Γραφική αναπαράσταση των αποτελεσμάτων
- Υπάρχει δυνατότητα εμφάνισης σε word και excel των αποτελεσμάτων των δοκιμών.
- Δυνατότητα δημιουργίας προσαρμοσμένων φίλτρων.
- Δυνατότητα δημιουργίας πολλών δοκιμαστικών επαναλήψεων.

3.4 Επιλογή Λογισμικού RTOS

Λαμβάνοντας υπόψιν τα χαρακτηριστικά των RTOS που μας ενδιέφεραν οδηγηθήκαμε να επιλέξουμε ένα από τα δύο. Όμως παρατηρήσαμε ότι συνήθως οι εταιρίες που παράγουν έναν μικροελεγχτή παρέχουν βιβλιοθήκες γι αυτόν για τον πιο βολικό και πιο εύκολο προγραμματισμό του βασισμένες σε κάποιο RTOS, δηλαδή ήταν συμβατές με ένα από αυτά και όχι με όλα. Βέβαια δεν είναι υποχρεωτικό να τις χρησιμοποιήσουμε, αλλά παρέχουν πολύ μεγάλη βοήθεια και κάνει όλη την διαδικασία της ανάπτυξης της εφαρμογής πολύ πιο γρήγορη. Επομένως, καταλήξαμε στο γεγονός ότι θα δουλεύαμε με το RTOS που θα ήταν συμβατό με αυτές τις βιβλιοθήκες ανάλογα με ποια πλατφόρμα μικροελεγχτή θα επιλέγαμε. Έτσι, συνεχίσαμε στην έρευνα των πιο διάσημων τέτοιων πλατφορμών.

3.5 Επίλογος

Η χρήση λειτουργικού συστήματος πραγματικού χρόνου κρίνεται αναγκαία για την δημιουργία της συσκευής. Η χρήση πολλών λειτουργιών ταυτόχρονα όπως για παράδειγμα αυτή του WiFi και η πλοήγηση του χρήστη στο μενού αποτελούν σημαντικές προϋποθέσεις για την ορθή λειτουργία του

Κεφάλαιο 3

λογισμικού. Αξιοσημείωτες επιλογές RTOS (real time operating system) αποτελούν η FreeRtos και η Zephyr. Η κύρια προϋπόθεση που πλυρούν είναι εκτέλεση πολλών εργασιών εντός πλαισίου μιας καθυστέρησης που θεωρείται αποδεκτή. Η τελική επιλογή του λογισμικού εξάρταται κυρίως από ποια υποστηρίζει ο κατασκευαστής. Αν για παράδειγμα ο κατασκευαστής υποστηρίζει άμεσα την χρήση FreeRtos στην πλατφόρμα ανάπτυξης που περιέχει τον μικροελεγχτή, τότε η επιλογή αυτού κάνει πιο εύκολη και γρήγορη την ανάπτυξη λογισμικού στο επίπεδο της εφαρμογής.

Κεφάλαιο 4ο: Ανάπτυξη συσκευής για τις μετρήσεις

4.1 Εισαγωγή

Πλέον, αφού γνωρίζουμε πως μας βοηθάει ένα λειτουργικό σύστημα πραγματικού ελέγχου πρέπει να βρεθεί μια πλατφόρμα με μικροελεγκτή για να αναπτυχθεί η συσκευή μέτρησης βιομετρικών και τελικά να μάθουμε με ποιο RTOS θα δουλέψουμε. Η πλατφόρμα αυτή διαδραματίζει σημαντικό ρόλο γιατί ουσιαστικά μας παρέχει τον εγκέφαλο της συσκευής, που είναι ο υπολογιστής. Συνήθως, η επιλογή του μικροελεγκτή γίνεται ανάλογα με τις απαιτήσεις που έχει ο δημιουργός για την εφαρμογή που θέλει να αναπτύξει. Αν δεν υπάρχει ακριβές πλάνο για το τι πρέπει να περιέχει αυτή η συσκευή, ο μικροελεγκτής είναι καλό να μας παρέχει περισσότερους πόρους από αυτούς που έχουμε υπολογίσει. Για παράδειγμα αν υπολογιστεί ότι η συσκευή χρειάζεται πέντε εισόδους και πέντε εξόδους, καλό θα είναι ο μικροελεγκτής να έχει μεγαλύτερο νούμερο από αυτές. Έτσι, υπάρχει χώρος για μελλοντική βελτίωση αλλά και χώρος και για οποιαδήποτε αλλαγή προκύψει. Αν όμως υπάρχει με σιγουριά το ακριβές πλάνο για την συσκευή μπορεί να γίνει επιλογή μικροελεγκτή που να πληροί ακριβώς αυτές τις προϋποθέσεις και τίποτα παραπάνω. Με αυτόν τον τρόπο γίνεται και η μέγιστη βελτιστοποίηση του κόστους παραγωγής της συσκευής, που συνήθως είναι και ο ποιο σημαντικός παράγοντας που απασχολεί έναν μηχανικό.

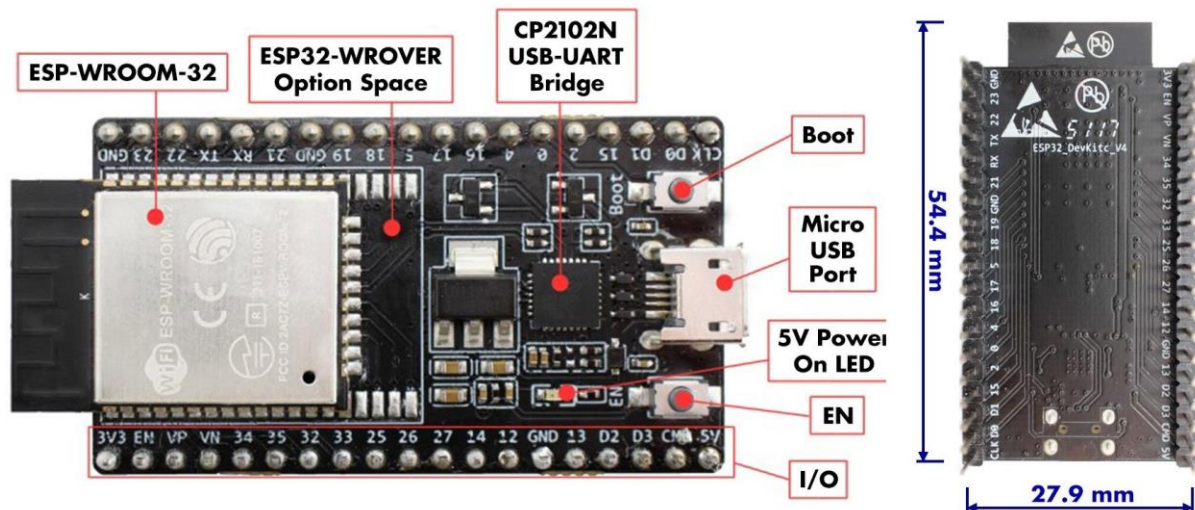
Αρχικά, κάναμε έρευνα για το ποιες πλατφόρμες μικροελεγκτών κυριαρχούν στην αγορά για την ανάπτυξη πρωτοτύπου ενός προϊόντος, έχοντας στο μυαλό μας ότι δεν θα χρειαστούμε πολλές εισόδους και εξόδους, άλλα ίσως χρειαζόμασταν πολλές διεπαφές για να συνδεθούν αισθητήρες έτσι ώστε να επικοινωνούν με τον μικροελεγκτή με κάποιο από τα βασικά πρωτοκόλλα. Οι πλατφόρμες που αποφασίσαμε να ασχοληθούμε είναι οι εξής:

- ESP32 (DEVKITC)
- STM32 (NUCLEO BOARD)
- ARDUINO (UNO R3)

4.2 Πλατφόρμες ανάπτυξης ηλεκτρονικής συσκευής με μικροελεγκτή

4.2.1 ESP32

Το ESP32 είναι μια πλατφόρμα ανάπτυξης συσκευής που αποτελείται από τον μικροελεγκτή, μια θύρα USB για τον εύκολο προγραμματισμό της, ένα κύκλωμα μετατροπής τάσης από 3V έως και 12V σε 3.3V διότι, αυτή είναι η τάση που χρειάζεται για λειτουργήσει και 2 απλά κουμπία. Το Esp32 κατασκευάζεται από την εταιρεία espressif, ο επεργαστής έχει δύο πυρήνες, οπότε αυτό σημαίνει ότι μπορεί με ευκολία να είναι ενεργές δύο λειτουργίες λογισμικού ταυτόχρονα. Επιπλέον έχει πολλά I/O (input output) ή εισόδους – εξόδους και πολλά από αυτά είναι γενικής χρήσης, δηλαδή μπορούν να χρησιμοποιηθούν ακόμη και ως διεπαφή για την επικοινωνία με περιφερειακές συσκευές.



Εικόνα 4.1– To ESP-32

Όπως φαίνεται στην παραπάνω φωτογραφία, το ESP-WROOM-32 ουσιαστικά είναι ένα ολοκληρωμένο που αποτελείται από τον μικροελεγχτή, από τα ελάχιστα στοιχεία που χρειάζεται για να δουλέψει, δηλαδή αντιστάσεις, πυκνωτές και από την κεραία. Έτσι μας δίνει και η δυνατότητα να χρησιμοποιηθούν και ασύρματα πρωτόκολλα, δηλαδή WiFi και Bluetooth. Επιπλέον, σημαντικό είναι και το ολοκληρωμένο που μετατρέπει την διεπαφή UART σε USB. Με τον τρόπο αυτό μπορούμε να συνδέσουμε την πλατφόρμα στον υπολογιστή με ευκολία για να γίνει η ανάπτυξη λογισμικού και δεν χρειαζόμαστε κάποια εξωτερική συσκευή για τον προγραμματισμό της.

Χαρακτηριστικά του ESP32 :

- Timers και Watchdog
- Ρολόι πραγματικού χρόνου
- ADC και ενσωματωμένοι αισθητήρες
- Μετατροπέας ψηφιακού σε αναλογικό (DAC)
- Αισθητήρας αφής
- Συνεπεξεργαστής Ultra Low Power (ULP).
- Διεπαφή MAC Ethernet.
- Ελεγκτής κεντρικού υπολογιστή SD/SDIO/MMC.
- Ασύγχρονος δέκτης και πομπός(UART).
- Διεπαφή I2C.
- Διεπαφή I2S.
- Διεπαφή SPI.
- Διαμόρφωση πλάτους παλμού (PWM).
- Επιταχυντής υλικού.

4.2.2 STM32

Το STM32 Nucleo Board είναι μία διάσημη πλατφόρμα ανάπτυξης συσκευής. Είναι κι αυτό σαν το ESP32 και αποτελείται από το κύκλωμα του προγραμματιστή που μας δίνει διεπαφή usb για την εύκολη επικοινωνία με τον υπολογιστή, από ένα μικροελεγχτή μοντέλου STM32, 2 κουμπιά. Επίσης αποτελείται από το ελάχιστο κύκλωμα για την σωστή του λειτουργία, δηλαδή αντιστάσεις, πυκνωτές, κρύσταλο για την σωστή συχνότητα και τέλος ολοκληρωμένο μετατροπής τάσεως από τα 5V του usb

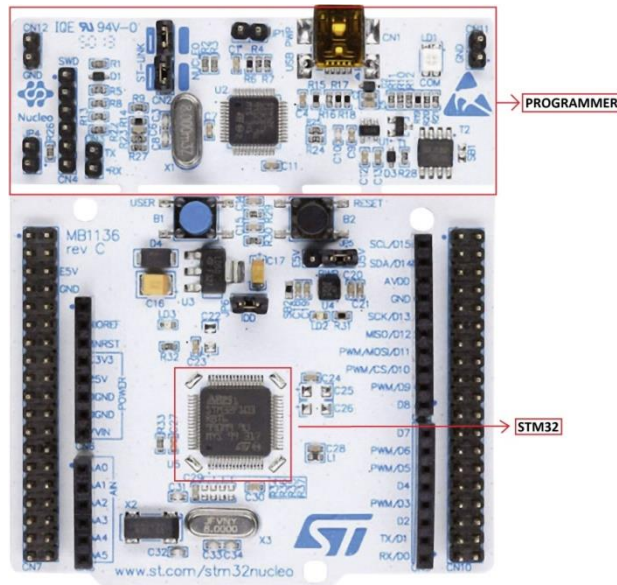
σε 3.3V. Σε αντίθεση με το ESP32 δεν έχει την δυνατότητα να χρησιμοποιήσει από μόνο του WiFi ή Bluetooth, έτσι χρειάζεται κάποια εξωτερική συσκευή για να το πετύχει αυτό. Η παρακάτω φωτογραφία δείχνει το πως μοιάζει η πλακέτα αυτή. Στο πάνω μέρος βλέπουμε το κύκλωμα που βοηθάει στον προγραμματισμό του μικροελεγκτή και στο κάτω μέρος το κύκλωμα του ίδιου του μικροελεγκτή. Έχει αρκετές εισόδους (5 αναλογικές και υπόλοιπες ψηφιακές) και εξόδους, καθώς μας παρέχει και αρκετές διεπαφές για να κάνουμε χρήση πρωτοκόλλων επικοινωνίας. Δυστυχώς το μέγεθος της πλακέτας δεν είναι και πολύ μικρό κι αν χρειαστεί να κάποιο πρωτόκολλο ασύμαρτης επικοινωνίας η απαραίτητη χρήση εξωτερικής συσκευής θα έπαιρνε ακόμη περισσότερο χώρο. Παρόλα αυτά αποτελεί μία καλή επιλογή λόγω της μεγάλης υποστήριξης από την κοινότητα που ασχολείται με αυτό τον τύπο μικροελεγκτή, αλλά και η εταιρεία που κατασκευάζει αυτή την πλατφόρμα, η STMicroelectronics, παρέχει εξαιρετικά έγγραφα και οδηγούς για το πώς πρέπει ο δημιουργός να προσεγγίσει τον προγραμματισμό της συσκευής. Τέλος, η STMicroelectronics παρέχει και δωρεάν λογισμικό για Windows και iOS για τον προγραμματισμό. Το λογισμικό αυτό ονομάζεται STM32CUBE MX και ουσιαστικά επιτρέπει τον δημιουργό με κάποιες ρυθμίσεις να δημιουργήσει αυτόματα κώδικα αρχικοποίησης των περιφερειακών του μικροελεγκτή.

Χαρακτηριστικά του STM32 :

- Επεξεργαστής ο cortex-m3 οποίος είναι 32 bit.
- 32-512 KB μνήμη flash.
- 2-3, 6V τάση τροφοδοσίας.
- Σειριακός εντοπισμός σφαλμάτων (SWD) και διεπαφή JTAG.

Πλεονεκτήματα του STM32 :

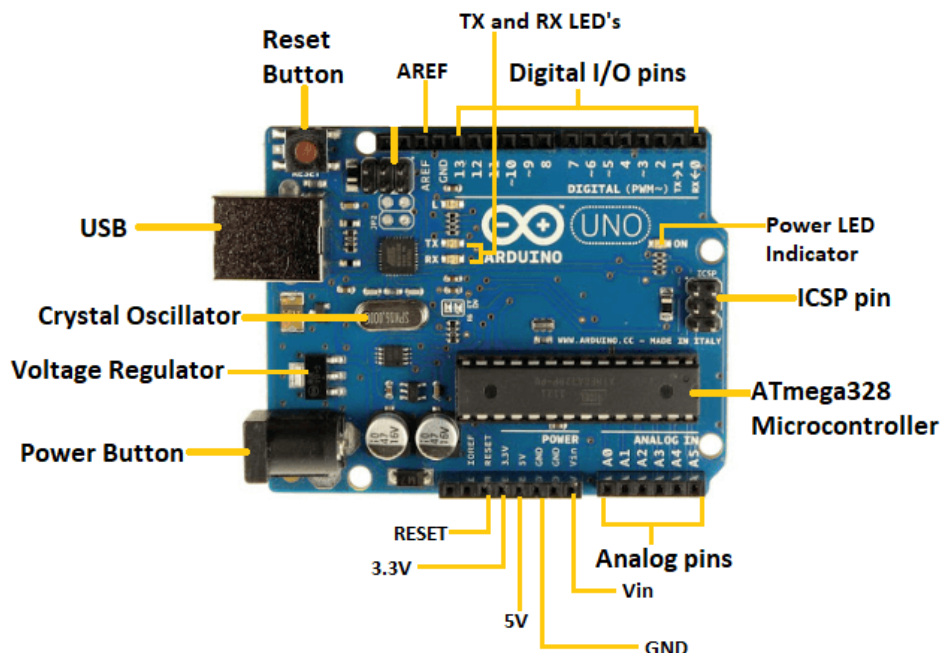
- Χαμηλή τάση.
- Εξοικονόμηση ενέργειας.
- Η τάση τροφοδοσίας ταιριάζει με τις μπαταρίες του εμπορίου όπως αυτές του λιθίου.
- Χαμηλή κατανάλωση ρεύματος υπό συνθήκες.
- Η εναλλαγή από κατάσταση sleep σε εκτέλεση εργασιών είναι πολύ γρήγορη.
- Επεξεργασία αναλογικών και ψηφιακών σημάτων με ευκολία



Εικόνα 4.2 - Το STM32

4.2.3 ARDUINO

Το Arduino είναι, ενδεχομένως, η πιο διάσημη πλατφόρμα ανάπτυξης λογισμικού και πρωτότυπο συσκευής που κυκλοφορεί. Ο λόγος είναι ότι η κοινότητα που ασχολείται με αυτή την πλατφόρμα είναι με διαφορά η μεγαλύτερη από τις άλλες των παραπάνω. Αυτό έχει ως αποτέλεσμα να έχουν αναπτυχθεί σχεδόν αμέτρητες βιβλιοθήκες προς αξιοποίηση.



Εικόνα 4.3 - Το Arduino

Όπως φαίνεται στην φωτογραφία το κύκλωμα της πλατφόρμας δεν έχει πολύ μεγάλη διαφορά σε σύγκριση με τα άλλα. Δηλαδή, παρατηρούμε εισόδους-εξόδους, θύρα USB για την επικοινωνία με τον υπολογιστή, τον μικροελεγκτή ATMEGA328p, μερικά στοιχεία δηλαδή αντιστάσεις, πυκνωτές,

ολοκληρωμένο μετατροπής τάσεως, τον κρύσταλλο και ολοκληρωμένο για μετατροπή UART σε USB. Τα τεχνικά χαρακτηριστικά όμως του Arduino έχουν πολύ μεγάλη διαφορά με την κύρια, όπως ότι δεν είναι 32bit αλλά 8bit. Αυτό σημαίνει ότι οι επιδόσεις του είναι πολύ πιο χαμηλές. Επίσης, σαν το STM32 δεν παρέχει από μόνο περιφεριακά για ασύρματα πρωτόκολλα. Το πλεονέκτημα του είναι όμως ότι μπορεί να επιτευχθεί πολύ χαμηλή κατανάλωση ενέργειας και εννοείται, όπως αναφέρθηκε η πληθώρα συλλογή βιβλιοθηκών.

Χαρακτηριστικά του Arduino Uno R3:

- Χρησιμοποιεί το τσιπάκι ATmega328.
- Οι έξοδοι του είναι στα 5V.
- Μπορεί να δεχτεί τάση τροφοδοσίας 6-20V.
- Περιέχει 14 ψηφιακές εξόδους.
- Περιέχει 6 αναλογικές εισόδους.
- Μέγιστο ρεύμα τα 40 mA.
- Η Flash μνήμη είναι 32 Kb.
- Η SRAM του είναι 2 Kb.
- Η EEPROM του είναι 1 Kb.
- Το ρολόι του δουλεύει στα 16 MHz.

Πλεονεκτήματα του Arduino Uno R3 :

- Δεν χρειάζεται η προσθήκη κάποιου άλλου υλικού για να λειτουργήσει.
- Εύκολο προς την εκμάθηση.
- Προσφέρει εργαλεία που βοηθούν τον προγραμματιστή.
- Έχει αρκετούς έτοιμους κώδικες.
- Έπαρξη μεγάλης κοινότητας για την παροχή βοήθειας.
- Παρέχει φθηνές λύσεις σε πράγματα της καθημερινότητας.

4.2.4 Καταληκτική επιλογή πλατφόρμας για την ανάπτυξη της συσκευής

4.2.4.1 Λόγοι επιλογής ESP32

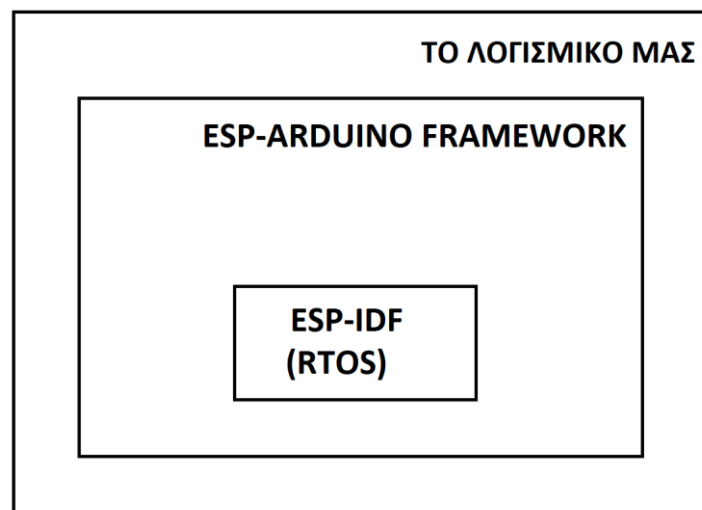
Τελικά αποφασίστηκε να χρησιμοποιηθεί η πλατφόρμα του ESP32 αντί του STM32 Nucleo board ή του ARDUNO UNO R3. Αυτή η απόφαση πάρθηκε για τους παρακάτω λόγους:

- Το ESP32 είναι με διαφορά ο πιο γρήγορος επεξεργαστής. Το Arduino απορρίφθηκε πρώτο διότι είναι 8bit και ο πιο αργός μικροελεγκτής
- Είναι η πιο μικρή πλακέτα
- Μας δίνει την δυνατότητα να χρησιμοποιήσουμε WiFi χωρίς να προσθέσουμε εξώτερικη συσκευή. Επομένως, κερδίζουμε και χώρο αλλά και το κόστος της συσκευής μειώνεται
- Τα Input/Output γενικής χρήσης μας επιτρέπουν να προσθέσουμε αισθητήρες με πολύ μεγάλη ευκολία
- Η χρήση RTOS είναι πολύ εύκολη

Η χρήση ενός συστήματος RTOS είναι επιβεβλημένη, καθώς σκοπός της παρούσας πτυχιακής ήταν να αναπτυχθεί μία εφαρμογή που θα αποστέλλονται και θα λαμβάνονται δεδομένα σε πραγματικό

χρόνο και επιπλέον θέλουμε οι λειτουργίες του λογισμικού να μην διακόπτουν η μία την άλλη, αλλά να είναι ενεργές ταυτόχρονα.

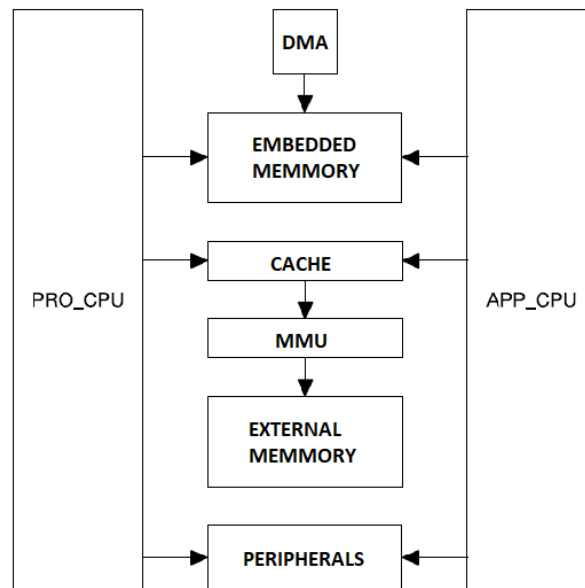
Η Epressif, δηλαδή η εταιρεία που ανέπτυξε το ESP32 παρέχει ένα πλαίσιο προγραμματισμού που αποτελείται από βιβλιοθήκες που χρησιμοποιούν το FreeRTOS. Αυτό σημαίνει ότι η χρήση του συγκεκριμένου RTOS είναι πιο ευνόητη, καθώς η ταχύτητα ανάπτυξης του λογισμικού για την συσκευή θα είναι και πολύ μεγαλύτερη. Αυτό το πλαίσιο προγραμματισμού ονομάζεται ESP-IDF. Όμως ακόμη και η χρήση του ESP-IDF δεν είναι και πολύ εύκολη και θέλει αρκετό χρόνο για να νιώσει κάποιος οικεία με την χρήση του. Όμως υπάρχει ένα λογισμικό που στο τέλος χρησιμοποιήσαμε που είναι γραμμένο πάνω το ESP-IDF. Αυτό λογισμικό ονομάζεται ESP-ARDUINO FRAMEWORK και ουσιαστικά κάνει την σύνταξη και την δομή του κώδικα του ESP να μοιάζει με αυτή του ARDUINO. Το μεγαλύτερο θετικό είναι ότι καθιστά το ESP32 συμβατό με πολλές βιβλιοθήκες που έχουν γραφτεί για το ARDUINO. Επίσης η πηγή του κώδικα είναι ανοιχτή, οπότε μπορεί να την δει και να συμβάλει στην περαιτέρω ανάπτυξη του. Οπότε συνοψίζοντας, η αρχιτεκτονική του λογισμικού μας μέχρι στιγμής φαίνεται στο σχήμα 4.3.1.



Σχήμα 4.4 – Αρχιτεκτονική λογισμικού στο ESP32

4.2.4.2 Τεχνικά χαρακτηριστικά ESP32

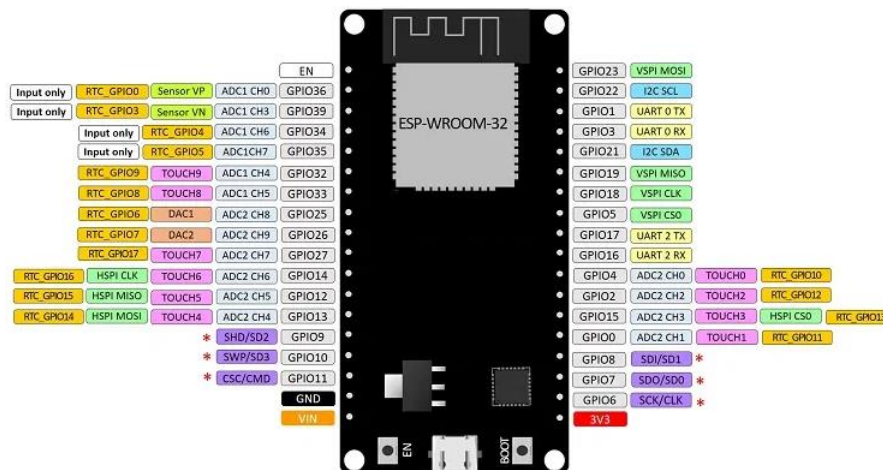
Το ESP32 είναι ένας μικροελεγκτής ο οποίος, ανάλογα την έκδοση, περιέχει μικροεπεξεργαστή της Xtensa ο οποίος μπορεί να είναι είτε μονοπύρηνος είτε διπύρηνος (32-bit). Το ESP32 δουλεύει σε συχνότητες από 160 Mhz έως 240 MHz. Οι πυρήνες είναι ο Protocol CPU (PRO_CPU) και ο Application CPU (APP_CPU). Ο **PRO_CPU** είναι υπεύθυνος για το WiFi, το Bluetooth, το SPI, το I2C, το ADC κ.λπ. Το ESP-IDF χρησιμοποιεί το freeRTOS για εναλλαγή μεταξύ των επεξεργαστών και ανταλλαγή δεδομένων μεταξύ τους. Το μοντέλο που χρησιμοποιήθηκε έχει δύο πυρήνες.



Σχήμα 4.5 - Αρχιτεκτονική ESP32 [30]

Οι επεξεργαστές του ESP32 έχουν εσωτερικές μνήμες αλλά και εξωτερικές και η κάθε μια απο αυτές υπάρχει για διαφορετικό σκοπό.

- Το DMA σημαίνει Direct Access Memory και έχει σχεδιαστεί για να μεταφέρει δεδομένα όσο πιο γρήγορα γίνεται. Επιτρέπει στην συσκευή να μεταφέρει δεδομένα κατευθείαν προς ή από την μνήμη, χωρίς να παρεμβαίνει η CPU δηλαδή, ο επεξεργαστής.
- Embedded memory είναι η εσωτερική μνήμη.
- Η cache και η MMU αφορούν την πρόσβαση και την χρήση της εξωτερικής μνήμης του ESP32.
- Τα περιφεριακά είναι οι είσοδοι-έξοδοι, το Wifi κτλπ...



εφαρμογή στο κινητό του ή να χρειάζεται να την χειρίζεται με διάφορα κουμπιά γιατί θα μπορούσε να γίνει περίπλοκη η χρήση της. Συνήθως μία έτοιμη πλακέτα εμπορίου με μια οθόνη περιέχει ένα ολοκληρωμένο του οποίου η δουλειά είναι να ελέγχει την οθόνη, τι θα δείξει και με ποιον τρόπο, και επίπλεον αν είναι αφής θα περιέχει και έναν ελεγχτή που θα διαβάξει σε ποιο σημείο την έχει ακουμπήσει ο χρήστης. Οι οθόνες αφής χωρίζονται σε δύο κατηγορίες που ονομάζονται :

- Capacitive
- Resistive

Οι capacitive οθόνες αφής αποτελούνται από πολλούς μικρούς πυκνωτές στην σειρά, έτσι όταν το δάχτυλο ακουμπήσει μια περιοχή αυτοί οι πυκνωτές φορτίζουν σε εκείνο σημείο και έτσι μπορούμε να καταλάβουμε σε ποιο σημείο έχει ακουμπήσει ο χρήστης

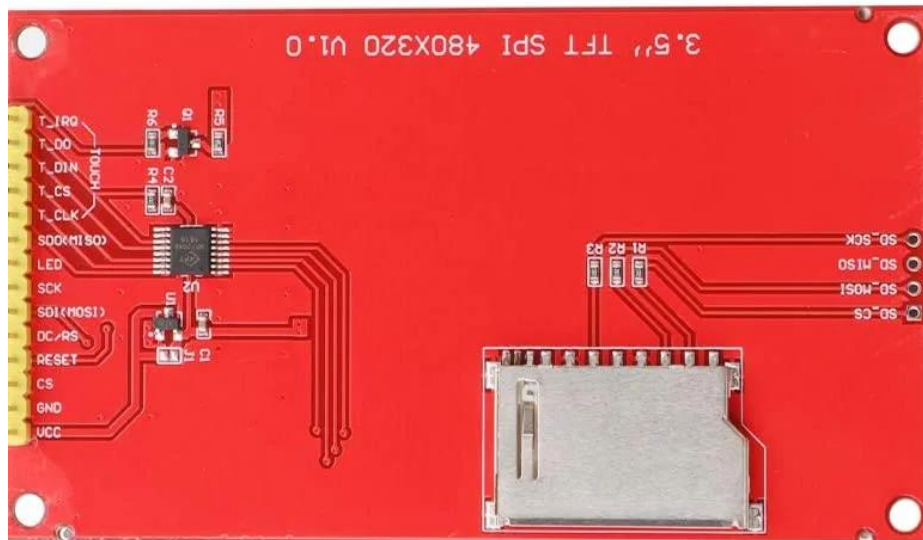
Οι Resistive οθόνες αφής αποτελούνται από δύο φύλλα που είναι φτιαγμένα από υλικά με ηλεκτρική αντίσταση. Αυτά τα φύλλα είναι χωρισμένα μεταξύ τους και όταν ο χρήστης με την πίεση του δάχτυλου του ακουμπάει την οθόνη αυτά έρχονται σε επαφή μεταξύ τους σε εκείνο το σημείο. Επομένως, έχουμε την δυνατότητα να καταλάβουμε που ήταν αυτό.

Τα Controller των οθονών που υπάρχουν στην πλακέτα παρέχουν κάποια διεπαφή για να επικοινωνίσουν με τον μικροελεγχτή όπως, SPI ή I2C ή ακόμη και καποιές απλές επαφές εξόδων-εισόδων. Το ίδιο ακριβώς ισχύει και για τους ελεγκτές της αφής. Από την έρευνα που κάναμε διαπιστώσαμε ότι κάποιοι από αυτούς του ελεγκτές είχαν πολύ μεγαλύτερη υποστήριξη σε σχέση με άλλους . Ως εκ τούτου, υπήρχαν διαθέσιμες βιβλιοθήκες με ανοιχτή πηγή κώδικα στις οποίες είχαν συμβάλει με κώδικα πάρα πολλά άτομα με σκοπό την κατασκευή και την βελτίωση τους. Εξαιτίας όσων αναφέρθηκαν θεωρούνται και αρκετά αξιόπιστες. Τα driver οθονών της εταιρίας ILITEK φάνηκαν να είναι καλά υποστηριγμένα από την κοινότητα, κάποια από αυτά είναι το ILI9341, το ILI9486 κ.α.

Η πλακέτα με την οθόνη που καταλήξαμε να χρησιμοποιήσουμε περιέχει οθόνη ανάλυσης 480x320 pixels, είναι 3.5 inch και έχει πάνω της το controller ILI9486 για την έλεγχο της και το XPT2046 touch controller για τον έλεγχο της αφής. Η πλακέτα και οι επαφές της φαίνονται στις παρακάτω φωτογραφίες (φωτογραφία 4.4.1 και 4.4.2).



Εικόνα 4.7– Η οθόνη από την μπροστινή πλευρά



Εικόνα 4.8– Η οθόνη από την πίσω πλευρά

Όπως φαίνεται η πλακέτα μας παρέχει διεπαφές για την χρήση SPI και για το ILI9486 δηλαδή για τον οδηγό της οθόνης, αλλά και για το XPT2046, τον οδηγό την αφής. Επίσης μας δίνει την δυνατότητα να χρησιμοποιήσουμε και κάρτα SD, αλλά δεν την χρειαζόμασταν για την εφαρμογή μας, όποτε οι επαφές στην δεξιά πλευρά παρέμειναν άδειες. Αυτή την στιγμή στο σημείο που βρισκόμαστε πρέπει να βρούμε κάποιο εργαλείο, έτσι ώστε να μας βοηθήσει να σχεδιάσουμε μία βολική εφαρμογή για τον χρήστη που να απεικονίζεται στην οθόνη.

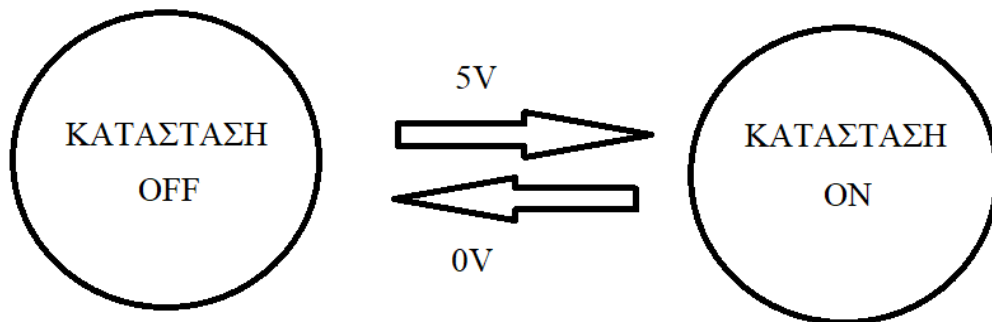
4.4 Αρχιτεκτονική λογισμικού της συσκευής

4.4.1 Finite State Machine

Εφόσον, πλέον έχουμε βρει και επιλέξει τα απαραίτητα εργαλεία για την συσκευή πρέπει να γίνει συνολικός σχεδιασμός του λογισμικού. Η δημιουργία πλάνου για την σωστή σχεδίαση του λογισμικού είναι σημαντικό, διότι κατά την διάρκεια της ανάπτυξης την εφαρμογής, ο σχεδιαστής μπορεί να πλοηγηθεί στα αρχεία με τον κώδικα πιο εύκολα, ως αποτέλεσμα η παραγωγικότητα να παραμένει υψηλή.

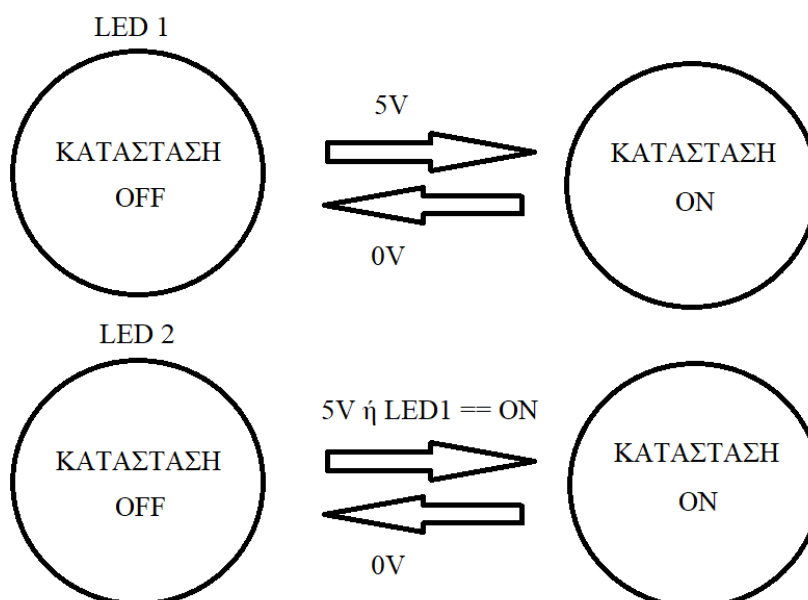
Αρχικά, η αρχιτεκτονική λογισμικού που επιλέξαμε να ακολουθήσουμε για την εφαρμογή την συσκευής ονομάζεται Finite State Machine. Είναι ένα διάσημο μοτίβο προγραμματισμού που χρησιμοποιείται κυρίως σε συσκευές embedded σαν κι αυτή της παρούσας πτυχιακής εργασίας. Μία απλή περιγραφή για αυτό το μοτίβο είναι η κατάσταση (state), περιγράφει τα χαρακτηριστικά της λειτουργίας την συσκευής εκείνη την ώρα. Για παράδειγμα, εάν ένα Led είναι κλειστό βρίσκεται σε κατάσταση OFF δηλαδή σε state OFF, εάν είναι ανοιχτό σε state ON, σε ανοιχτή κατάσταση. Η μετάβαση ανάμεσα σε αυτές τις καταστάσεις αποτελείται από μία συλλογή πράξεων και εκτελείται, μόνο, εφόσον ακολουθούνται συγκεκριμένα κριτήρια [18]. Πιο συγκεκριμένα, εάν ορίσουμε ότι ένας μικροελεγχτής την ώρα που στείλει 5V στο led, αυτό θα ανάψει, όταν και μόνο όταν, ο μικροελεγχτής στείλει αυτά τα 5V, τότε θα γίνει η μετάβαση από την κατάσταση OFF στη κατάσταση ON. Έτσι έχουμε μια προβλέψιμη μετάβαση ανάμεσα στις καταστάσεις και δεν μπορούμε να βρεθούμε σε κατάσταση την οποία δεν έχουμε ορίσει. Με αυτόν τον τρόπο είναι εύκολο να εντοπίσουμε κάποιο προγραμματιστικό λάθος στην συσκευή διότι, μπορούμε να δούμε το ιστορικό μετάβασης τον

καταστάσεων και να εντοπίσουμε ποια κριτηρία έχουνε τηρηθεί. Στο παρακάτω σχήμα (4.4.1) μπορούμε να δούμε τις καταστάσεις για το παράδειγμα που δόθηκε.



Σχήμα 4.9 - Κατάσταση Finite State Machine

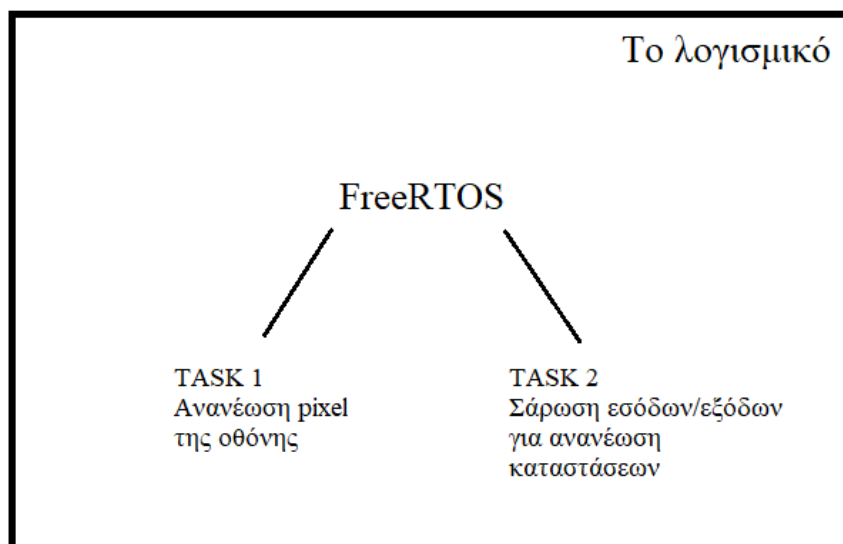
Οι κύκλοι απεικονίζουν τις καταστάσεις και τα βελάκια τις συνθήκες για να πραγματοποιηθεί η μετάβαση. Αν για παράδειγμα η είσοδος είναι λιγότερο ή περισσότερο από 5V η κατάσταση θα παραμείνει OFF και αντίστοιχα όταν βρισκόμαστε στην κατάσταση ON, αν η τάση δεν γίνει μηδενική, η κατάσταση δεν θα ξαναγυρίσει στην OFF. Το παραπάνω αποτελεί ένα πολύ απλό παράδειγμα και εύκολα θα μπορούσαμε να είχαμε προσθέσει περισσότερες καταστάσεις ή πιο περίπλοκες συνθήκες. Όπως θα χρησιμοποιηθεί στο λογισμικό, μπορούμε να ορίσουμε και μία συλλογή από καταστάσεις, δηλαδή σαν να έχουμε παραπάνω από ένα LED. Κάθε κατάσταση μπορεί να είναι ανεξάρτητη από την άλλη ή όμως μπορεί να είναι και αλληλοεξαρτώμενες. Για παράδειγμα, αν είχαμε 2 LED συνδεδεμένα σε ένα μικροελεγχτή, αν είχαμε είσοδο 5V δηλαδή ψηφιακό «1» για το πρώτο και άναβε, τότε θα μπορούσαμε να ορίσουμε να ανάψει και το δεύτερο. Έτσι βλέπουμε ότι, οι συνθήκες και οι μεταβάσεις μπορούν να γίνουν με εύκολο τρόπο αρκετά περίπλοκες. Το θετικό από την χρήση του μοτίβου παραμένει και μπορούμε να εντοπίσουμε και να δούμε τις καταστάσεις και τις μεταβάσεις που έχουν τα περιφεριακά του μικροελεγχτή μας.



Σχήμα 4.10 - Καταστάσεις Finite State Machine

Σε αυτό εδώ το μοτίβο μπορεί εύκολα να μπει και η χρήση του RTOS, διότι είναι δυνατό να ελέγχει ταυτόχρονα πολλές εισόδους-εξόδους καθώς, και να πραγματοποιεί τις μεταβάσεις ανάλογα με τα κριτήρια.

Έχοντας πολλά περιφεριακά και πολλές λειτουργίες που θέλουμε να φτιάξουμε για την συσκευή, είναι λογικό να έχουμε πίνακες από Finite State Machine, όπου κάποιοι από αυτούς να έχουν σχέση μεταξύ τους και κάποιοι όχι. Στην συσκευή έχουμε περιφερειακές λειτουργίες, όπως την σύνδεση στο WiFi, την σάρωση των δικτύων, την χρήση των αισθητήρων, την απεικόνιση της εφαρμογής στην οθόνη κα. Σε περίπτωση που αλλάξει κάποια κατάσταση, όπως για παράδειγμα ο χρήστης να πατήσει ένα κουμπί στην οθόνη, η ανανέωση των pixel της είναι σημαντική. Η άμεση ανανέωση της οθόνης κάνει την συσκευή φαίνεται ότι αποκρίνεται πολύ γρήγορα. Για τον λόγο αυτό δημιουργήσαμε ένα ξεχωριστό έργο (task) στο επίπεδο του RTOS. Το άλλο task που δημιουργήσαμε είναι ουσιαστικά όλη η κατάσταση της συσκευής που περιέχει όλες τις υποκαταστάσεις κάθε περιφερειακής λειτουργίας της συσκευής.



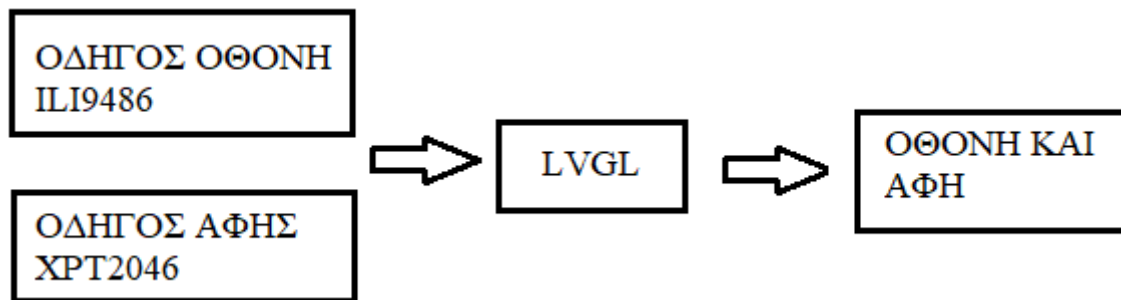
Σχήμα 4.11 Tasks (έργα) του λογισμικού FreeRtos

4.4.2 Ανανέωση των Pixels της οθόνης (Task 1)

Όπως προαναφέρθηκε για την δημιουργία της εφαρμογής στην συσκευή χρειαζόμαστε να βρούμε και να χρησιμοποιήσουμε ένα εργαλείο με ανοιχτή πηγή κώδικα, έτσι ώστε να μας βοηθήσει να γίνει πιο εύκολα και πιο γρήγορα. Το εργαλείο που μας κέντρισε το ενδιαφέρον είναι μία βιβλιοθήκη που ονομάζεται LVGL. Η LVGL μας βοηθάει να «χτίσουμε» την εφαρμογή μας με αρκετά γρήγορο ρυθμό και μας παρέχει λειτουργίες, ακόμη και για χρήση φωτογραφιών στο User Interface (UI). Ειδικότερα ο τρόπος που λειτουργεί αυτή η βιβλιοθήκη είναι [18]:

- Βοηθάει να σχεδιάσεις την εφαρμογή εικονικά
- Συμβάλλει στο να αντιστοιχείς εικονικά κουμπιά στην εφαρμογή με λειτουργίες που έχουν προγραμματιστεί στον μικροελεγχτή
- Παρέχει μέσω λογισμικού διεπαφές για να συνδέσεις το λογισμικό της οθόνης, δηλαδή του controller, στην περίπτωση μας το ILI9486, έτσι ώστε το LVGL να “ξέρει” πως να απικονερίζει πίξελ στην οθόνη

- Παρέχει μέσω λογισμικού διεπαφές για να συνδέσεις το λογισμικό της οθόνης αφής, έτσι ώστε το LVGL να «γνωρίζει» σε ποιο σημείο της εφαρμογής πάτησε ο χρήστης.

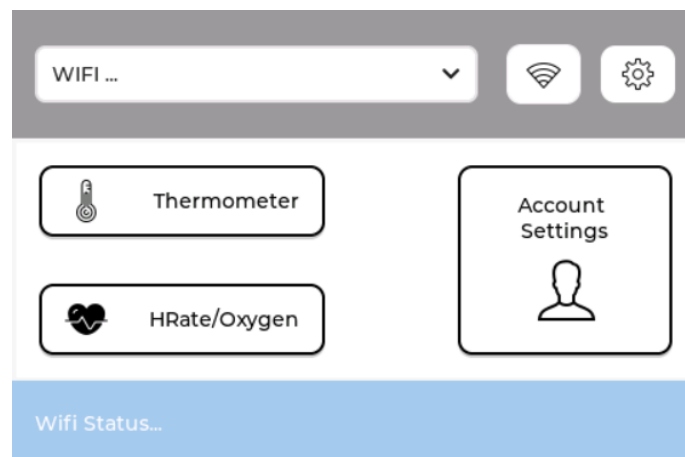


Σχήμα 4.12 - Αρχιτεκτονική λογισμικού οθόνης

Ουσιαστικά η LVGL έχει την δυνατότητα να καλέσει τις συναρτήσεις των οδηγών και έτσι να γίνει σωστή απεικόνιση της εφαρμογής σχεδόν αυτόματα. Επομένως, αποτελεί ένα ενδιάμεσο λογισμικό που χρησιμοποιείται σε μικροελεγχτές για τον έλεγχο των οθονών. Έτσι το πρώτο task αποτελείται από την ανίχνευση της οθόνης αφής και από μία συνάρτηση που μας παρέχει η LVGL η οποία ανανεώνει την οθόνη με βάση το σημείο στο οποίο ακούμπησε ο χρήστης. Αν ο χρήστης για παράδειγμα ακουμπήσει ένα κουμπί το οποίο ανοίγει την λίστα με συσκευές wifi, μόλις γίνει ανίχνευση, κατευθείαν, η LVGL θα δείξει στην οθόνη ένα μικρό παραθυράκι που θα περιέχει τις συσκευές αυτές.

4.4.3 Η εφαρμογή της συσκευής

Η αρχική απεικόνιση της εφαρμογής που βλέπει ο χρήστης, μόλις ανοίξει στην συσκευή, φαίνεται στην παρακάτω φωτογραφία και δίνει διάφορες επιλογές. Κάθε φορά που γίνεται επανακκίνηση της συσκευής, η εφαρμογή ξεκινάει σε αυτό το σημείο που είναι το λεγόμενο «Home» δηλαδή, η αρχική οθόνη. Από αυτό το σημείο ο χρήστης μπορεί να πλοηγηθεί σε άλλα παράθυρα από τα οποία μπορεί να ξαναγυρίσει πίσω στο «Home».



Εικόνα 4.13 - Αρχική οθόνη

Οι επιλογές που βρίσκονται μέσα στο γκρι πλαίσιο είναι οι εξής:

- Το πρώτο άσπρο πλαίσιο, αν το ακουμπήσει ο χρήστης θα εμφανιστεί μία λίστα με τις συσκευές που μπορεί να συνδεθεί το ESP32 και υπάρχει για την σύνδεση στο διαδίκτυο. Αν ο χρήστης επιλέξει κάποιο από αυτά τα δίκτυα, τότε εμφανίζεται μικρότερο παράθυρο για την εισαγωγή του κωδικού

πρόσβασης της συσκευής στην οποία προσπαθεί να συνδεθεί. Επίσης εμφανίζεται πληκτρολόγιο σαν αυτό στα κινητά τηλέφωνα για άνετη πληκτρολόγηση.

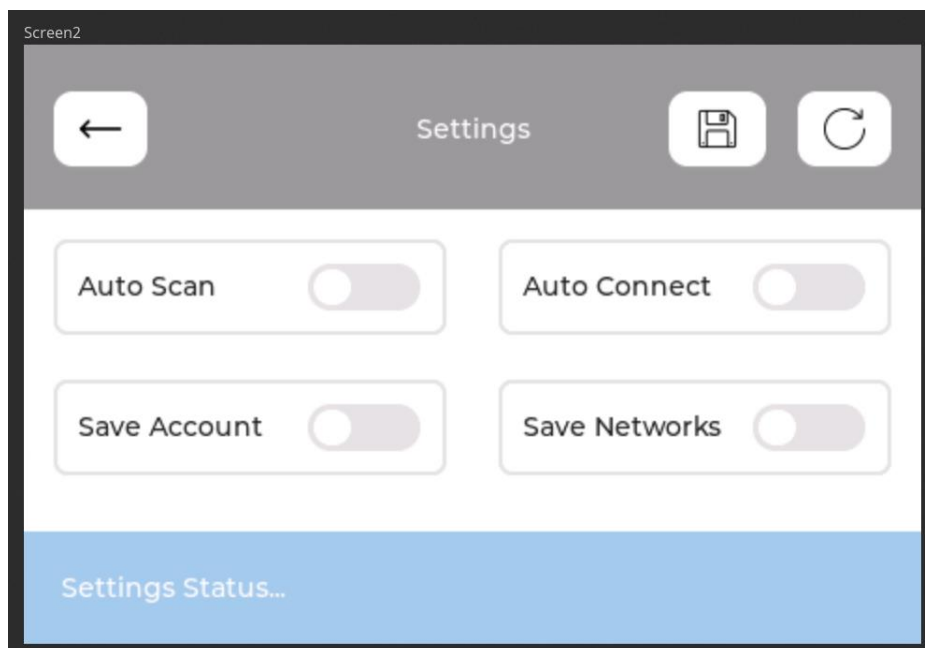
- Το δεύτερο στοιχείο είναι ένα κουμπί που αν πατηθεί παρατεταμένα θα ξαναγίνει σάρωση των συσκευών.
- Το τρίτο στοιχείο οδηγεί σε νέο παράθυρο, το οποίο αφορά διάφορες ρυθμίσεις για την συσκευή.

Οι επιλογές που βρίσκονται στο μεσαίο (άσπρο) πλαίσιο είναι οι εξής:

- Το πρώτο στοιχείο είναι ένα κουμπί που οδηγεί σε νέο παράθυρο το οποίο αφορά την χρήση του θερμομέτρου και την αποστολή των τιμών στην διαδικτυακή εφαρμογή.
- Το δεύτερο στοιχείο είναι ένα κουμπί που οδηγεί σε νέο παράθυρο που αφορά την χρήση του οξύμετρου και την μέτρηση των καρδιακών παλμών και μετέπειτα την αποστολή των τιμών στην διαδικτυακή εφαρμογή
- Το τρίτο στοιχείο το μεγαλύτερο στα δεξιά του άσπρου πλαισίου είναι ένα κουμπί που οδηγεί σε νέο παράθυρο και αφορά την σύνδεση του χρήστη στον λογαριασμό της διαδικτυακής εφαρμογής.

Τέλος, το τελευταίο γαλάζιο πλαίσιο μας δείχνει την κατάσταση που βρίσκεται το WiFi του μικροελεγκτή. Ουσιαστικά μας δείχνει την κατάσταση του λογισμικού. Για παράδειγμα αν ο χρήστης πληκτρολογήσει λάθος κωδικό για την συσκευή WiFi, τότε κατά την διάρκεια της προσπάθειας σύνδεσης σε εκείνο το σημείο θα γράφει «connecting...». Όταν αποτύχει η σύνδεση θα γράψει «Connection Failed». Με τον τρόπο αυτό ο χρήστης έχει πληροφορίες για τι προσπαθεί να κάνει η συσκευή.

Η απεικόνιση του παραθύρου των ρυθμίσεων της συσκευής φαίνεται στην παρακάτω φωτογραφία.



Εικόνα 4.14 - Οθόνη Ρυθμίσεων

Οι επιλογές που βρίσκονται μέσα στο γκρι πλαίσιο είναι οι εξής:

- Το πρώτο στοιχείο είναι κουμπί το οποίο οδηγεί τον χρήστη στην αρχική οθόνη.

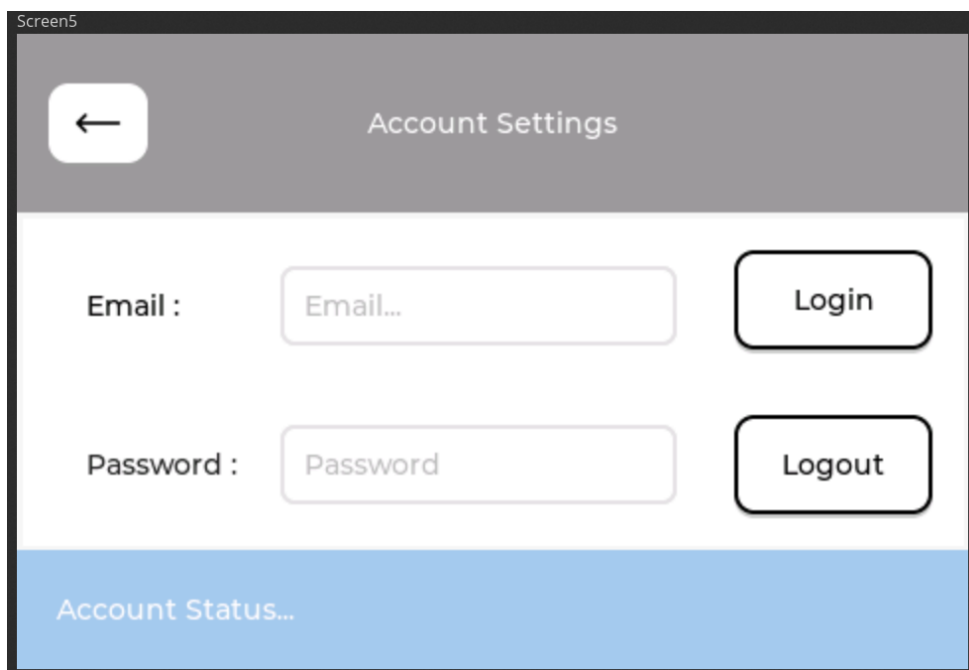
- Το δεύτερο στοιχείο που το εικονίδιο του απεικονίζει μία δισκέτα είναι για την αποθήκευση των παρακάτω ρυθμίσεων σε περίπτωση που τις αλλάξει ο χρήστης, έτσι ώστε αν κλείσει η συσκευή να μην χαθούν οι αλλαγές.
- Το τρίτο στοιχείο είναι η επαναφορά των ρυθμίσεων στην κατάσταση που είναι αποθηκευμένες. Για παράδειγμα αν ο χρήστης αλλάξει μια ρύθμιση και δεν έχει αποθηκεύσει τις αλλαγές, όταν πατήσει αυτό το κουμπί, θα γυρίσει πίσω στην προηγούμενη της κατάσταση.

Οι επιλογές που βρίσκονται στο μεσαίο (άσπρο) πλαίσιο είναι οι εξής:

- Auto Scan: Αν αυτή η ρύθμιση είναι ενεργή, όταν ανοίξει την συσκευή ο χρήστης, η συσκευή θα κάνει αυτόματα σάρωση συσκευών για την σύνδεση στο WiFi
- Auto Connect : Αν αυτή η ρύθμιση είναι ενεργή τότε μετά την σάρωση των συσκευών δικτύου, αν βρεθεί η τελευταία συσκευή που ήταν συνδεδεμένος ο μικροελεγχτής, θα προσπαθήσει να συνδεθεί σε αυτή αυτόματα.
- Save Account: Αν αυτή η ρύθμιση είναι ενεργή τότε θα γίνει αυτόματη αποθήκευση του λογαριασμού του χρήστη της διαδικτυακής εφαρμογής και θα συνδεθεί σε αυτόν αυτόματα αν υπάρχει σύνδεση διαδικτύου.
- Save Networks : Αν αυτή η σύνδεση είναι ενεργή τότε θα αποθηκεύονται αυτόματα τα δίκτυα που συνδέεται η συσκευή με επιτυχία. Αν απενεργοποιηθεί αυτά τα στοιχεία θα διαγραφούν.

Τέλος, το τελευταίο γαλάζιο πλαίσιο μας δείχνει την κατάσταση που βρίσκονται οι ρυθμίσεις του μικροελεγχτή. Δηλαδή, αν για κάποιο λόγο οι ρυθμίσεις δεν μπορούν να αποκτήσουν τις αποθηκευμένες τιμές θα εμφανίσει «Setting's Loading Failed». Αντίθετα, αν όλα λειτουργήσουν σωστά τότε θα εμφανίσει «Settings Loaded». Άλλο παράδειγμα είναι η αποθήκευση των νέων τιμών των ρυθμίσεων, αν γίνει η απόθεση με επιτυχία θα δείξει «Saved!».

Η απεικόνιση του παραθύρου των λειτουργιών του λογαριασμού της χρήστη φαίνεται στην παρακάτω φωτογραφία.



Εικόνα 4.15 - Οθόνη σύνδεσης στον λογαριασμό

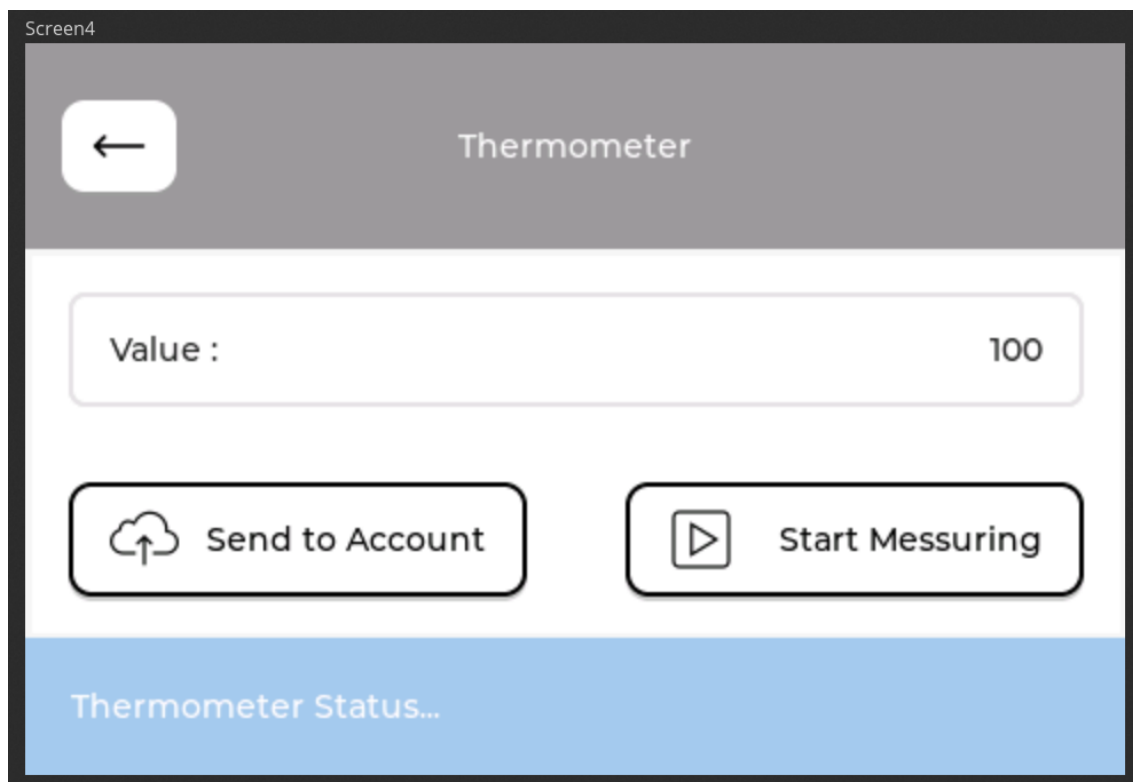
Η μόνη επιλογή που βρίσκεται μέσα στο γκρι πλαίσιο είναι το κουμπί που οδηγεί τον χρήστη πίσω στην αρχική οθόνη.

Τα στοιχεία που βρίσκονται μέσα στο μεσαίο (άσπρο) πλαίσιο είναι τα εξής:

- Το πρώτο στοιχείο είναι ένα πλαίσιο στο οποίο μπορεί ο χρήστης να πληκτρολογήσει το email του λογαριασμού του στην διαδικτυακή εφαρμογή.
- Το δεύτερο στοιχείο είναι ένα πλαίσιο στο οποίο μπορεί ο χρήστης να πληκτρολογήσει τον κωδικό του λογαριασμού του στην διαδικτυακή εφαρμογή.
- Το τρίτο στοιχείο είναι ένα κουμπί «Login» κατά το οποίο ο χρήστης συνδέεται στον λογαριασμό του αφότου πρώτα πληκτρολογήσει τα προσωπικά στοιχεία του λογαριασμού του.
- Το τέταρτο στοιχείο είναι ένα κουμπί το οποίο αποσυνδέει τον χρήστη από την διαδικτυακή εφαρμογή και λειτουργεί μόνο εφόσον είναι ήδη συνδεδεμένος.

Τέλος το τελευταίο γαλάζιο πλαίσιο μας δείχνει την κατάσταση που βρίσκεται ο λογαριασμός του χρήστη, δηλαδή αν ο χρήστης έχει συνδεθεί στην διαδικτυακή εφαρμογή. Για παράδειγμα αν είναι συνδεδεμένος θα γράφει «Connected» ή αν προσπαθήσει να συνδεθεί και η σύνδεση αποτύχει και πληκτρολογήσει λάθος κωδικό ή e-mail, θα εμφανίσει «Wrong Credentials».

Η απεικόνιση του παραθύρου για την χρήση του θερμομέτρου φαίνεται στην παρακάτω φωτογραφία.



Εικόνα 4.16 - Οθόνη Μέτρησης θερμοκρασίας

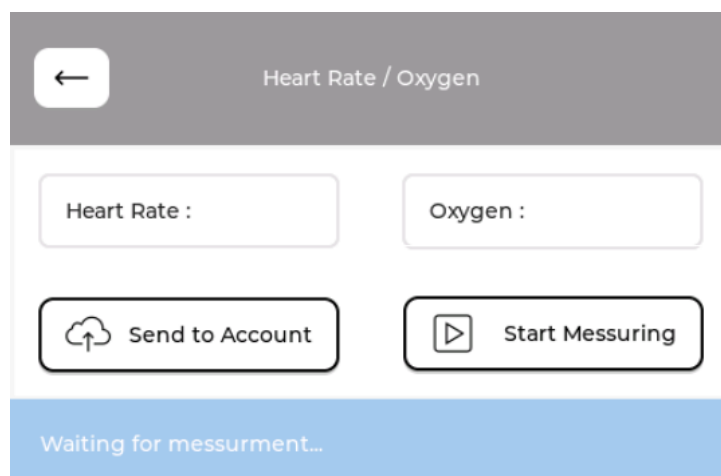
Η μόνη επιλογή που βρίσκεται μέσα στο γκρι πλαίσιο, όπως και στο παράθυρο για τις λειτουργίες του λογαριασμού, είναι το κουμπί που οδηγεί τον χρήστη πίσω στην αρχική οθόνη.

Τα στοιχεία που βρίσκονται μέσα στο μεσαίο (άσπρο) πλαίσιο είναι τα εξής:

- Το άσπρο πλαίσιο με το γκρι περίγραμμα είναι η τιμή της μέτρησης που παίρνουμε από το θερμόμετρο.
- Το πρώτο κουμπί, ακριβώς κάτω από το άσπρο πλαίσιο, το οποίο γράφει «Send To Account», στέλνει την μέτρηση στον λογαριασμό του χρήστη, μόνο εάν είναι συνδεδεμένος στον λογαριασμό και έχει προηγηθεί επιτυχημένη μέτρηση
- Το δεύτερο κουμπί ακριβώς δίπλα ξεκινάει την μέτρηση της θερμοκρασίας με την χρήση του θερμόμετρου που είναι συνδεδεμένο στην συσκευή.

Τέλος το τελευταίο γαλάζιο πλαίσιο μας δείχνει την κατάσταση που βρίσκεται το θερμόμετρο. Πιο συγκεκριμένα, τη μέτρηση της θερμοκρασίας. Για παράδειγμα αν δεν γίνει σωστή λήψη των δειγμάτων της θερμοκρασίας θα γράψει «Thermometer Messurment Failed» ή αντίθετα αν τα δείγματα ληφθούν σωστά θα εμφανίσει «Thermometer Messurment Was Successful». Ακόμη ένα παράδειγμα αποτελεί η αποστολή της τιμής της θερμοκρασίας στον λογαριασμό του χρήστη. Αν η αποστολή γίνει με επιτυχία θα γράψει «Sent to account successfully», αν όχι «Failed to upload the messurment».

Η απεικόνιση του παραθύρου για την χρήση του οξύμετρου και την μέτρηση των καρδιακών παλμών φαίνεται στην παρακάτω φωτογραφία.



Εικόνα 4.17 - Οθόνη Μέτρησης ποσοστό οξυγόνου / καρδιακών παλμών

Η μόνη επιλογή που βρίσκεται μέσα στο γκρι πλαίσιο ,όπως και στο παράθυρο για τις λειτουργίες του λογαριασμού, είναι το κουμπί που οδηγεί τον χρήστη πίσω στην αρχική οθόνη.

Τα στοιχεία που βρίσκονται μεσα στο μεσαίο (άσπρο) πλαίσιο είναι τα εξής:

- Τα άσπρα πλαίσια με τα γκρι περιγράμματα είναι οι τιμές των μετρώσεων που παίρνουμε από το αισθητήρα.
- Το πρώτο κουμπί, ακριβώς κάτω από το άσπρο πλαίσιο, το οποίο γράφει «Send To Account», στέλνει την μέτρηση στον λογαριασμό του χρήστη, μόνο εάν είναι συνδεδεμένος στον λογαριασμό και έχει προηγηθεί επιτυχημένη μέτρηση
- Το δεύτερο κουμπί ακριβώς δίπλα ξεκινάει την μέτρηση των τιμών

Τέλος το τελευταίο γαλάζιο πλαίσιο μας δείχνει την κατάσταση που βρίσκεται ο αισθητήρας όπως γίνεται και με το θερμόμετρο.

Τα χρώματα , τα σχήματα , οι επιλογές αλλά και η δομή τους μοιάζουν μεταξύ τους σε όλα τα παράθυρα. Αυτό έγινε για να νιώθει ο χρήστης ότι χρησιμοποιεί μία ενιαία εφαρμογή , και να μην μπερδεύεται στην χρήση της εφαρμογής , αφού για παράδειγμα σε κάθε παράθυρο θα περιμένει στην πάνω αριστερή θέση το κουμπί με το βελάκι για να γυρίσει πίσω στην αρχική οθόνη και θα κοιτάει στο τρίτο γαλάζιο πλαίσιο για να δει την κατάσταση της λειτουργίας στην οποία βρίσκεται.

4.4.4 Πίνακας καταστάσεων και διαχείριση τους (Task 2)

Το δεύτερο task επιβλέπει και τρέχει το RTOS είναι και το πιο μεγάλο.Οι τρεις πιο κρίσιμες λειτουργίες που επιβλέπει είναι :

1. Η κατάσταση του σαρωτή των δικτύων.
2. Την σύνδεση στο δίκτυο.
3. Την σύνδεση του χρήστη στην διαδικτυακή εφαρμογή.

Οι καταστάσεις που έχουμε ορίσει για τα παραπάνω είναι :

Για το σαρωτή δικτύων:

1. IDLE ,δηλαδή δεν κάνει τίποτα
2. SCANNING , δηλαδή η κατάσταση κατά την οποία γίνεται η σάρωση
3. SCAN_DONE , δηλαδή όταν έχει ολοκληρωθεί η σάρωση

Για την σύνδεση στο δίκτυο:

1. CONNECTING, δηλαδή προσπάθεια σύνδεσης
2. RECONNECTING, δηλαδή προσπάθεια σύνδεσης , μετά από κάποια διακοπή
3. CONNECTED , δηλαδή λειτουργική σύνδεση
4. CONNECTION FAILED , αποτυχής σύνδεση
5. CONNECTION LOST , διακοπή σύνδεσης
6. DISCONNECTED , όταν δεν υπάρχει σύνδεση

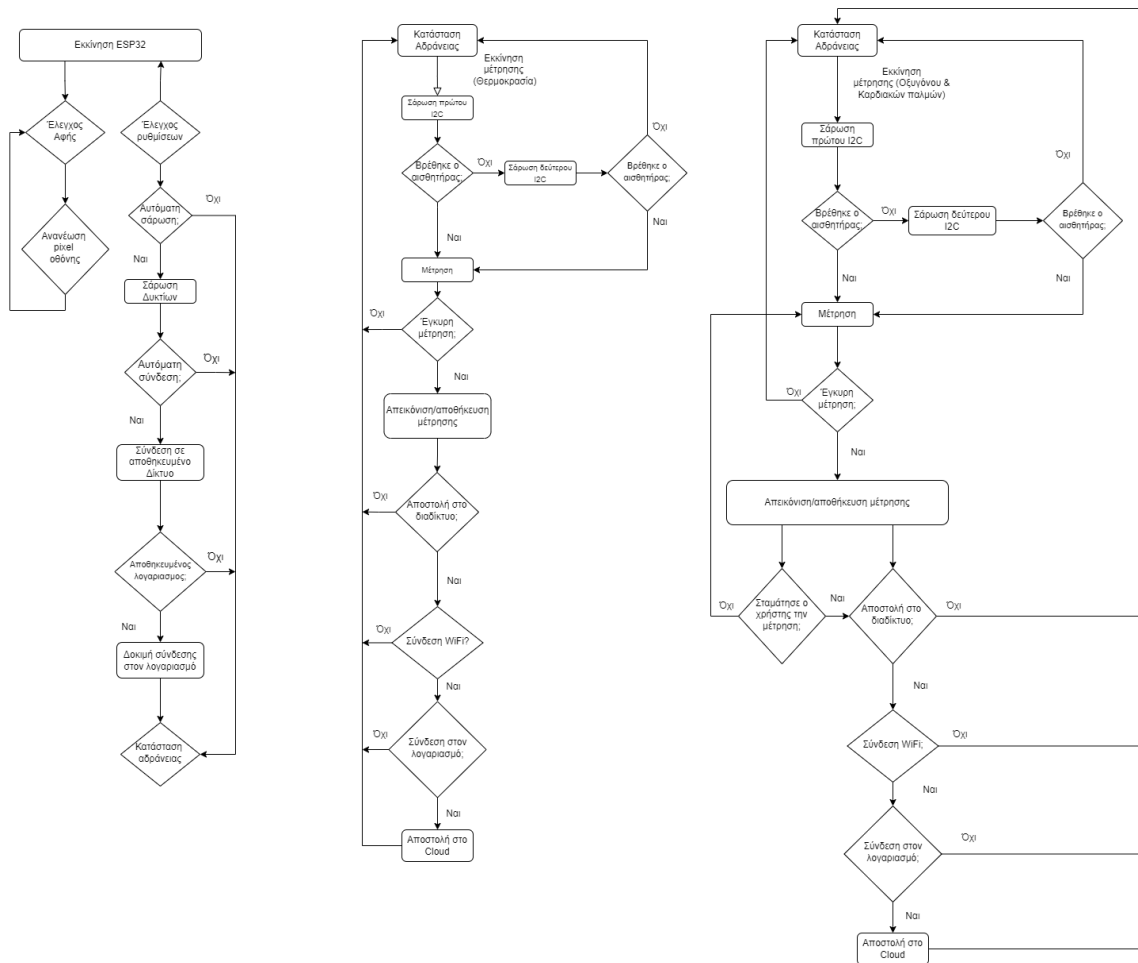
Και τέλος για την σύνδεση του χρήστη στην διαδικτυακή εφαρμογή:

1. ACC_NOT_CONN , όταν ο χρήστης δεν είναι συνδεδεμένος
2. ACC_CONN , όταν ο χρήστης είναι συνδεδεμένος
3. ACC_ERR , όταν υπάρχει κάποιο πρόβλημα με την σύνδεση

Στο λογισμικό που προγραμματίσαμε η διαχείριση και η ενημέρωση των παραπάνω καταστάσεων γίνεται σε μία λούπα που ονομάζεται «manageState». Παρόλα αυτά, οι καταστάσεις είναι ανεξάρτητες μεταξύ τους .Για να ενημερώσουμε την παραπάνω λούπα ότι έχει γίνει μια αλλαγή σε κάποια κατάσταση ,όπως για παράδειγμα όταν ο χρήστης πατάει στην οθόνη το κουμπί για να γίνει σάρωση των δικτύων, υψώνουμε μία σημαία. Στην συγκεκριμένη περίπτωση την έχουμε ονομάσει «scannerStatusFlag». Η δουλειά της λούπας είναι να δει αν κάποια τέτοια σημαία έχει κατάσταση «1».Σε περίπτωση που έχει ανάλογα με το καινούριο State της λειτουργίας, θα «τρέξει» κάποιες συναρτήσεις. Πιο συγκεκριμένα στο παραπάνω παράδειγμα θα «τρέξει» την συνάρτηση «scanNetworks» και θα γίνει σάρωση των συσκευών δικτύου. Έτσι, όπως αναφέρθηκε παραπάνω, κατά την διάρκεια της ανάπτυξης του λογισμικού, μπορούμε εύκολα να δούμε πότε οι σημαίες αλλάζουν καταστάσεις και επίσης να δούμε την προηγούμενη κατάσταση των λειτουργιών, την τωρινή και ποια μπορεί να πάρει στο μέλλον.

4.4.5 Διάγραμμα ροής λογισμικού

Τα βασικά διαγράμματα ροής φαίνονται στις παρακάτω εικόνες.



Εικόνα 4.18 – Διάγραμμα ροής

Το αριστερό διάγραμμα ροής αφορά την ακολουθία των λειτουργιών που εκτελεί ο μικροελεγκτής, μόλις του δοθεί τροφοδοσία. Πιο συγκεκριμένα, ακριβώς μετά την εκκίνηση του, ξεκινάει ο έλεγχος την οθόνης και η λειτουργία της αφής, δηλαδή το Task 1, όπως αναφέρθηκε παραπάνω. Η συγκεκριμένη λειτουργία εκτελείται συνεχώς, έτσι ώστε η συσκευή να αποκρίνεται με επιτυχία σε σύντομο χρονικό διάστημα στις εντολές του χρήστη. Με την ανανέωση των pixels της οθόνης ξεκινάει και ο έλεγχος των ρυθμίσεων της συσκευής. Οι ρυθμίσεις αυτές φαίνονται στην εικόνα 4.14 - Οθόνη Ρυθμίσεων. Πριν η συσκευή καταλήξει σε κατάσταση αδράνειας (IDLE STATE), εάν κάποιες από αυτές είναι ενεργές, θα εκτελέσει κάποιες συναρτήσεις, όπως αυτόματα σύνδεση σε δίκτυο.

Το κεντρικό διάγραμμα ροής αποτελεί τη λογική που εκτελεί ο μικροελεγκτής σε περίπτωση που ο χρήστης μετρήσει την θερμοκρασία με την χρήση του αισθητήρα. Το λογισμικό έχει δημιουργηθεί έτσι ώστε να μην έχει σημασία σε ποια θύρα I2C συνδέσει ο χρήστης τον αισθητήρα, διότι γίνεται σάρωση για να βρεθεί που είναι συνδεδεμένος. Έαν βρεθεί, τότε ξεκινάει η μέτρηση και εφόσον αυτή είναι έγκυρη αποθηκεύεται προσωρινά στην συσκευή και είναι έτοιμη να σταλθεί στον διαδικτυακό

λογαριασμό του χρήστη με την χρήση WiFi. Γενικότερα κάθε φορά που εκτελείται μία αλληλουχία συναρτήσεων, στο τέλος ο μικροελεγκτής καταλήγει σε κατάσταση αδράνειας. Εάν υπάρξει κάποιο σφάλμα ή κάποια ενημέρωση, όπως αναφέρθηκε παραπάνω, τότε αυτά απεικονίζονται στο μπλε πλαίσιο της οθόνης. Επειδή η μέτρηση της θερμοκρασίας γίνεται πολύ γρήγορα, δεν χρησιμοποιείται έργο (task) στο επίπεδο του RTOS. Έτσι, ο μικροελεγκτής δεν μπορεί να εκτελέσει άλλη λειτουργία ταυτόχρονα, αλλά πρακτικά αυτό γίνεται για λιγότερο από ένα δευτερόλεπτο και δεν το νιώθει ο χρήστης.

Το δεξί διάγραμμα ροής αποτελεί τη λογική που εκτελεί ο μικροελεγκτής όταν ο χρήστης χρησιμοποιεί τον αισθητήρα οξυγόνου/καρδιακών παλμών. Όπως και με τον αισθητήρα θερμοκρασίας έτσι και με του οξυγόνου/καρδιακών παλμών δεν έχει σημασία σε πιο βύσμα θα συνδεθεί. Η διαφορά τους όμως είναι ότι για την μέτρηση των παλμών και του οξυγόνου απαιτούμε πολλά δείγματα μετρήσεων, τα οποία προϋποθέτουν σημαντικό χρονικό διάστημα για να συλλεχθούν (4-30 δευτερόλεπτα). Έτσι δημιουργήσαμε έργο (task) στο επίπεδο του FreeRtos, οπότε ο μικροελεγκτής δεν μπλοκάρει την εκτέλεση άλλων λειτουργιών κατά την διάρκεια των μετρήσεων. Ο μικροελεγκτής συνεχίζει να κάνει μετρήσεις μέχρι ο χρήστης να τις σταματήσει με το κουμπί «Stop Messurment». Η μέτρηση αυτή θα συμπεριληφθεί στα δείγματα, μόνο εφόσον είναι έγκυρη. Επομένως, ο χρήστης έχει την δυνατότητα να σηκώσει το δάχτυλο του από τον αισθητήρα και να κάνει κάποια άλλη ενέργεια, όπως να σταματήσει την μέτρηση, να πάει στις ρυθμίσεις ή ακόμη και να συνδεθεί στο διαδίκτυο, χωρίς να αλλοιωθεί σημαντικά η τελική δειγματοληψία των παραπάνω βιομετρικών δεδομένων.

4.5 Φυσικά στοιχεία συσκευής και σύνδεση μεταξύ τους

4.5.1 Πρωτόκολλο SPI και οι επαφές του

Πλέον υπάρχει ένα ολοκληρωμένο λογισμικό που αφορά την επικοινωνία των περιφερειακών της συσκευής και της εφαρμογής. Εφόσον πλέον έχουμε τελειώσει με τον πρωτότυπο, το επόμενο βήμα είναι να διαλέξουμε ποιες επαφές της πλατφόρμας του μικροελεγκτή θα χρησιμοποιήσουμε για να σχεδιάσουμε την ηλεκτρονική πλακέτα PCB.

Ξέρουμε ότι η πλακέτα της οθόνης με την αφή παρέχει διεπαφές για την σύνδεση της στον μικροελεγκτή με την χρήση SPI. Παρέχονται pins για την χρήση SPI και για το ολοκληρωμένο που αφορά τον έλεγχο της οθόνης και για το ολοκληρωμένο που αφορά τον έλεγχο την αφής. Άρα, δύο διαφορετικές διεπαφές.

Το SPI είναι μία διεπαφή, η οποία χρησιμοποιείται για την μεταφορά πληροφοριών ανάμεσα σε μικροελεγκτές και αισθητήρες ή για την επικοινωνία ανάμεσα σε μικροελεγκτές. Το SPI λειτουργεί με σύγχρονο τρόπο, δηλαδή η αποστολή των πληροφοριών γίνεται όταν υπάρχει αλλαγή του παλμού ενός ρολογιού. Στο SPI υπάρχει μόνο μία συσκευή που έχει το ρολόι και αυτή ονομάζεται ελεγκτής. Είναι μοναδικός και υπάρχουν και τα περιφερειακά που μπορεί να είναι παραπάνω από ένα.

Όταν οι πληροφορίες πάνε από τον ελεγκτή προς περιφερειακή συσκευή, μεταφέρονται με μια γραμμή πληροφοριών που ονομάζεται MOSI στην οποία ο ελεγκτής είναι η έξοδος και η περιφερειακή συσκευή είναι η είσοδος. Στην περίπτωση που θέλει η περιφερειακή συσκευή να στείλει πληροφορίες πίσω στον ελεγκτή, στέλνει τις πληροφορίες μέσω της γραμμής MISO, στην οποία ο ελεγκτής είναι η είσοδος και η περιφερειακή συσκευή είναι η έξοδος. Το SPI παρέχει την δυνατότητα να στέλνονται και να λαμβάνονται πληροφορίες την ίδια ακριβώς στιγμή. Εκτός από αυτές τις γραμμές το SPI έχει ακόμα μία γραμμή που λέγεται «Chip Select» και ουσιαστικά είναι αυτό που ενεργοποιεί ή απενεργοποιεί την περιφερειακή συσκευή και επιπλέον χρησιμοποιείται σαν

διευθυνσιοδότης όταν υπάρχουν παραπάνω απο μία περιφερειακές συσκευές. Όταν το Chip Select βρίσκεται σε υψηλό δυναμικό, τότε η περιφερειακή συσκευή είναι απενεργοποιημένη, ενώ όταν πηγαίνει σε χαμηλό δυναμικό ενεργοποιείται [20].

Επομένως, συνδέσαμε στην ίδια γραμμή MOSI και MISO τον ελεγχτή την αφής και τον ελεγχτή την οθόνης. Αλλά η γραμμή CHIP SELECT που στην ουσία ενεργοποιεί και την επικοινωνία μεταξύ του μικροελεγκτή και του ολοκληρωμένου έπρεπε να είναι διαφορετική. Αρα είχαμε δύο γραμμές CHIP SELECT. Εκτός από τις επαφές για την χρήση SPI, η πλακέτα της οθόνης χρειάζεται 5V για την σωστή λειτουργία του φωτισμού της όπως επίσης και γείωση.

4.5.2 Πρωτόκολλο I2C και οι επαφές του

Για την επικοινωνία μεταξύ των αισθητήρων και του μικροελεγκτή αποφασίσαμε να χρησιμοποιήσουμε το πρωτόκολλο I2C, καθώς υπήρχαν επαφές για την χρήση του. Ο τρόπος λειτουργίας του I2C είναι ο εξής: Το I2C είναι ένα πρωτόκολλο σειριακής επικοινωνίας. Σειριακή επικοινωνία ορίζεται η επικοινωνία στην οποία η μεταφορά της πληροφορίας γίνεται το κάθε bit ξεχωριστά μόνο μέσω ενός καλωδίου, το οποίο επιτρέπει την επικοινωνία μεταξύ πολλών συσκευών που κάποιες απο αυτές χρησιμοποιούνται σαν master και κάποιες άλλες σαν slave. Το προνόμιο της συγκεκριμένης τεχνολογίας είναι ότι μπορούν πολλοί slave να συνδεθούν με έναν μοναδικό master αλλά και πολλοί master μπορούν να συνδεθούν σε όσους slave θέλουν.

Για την επικοινωνία μεταξύ master και slave χρειάζονται μόνο δύο pin. Το ένα είναι το SDA, το οποίο είναι για την σειριακή μεταφορά των πληροφοριών απο τον master στον slave και αντίστροφα. Το δεύτερο pin είναι το SCL, το οποίο είναι υπεύθυνο για τον συγχρονισμό του σήματος. Η λειτουργία του I2C είναι σύγχρονη που σημαίνει ότι οι πληροφορίες που στέλνονται με τις πληροφορίες που παραλαμβάνονται είναι στον ίδιο χρόνο λόγω του ίδιου ρολογιού που περιέχεται και στον master και στον slave και ρυθμίζεται πάντα από τον master. Η μεταφορά της πληροφορίας γίνεται με την μορφή μηνυμάτων. Το κάθε μήνυμα είναι χωρισμένο σε κάποια frames. Επίσης, το μήνυμα έχει ένα frame με την διεύθυνση του slave, αλλά και κάποια ακόμη για την μεταφορά της πληροφορίας. Το μήνυμα ορίζει δύο συνθήκες ως αρχή και ως τέλος και επιπλέον ενδιάμεσα απο κάθε frame πληροφορίας περιέχει τα read/write bits και ACK/NACK bits.

Τα στοιχεία του I2C είναι τα εξής:

- Κατάσταση Αρχής: Το SDA μεταβαίνει από υψηλό δυναμικό σε χαμηλό δυναμικό πριν το SCL αλλάξει από υψηλό δυναμικό σε χαμηλό δυναμικό.
- Κατάσταση τέλους: Το SDA αλλάζει από χαμηλό δυναμικό σε υψηλό δυναμικό αφού το SCL αλλάξει από χαμηλό δυναμικό σε υψηλό δυναμικό.
- Πλαίσιο διευθύνσεων: Μια σειρά 7 ή 10 bit διαφορετική για κάθε slave για να γνωρίζει ο master με ποιον θέλει να επικοινωνήσει.
- Read/Write Bit: Ένα ξεχωριστό bit που ορίζει εάν ο master αποστέλει (χαμηλό δυναμικό) ή δέχεται πληροφορίες (υψηλό δυναμικό).
- ACK/NACK Bit: Κάθε frame σε ένα μήνυμα ακολουθείται από ένα bit επιβεβαίωσης/μη επιβεβαίωσης. Εάν ένα frame διεύθυνσης ή frame πληροφοριών ελήφθη με επιτυχία, ένα bit ACK επιστρέφεται στον αποστολέα από τον παραλήπτη.

Ο master στέλνει την συνθήκη αρχής σε κάθε slave, αλλάζοντας το SDA από ένα υψηλό δυναμικό σε ένα χαμηλό δυναμικό πριν αλλάξει το SCL από υψηλό δυναμικό σε χαμηλό δυναμικό. Έπειτα ο master στέλνει σε κάθε slave τη διεύθυνση 7 ή 10 bit του slave με τον οποίο θέλει να επικοινωνήσει,

μαζί με το read/write bit. Αφού τελειώσει αυτό κάθε slave συγκρίνει τη διεύθυνση που στάλθηκε από τον master με τη δική του διεύθυνση. Εάν η διεύθυνση ταιριάζει, ο slave επιστρέφει ένα bit ACK φέρνοντας το SDA σε χαμηλό δυναμικό για ένα bit. Εάν η διεύθυνση από το master δεν ταιριάζει με τη διεύθυνση του slave, ο slave φέρνει το SDA σε υψηλό δυναμικό. Έπειτα ο master στέλνει ή παίρνει το frame πληροφορίας. Αφού σταλθεί κάθε frame πληροφορίας, η συσκευή λήψης επιστρέφει ένα άλλο bit ACK στον αποστολέα για να επιβεβαιώσει την επιτυχή λήψη του frame. Τέλος, για να τερματίσει τη μεταφορά πληροφοριών, ο master στέλνει μια συνθήκη διακοπής στον slave, αλλάζοντας το SCL σε υψηλό δυναμικό πριν από την αλλαγή SDA σε υψηλό δυναμικό [20].

Επομένως, χρησιμοποιήσαμε δύο γραμμές I2C, σε περίπτωση που για κάποιο λόγο θέλαμε να συνδέσουμε δύο ίδιους αισθητήρες ή αν τύχαινε οι συσκευές να έχουν ακριβώς την ίδια διεύθυνση. Αν ξέραμε με σιγουρία ότι οι συσκευές έχουν διαφορετική διεύθυνση θα χρησιμοποιούσαμε μόνο μία γραμμή I2C και θα συνδέαμε τους slave, δηλαδή τους αισθητήρες πάνω σε αυτή την γραμμή. Τελικός σκοπός ήταν η πλακέτα PCB να ήταν όσο πιο ευέλικτη γίνεται.

4.5.3 Παροχή τροφοδοσίας

Για την παροχή τροφοδοσίας της πλακέτας PCB χρησιμοποιήσαμε δύο εισόδους. Ο χρήστης μπορεί να συνδέσει ή την πρώτη είσοδο ή την δεύτερη. Η πρώτη είσοδος είναι βύσμα DC POWER JACK και η δεύτερη είσοδος είναι μια απλή κλέμα με δύο επαφές, μία επαφή για την γείωση και μία επαφή για την τροφοδοσία. Η τροφοδοσία πηγαίνει σε ένα μετατροπέα τάσης που λειτουργεί με διακόπτη (switching regulator) που μετατρέπει την τάση εισόδου σε 5V. Έπειτα αυτή η τάση τροφοδοτεί το ESP32 και την πλακέτα της οθόνης για τον σωστό φωτισμό και λειτουργίας της. Η τροφοδοσία του επεξεργαστή του ESP32 πρέπει να είναι 3.3V, όμως όταν δίνεται τάση στο ESP32 από συγκεκριμένη επαφή ή από το USB, περνάει από μετατροπέα τάσης που βρίσκεται πάνω στην πλακέτα ESP32. Αυτός ο μετατροπέας τάσης την χαμηλώνει στα επίπεδα που χρειάζεται, δηλαδή στα 3.3V. Ο λόγος που χρησιμοποιούμε και ξεχωριστό εξωτερικό μετατροπέα τάσης είναι επειδή ο μετατροπέας τάσης πάνω στο ESP32 αντέχει μόνο μέχρι 9V με αποτέλεσμα θερμαίνεται γρήγορα σε αυτά τα επίπεδα και δεν μπορεί να δώσει αρκετό ρεύμα απευθείας σε περιφεριακές συσκευές. Ενώ ο εξωτερικός μετατροπέας τάσης μπορεί να δώσει μέχρι 3A και αντέχει εισόδους μέχρι και 24V. Το μόνο αρνητικό είναι ότι μαζί με το μετατροπέα τάσης, για να λειτουργήσει σωστά, έπρεπε να χρησιμοποιηθούν κάποιοι πυκνωτές και ένα πυνίο. Αυτό αύξησε το κόστος και το μέγεθος της συσκευής. Το κύκλωμα αυτό το σχεδιάσαμε με βάση την σύσταση του Datasheet. Έτσι επιλέχθηκαν και κατάλληλες τιμές γι' αυτά τα στοιχεία.

4.5.4 Λογισμικό σχεδιασμού PCB (KiCad)

Για την υλοποίηση του σχεδιασμού της πλακέτας (PCB) για την συσκευή χρησιμοποιήσαμε το λογισμικό kiCad. Το λογισμικό αυτό έχει ανοιχτή πηγή κώδικα και δίνει άδεια για την δωρεάν χρήση του ακόμη και για σχεδίαση προϊόντων μαζικής παραγωγής. Δεν έχει κανένα περιορισμό στις λειτουργίες που παρέχει, ενώ αντίθετα πολλά άλλα λογισμικά σχεδίασης συνήθως παρέχουν λειτουργίες αλλά είναι περιορισμένης χρήσεως και πρέπει ο σχεδιαστής να αγοράσει κάποια άδεια. Το kiCad παρέχει την δυνατότητα απεικόνισης τρισδιάστατου μοντέλου του σχεδίου. Αυτό βοηθάει ιδιαίτερα έτσι ώστε ο σχεδιαστής να ξέρει πως μοιάζει το τελικό αποτέλεσμα ακόμη και με τα στοιχεία πάνω στην πλακέτα. Οι βασικές βιβλιοθήκες των αποτυπωμάτων και τρισδιάστατων μοντέλων των στοιχείων είναι αρκετά πλούσια, όμως δεν βρήκαμε αυτά που θέλαμε να χρησιμοποιήσουμε. Ευτυχώς όμως οι περισσότεροι κατασκευαστές παρέχουν τα απαραίτητα αρχεία των στοιχείων που παράγουν και πούλανε και έτσι μπορέσαμε με ευκολία να βρούμε το αποτύπωμα

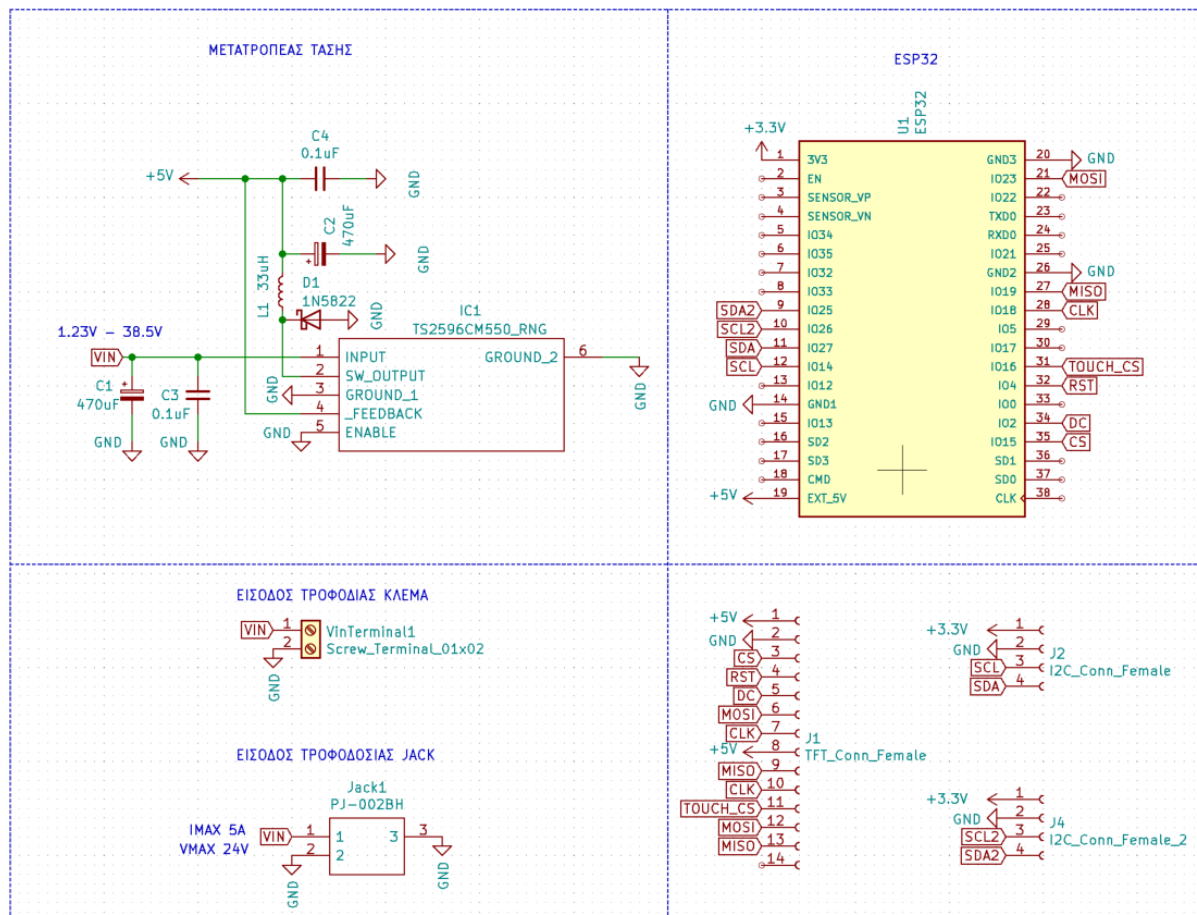
τους και τρισδιάστατο σχεδιασμό τους και να τα κάνουμε εισαγωγή στο kiCad.Επιπλέον, αποφασίσαμε να κάνουμε τον σχεδιασμό με δύο επιφάνειες (2 layer) για διευκόλυνση της σύνδεσης των στοιχείων, επιλογή που μας έδινε το σχεδιαστικό λογισμικό, όπως προαναφέρθηκε, δωρεάν.

4.5.5 Το κύκλωμα και η λίστα των στοιχείων

Το συνολικό κύκλωμα της συσκευής φαίνεται στην παρακάτω φωτογραφία και είναι χωρισμένο σε τέσσερα κομμάτια. Το πρώτο είναι ο μετατροπέας τάσης για την τροφοδοσία του ESP32, το δεύτερο είναι οι συνδέσεις του ESP32 , το τρίτο οι είσοδοι για τροφοδοσία, δηλαδή η κλέμα και το Dc Jack. Τέλος, το τέταρτο είναι οι συνδέσεις για την οθόνη και οι δύο διεπαφές για χρήση του I2C.

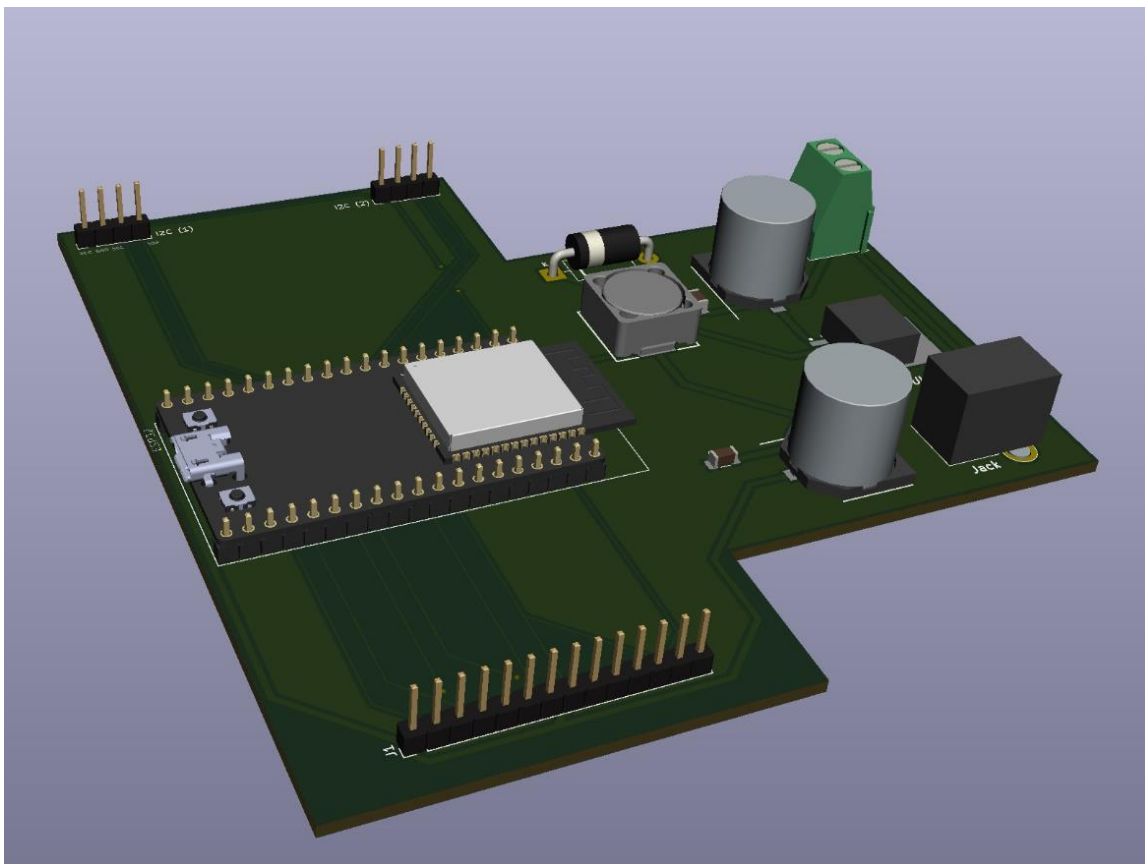
Τα στοιχεία που χρησιμοποιήθηκαν είναι τα εξής:

- 1 δίοδος schottky 1N5822 [1]
- 1 μετατροπέας τάσης (TS2596CM550) Switching Voltage Regulators 3A STEP DOWN 5.0V [2]
- 1 Dc Jack PJ-002BH [3]
- 1 ESP32 [4]
- 2 πυκνωτές με τιμή 470uf1 [5]
- 2 πυκνωτές με τιμή 0.1uF [6]
- 1 πηνίο 33uH [7]
- 1 κλέμα δύο επαφών
- 1 1x14 Connector Female 2.54mm
- 2 1x04 Connector Female 2.54mm

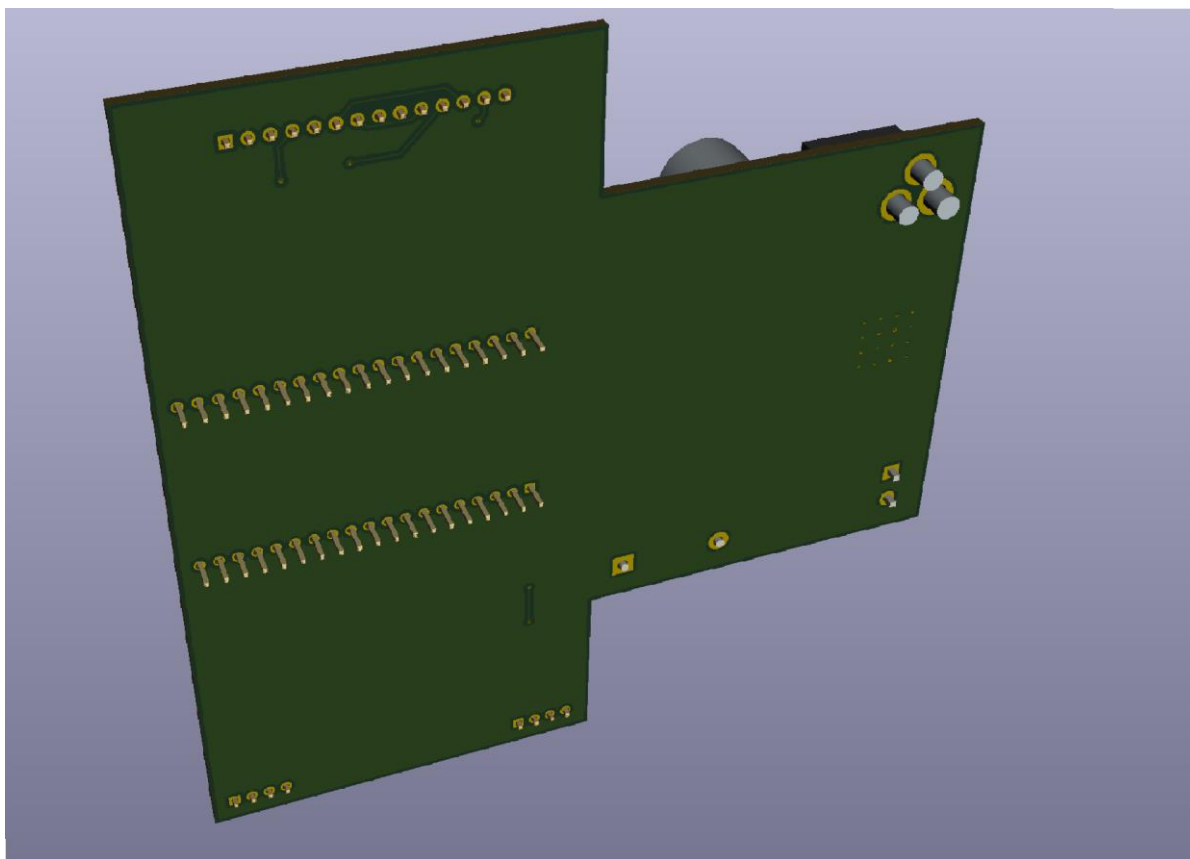


Εικόνα 4.19 - Το κύκλωμα της συσκευής

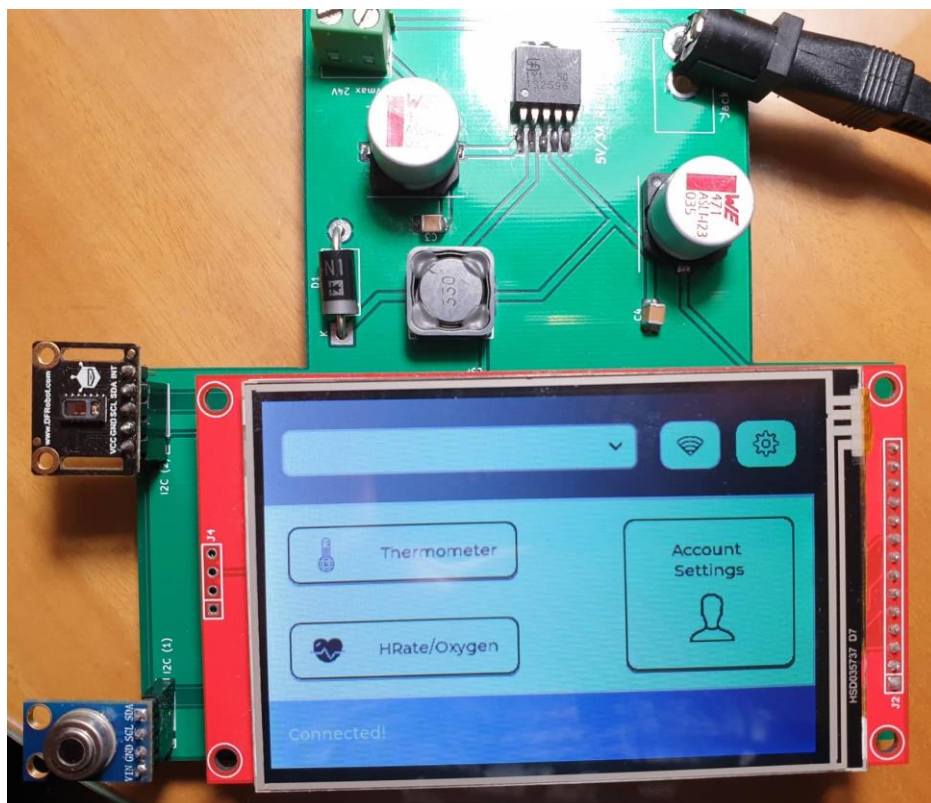
Οι επόμενες φωτογραφίες δείχνουν πως έγινε η σύνδεση των στοιχείων, το σχήμα της πλακέτας, μερικές διαστάσεις και την τρισδιάστατη μορφή της από την πάνω και την κάτω πλευρά:



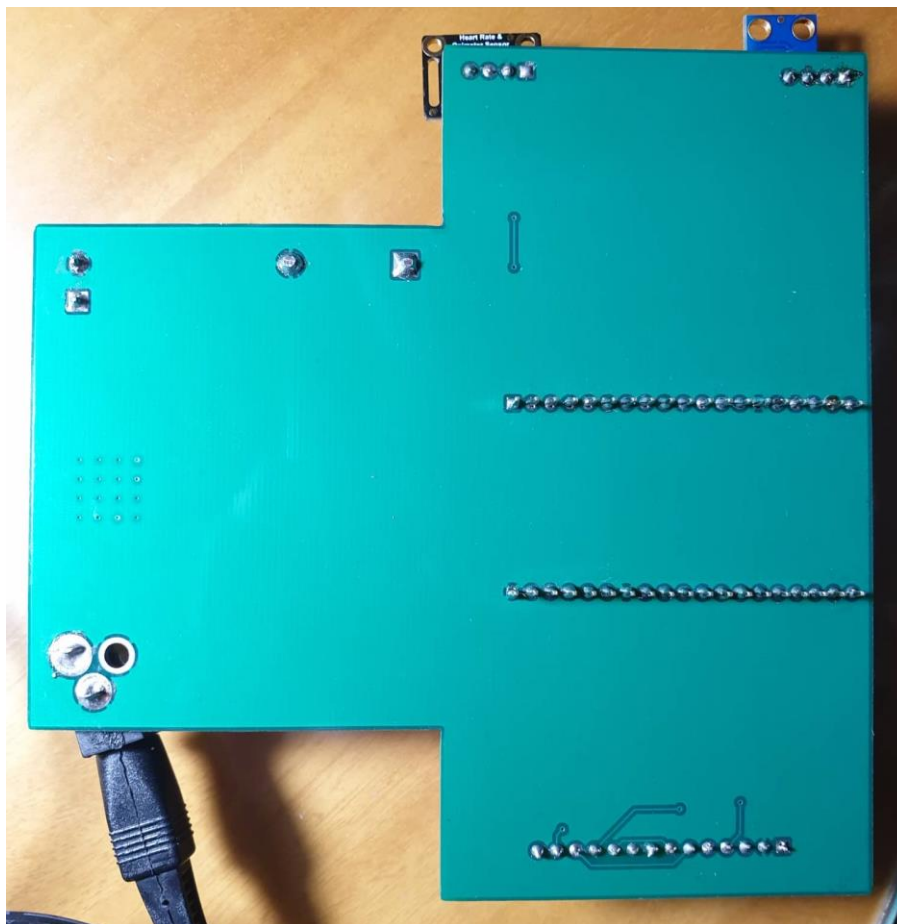
Εικόνα 4.22 - Η πάνω πλευρά της συσκευής σε τρισδιάστατο μοντέλο



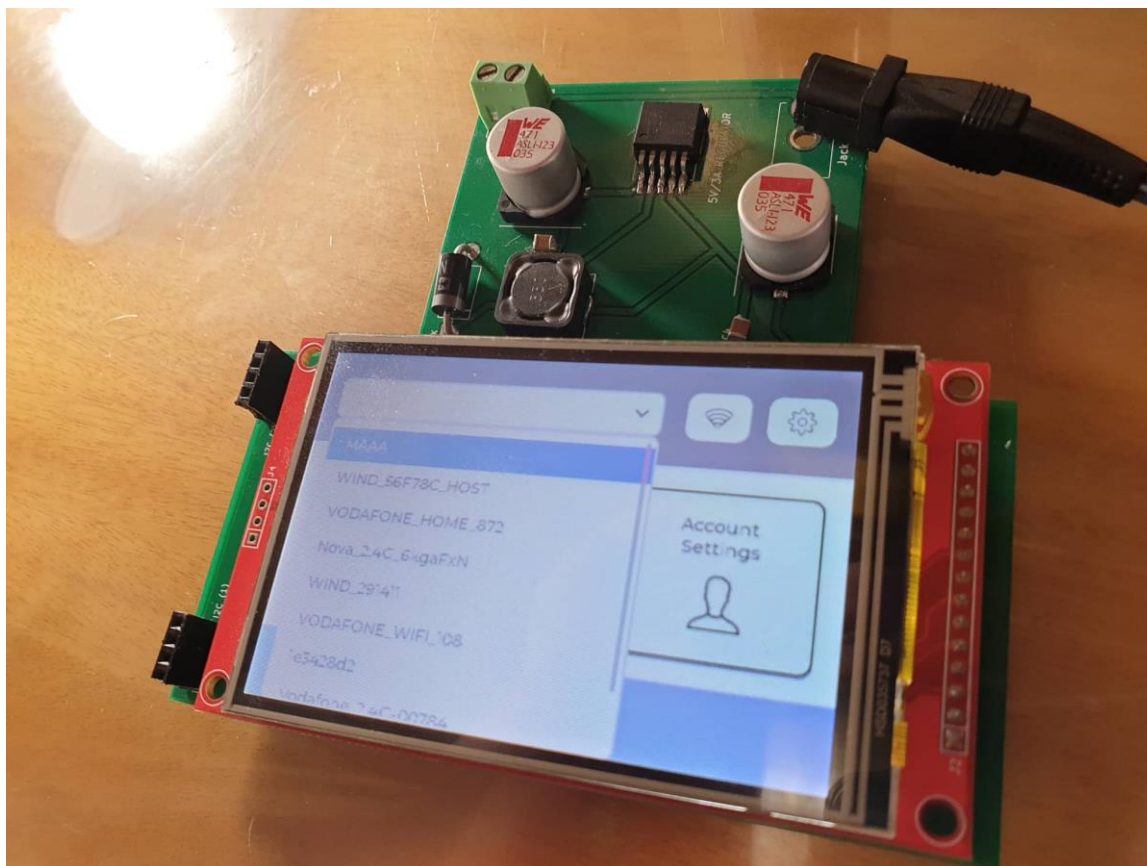
Εικόνα 4.23 - Η κάτω πλευρά της συσκευής σε τρισδιάστατο μοντέλο



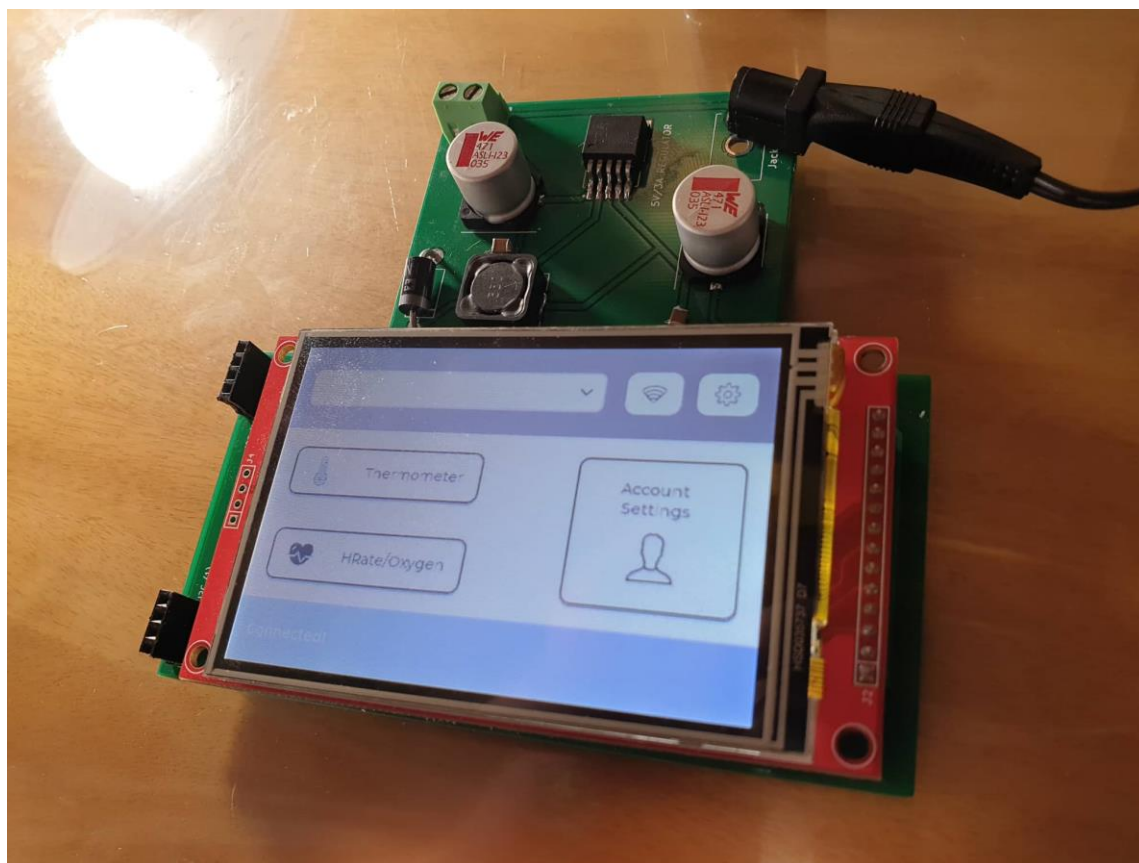
Εικόνα 4.24- Η συσκευή απο την πάνω πλευρά



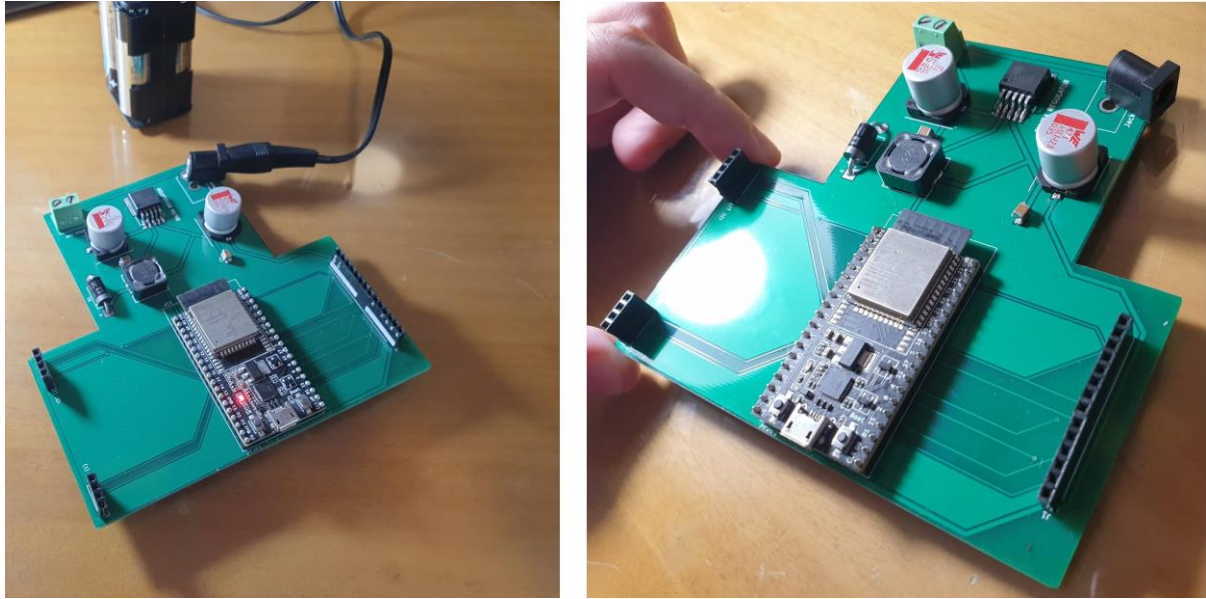
Εικόνα 4.25 - Η συσκευή απο την κάτω πλευρά



Εικόνα 4.26 – Η συσκευή με ανοιχτά τα δίκτυα



Εικόνα 4.27 – Αρχική οθόνη συσκευής



Εικόνα 4.28 – Η συσκευή χωρίς τους αισθητήρες και την οθόνη

4.6 Αισθητήρες και μετρήσεις

4.6.1 Οι Αισθητήρες

Στην συγκεκριμένη πτυχιακή για την μέτρηση της θερμοκρασίας του χρήστη χρησιμοποιήθηκε ο αισθητήρας MLX90614, ενώ για την μέτρηση του οξυγόνου και των καρδιακών παλμών χρησιμοποιήθηκε ο αισθητήρας MAX30102.

Ο MLX90614 είναι ένας αισθητήρας υπερύθρων και για αυτόν τον λόγο δεν είναι απαραίτητη η επαφή του σώματος του χρήστη με τον συγκεκριμένο αισθητήρα. Επομένως, για παράδειγμα το χέρι του χρήστη αρκεί να είναι σε μια ορισμένη απόσταση για να εκτελεστεί η μέτρηση της θερμοκρασίας του με επιτυχία. Ο MLX90614 υποστηρίζει το πρωτόκολλο I2C τελικά χρησιμοποιήσαμε για την επικοινωνία των αισθητήρων με το ESP32.

Χαρακτηριστικά MLX90614 [8]:

- Τάση λειτουργίας: 3,6V έως 5V (διατίθεται σε έκδοση 3V και 5V)
- Ρεύμα τροφοδοσίας: 1,5 mA
- Εύρος θερμοκρασίας αντικειμένου: -70 ° C έως 382,2 ° C
- Εύρος θερμοκρασίας περιβάλλοντος: -40 ° C έως 125 ° C
- Ακρίβεια: 0,02 ° C
- Οπτικό πεδίο: 80 °
- Απόσταση μεταξύ αντικειμένου και αισθητήρα: 2cm-5cm

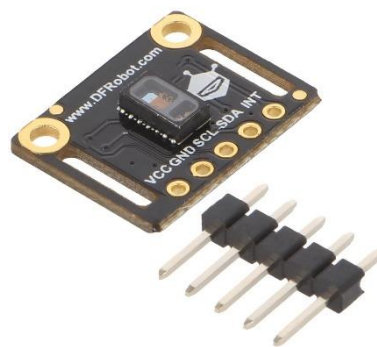


Εικόνα 4.29 - Αισθητήρας θερμοκρασίας

Ο MAX30102 είναι ένας αισθητήρας ο οποίος μετράει το επίπεδο οξυγόνου στο αίμα (SPO2) και τους καρδιακούς παλμούς με την μέθοδο PPG. Δηλαδή χρησιμοποιεί το φως για την ανίχνευση των μεταβολών του οξυγόνου και των καρδιακών παλμών. Ως αποτέλεσμα, ο χρήστης πρέπει να ακουμπά και να ασκεί ελάχιστη πίεση πάνω στον αισθητήρα. Επιπλέον, ο MAX30102 χρησιμοποιεί το πρωτόκολλο I2C για να επικοινωνήσει με το esp32, όπως δηλαδή και ο MLX90614.

Χαρακτηριστικά MAX30102 [9]:

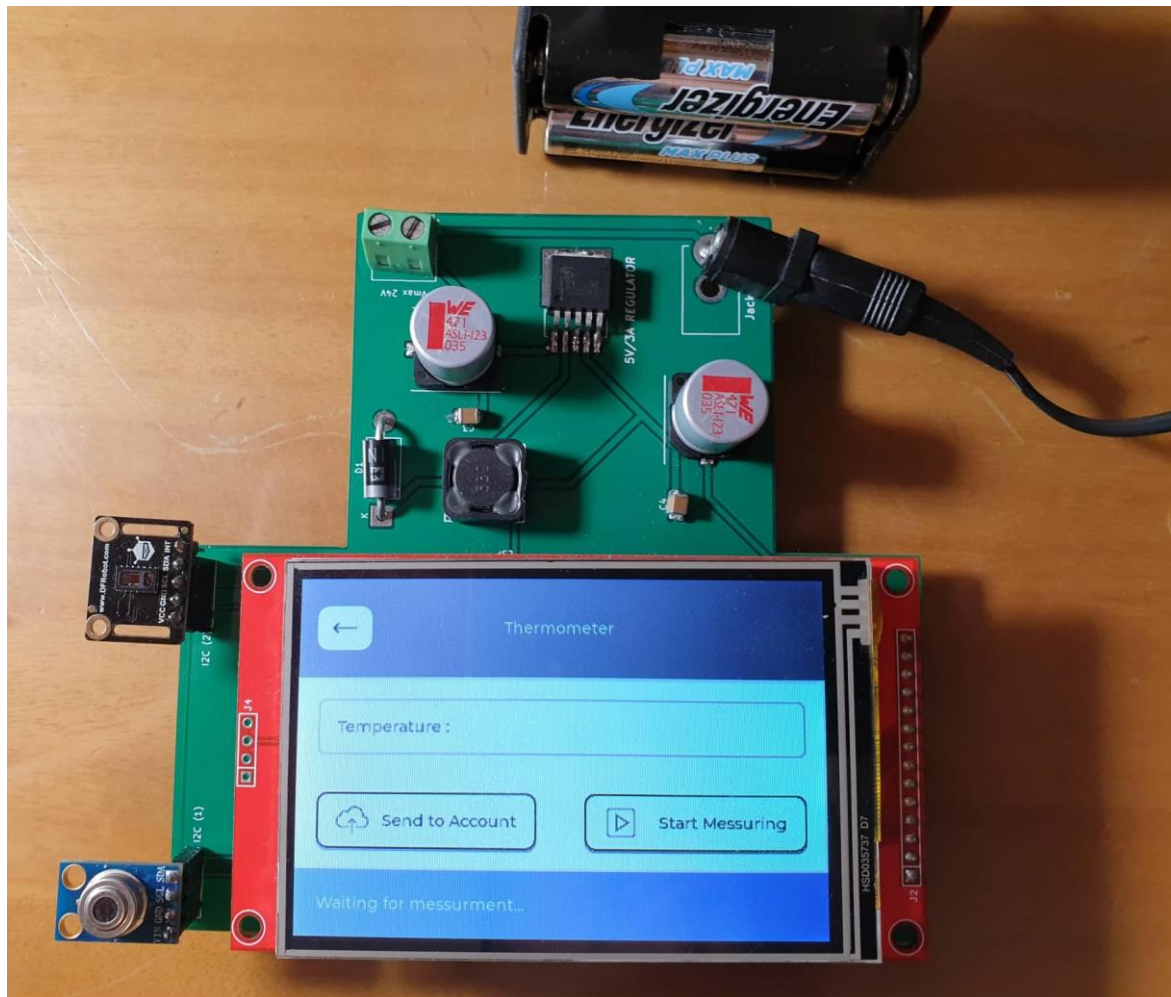
- Τροφοδοσία 3,3V~5V
- Ρεύμα εργασίας (LED σβηστό): <5 mA
- Κόκκινο IR LED Ρεύμα οδήγησης: 0-50mA
- Επικοινωνία: I2C
- Διεύθυνση I2C: 0x57
- Θερμοκρασία λειτουργίας: -40°C- 85°C



Εικόνα 4.30 – Αισθητήρας Οξυγόνου / Καρδιακών παλμών

4.6.2 Χρήση συσκευής και μετρήσεις

Ο πιο εύκολος τρόπος για να δοθεί τροφοδοσία είναι η χρήση του βύσματος DC-JACK, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 4.31 – Τροφοδοσία συσκευής

Το μόνο απαραίτητο είναι η τάση να μην είναι παραπάνω από 24V και όχι μικρότερη από 4.5V. Η μέτρηση της θερμοκρασίας γίνεται στο μενού «Thermometer» και ο χρήστης απλώς πατάει «Start Messuring».



Εικόνα 4.32 μέτρηση θερμοκρασίας

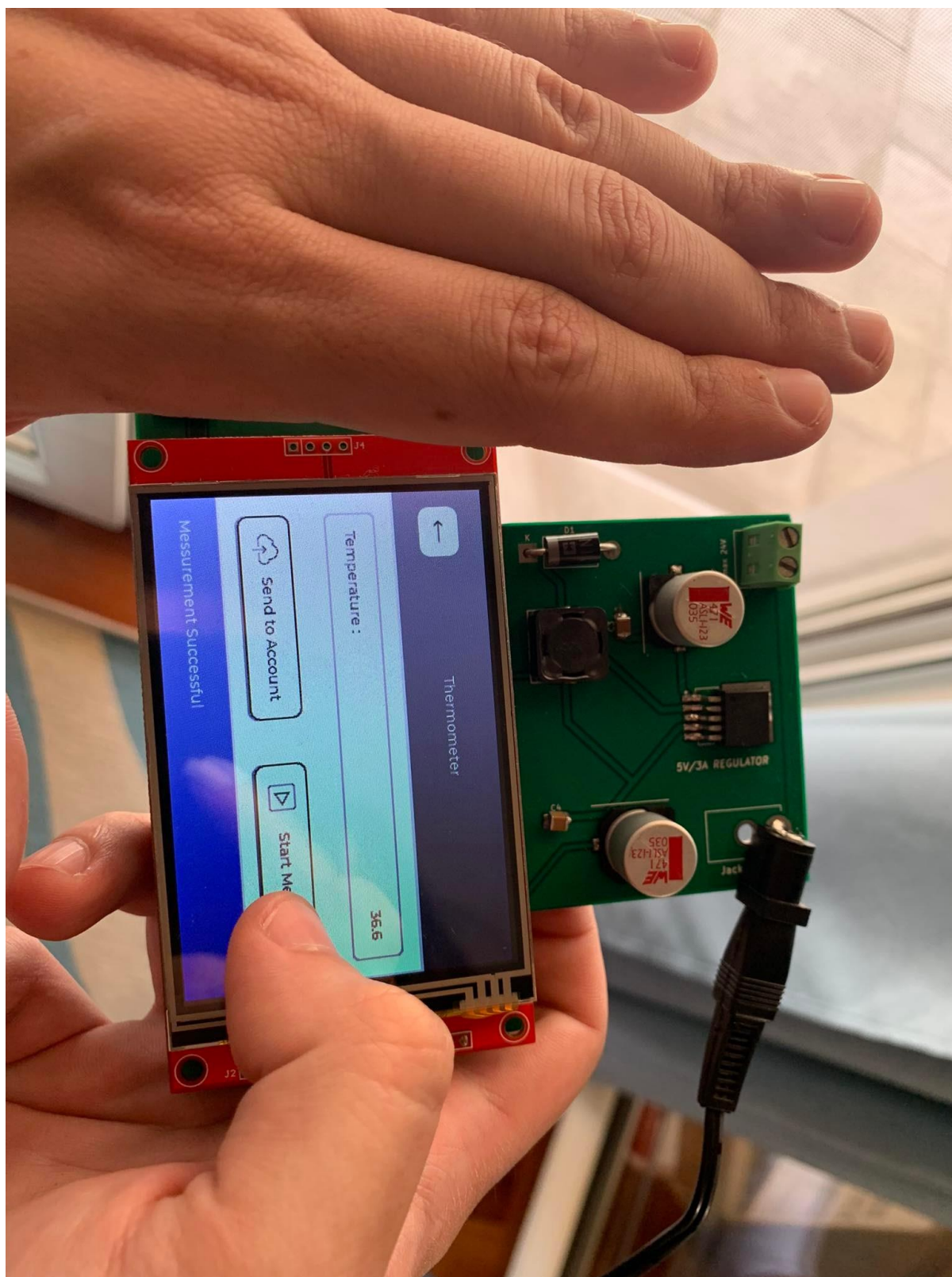
Η συσκευή έχει τέτοιο μέγεθος ώστε ο χρήστης να μπορεί να την κρατήσει σαν ηλεκτρονική κονσόλα για βιντεοπαιχνίδια. Έτσι, με τον δεξί του αντίχειρα μπορεί εύκολα να πατήσει το κουμπί «Start Messuring» και να τοποθετήσει το αριστερό του χέρι σε μικρή απόσταση από τον αισθητήρα θερμοκρασίας. Μόλις ολοκληρωθεί η μέτρηση, το μπλε πλαίσιο θα δείξει «Messurement Was Successful», όπως φαίνεται και στην εικόνα 4.29.



Εικόνα 4.33 – Χρήση αισθητήρα θερμοκρασίας

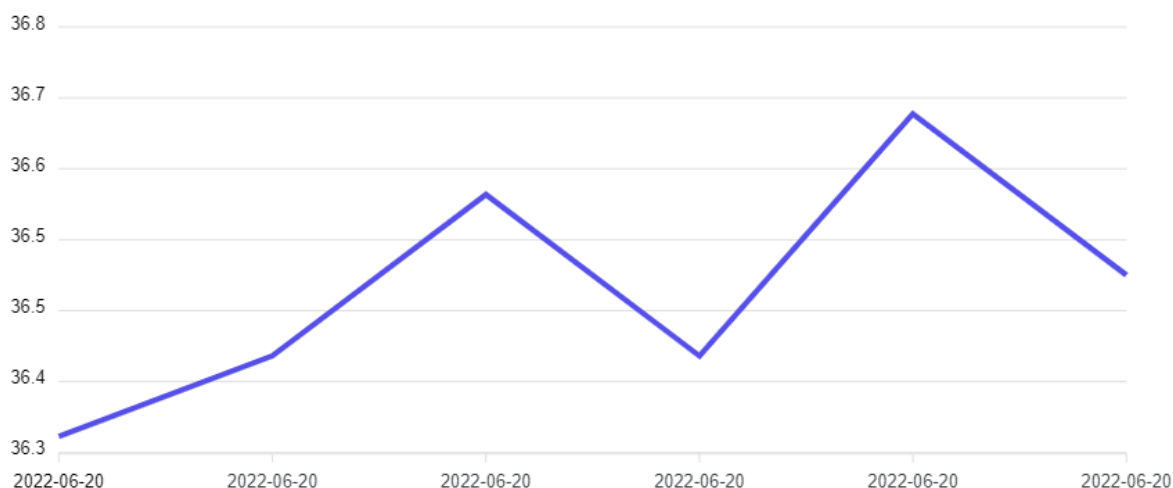


Εικόνα 4.34 – Χρήση αισθητήρα θερμοκρασίας 2



Εικόνα 4.35 – Χρήση αισθητήρα θερμοκρασίας 3

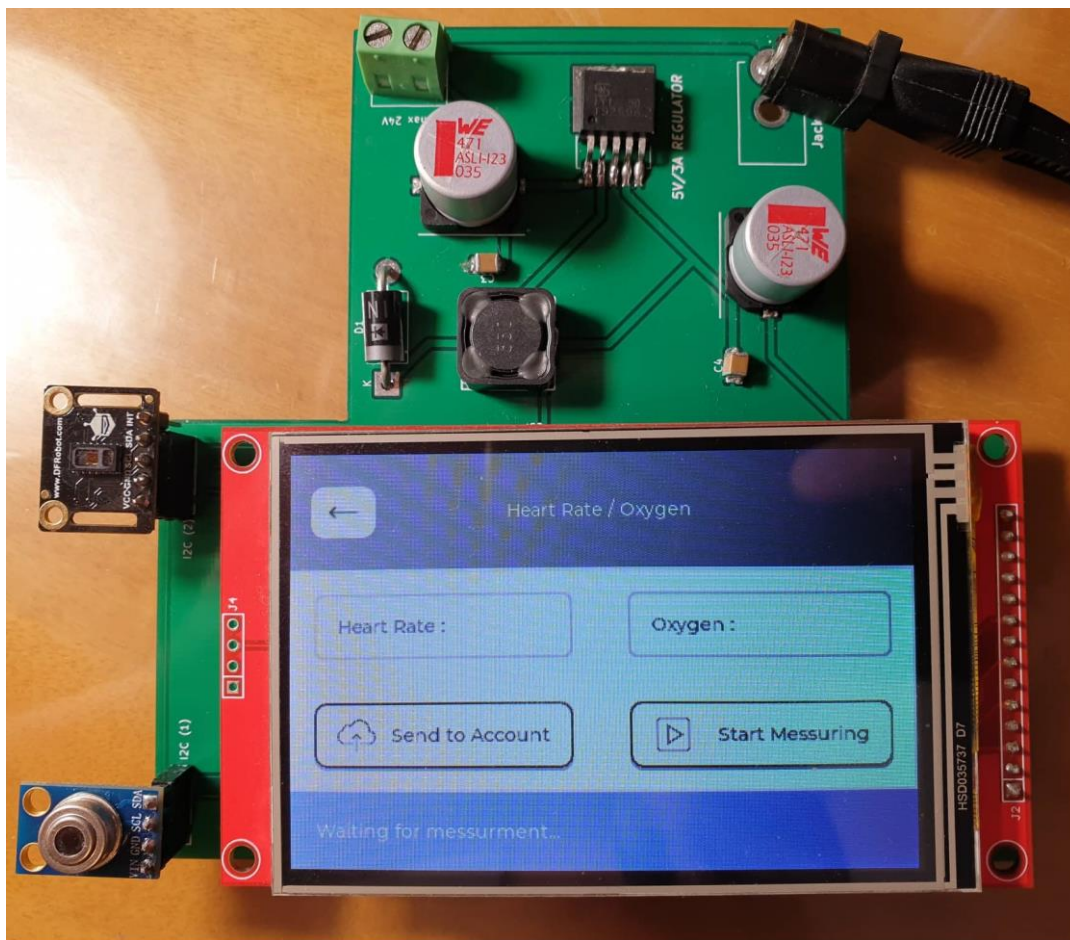
Η απόσταση που προτείνει το datasheet του αισθητήρα θερμοκρασίας για την καλύτερη δυνατή μέτρηση είναι 2-5 εκατοστά [8]. Έπειτα από μερικές συνεχόμενες μετρήσεις με απόσταση χεριού περίπου την ίδια (2-8 εκατοστά) και την αποστολή αυτών στον διαδικτυακό λογαριασμό του χρήστη έχουμε το παρακάτω διάγραμμα.

Temperature

Εικόνα 4.36 – Μετρήσεις θερμοκρασίας

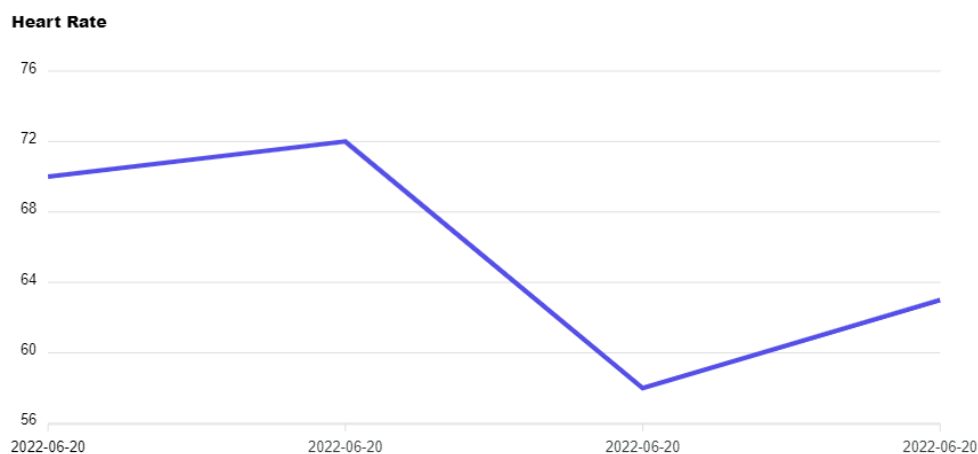
Η μικρότερη τιμή που παίρνει το θερμόμετρο είναι 36.3, ενώ η μεγαλύτερη 36.7. Σημειώνεται ότι ο συγκεκριμένος αισθητήρας δεν είναι ειδικός για ιατρική χρήση. Με την χρήση εξωτερικού θερμομέτρου η θερμοκρασία μας ήταν 36.6. Επομένως, οι μετρήσεις που πήραμε από τον αισθητήρα είχαν μικρή απόκλιση. Παρατηρούμε ότι η μέτρηση είναι 36.6 όταν το χέρι μας έχει απόσταση περίπου 5 εκατοστά. Για να χρησιμοποιήσουμε την ακρίβεια που μας δίνει ο αισθητήρας πρέπει να γίνει προσαρμογή του σε εξωτερική συσκευή, η οποία έρχεται σε απόσταση 5 εκατοστών από το δέρμα του χρήστη. Αλλά με τον τρόπο που έγινε χρήση στην συγκεκριμένη εφαρμογή, ο χρήστης μπορεί να πάρει μια καλή ένδειξη για την θερμοκρασία του.

Η μέτρηση του ποσοστού οξυγόνου SPO2 και καρδιακών παλμών γίνεται από το μενού Heart Rate/Oxygen. Ο χρήστης κατευθύνεται σε αυτό από το κουμπί της αρχικής οθόνης που ονομάζεται «HRate/Oxygen». Έπειτα, πατάει το κουμπί «Start Messuring» και τοποθετεί τον αριστερό του αντίχειρα επάνω στον αισθητήρα.



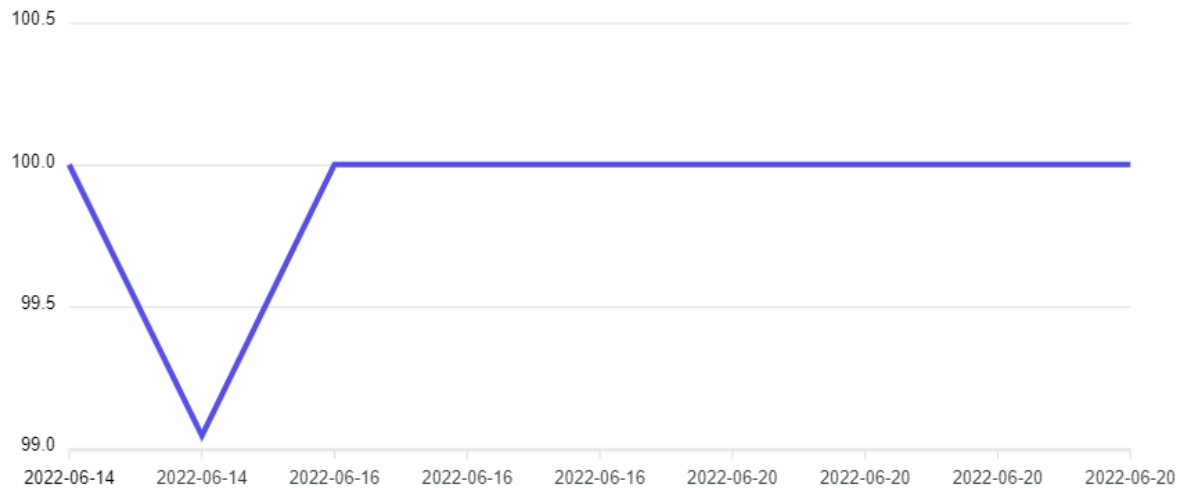
Εικόνα 4.37 – Μενού μέτρησης οξυγόνου/Παλμών

Όσο μεγαλύτερο χρονικό διάστημα το δάχτυλο βρίσκεται και ακουμπάει απαλά τον αισθητήρα, τόσο καλύτερες θα είναι και οι μετρήσεις. Σημειώνεται ότι ο παραπάνω αισθητήρας σύμφωνα με το datasheet ενδύκνεται για χρήση σε συσκευές που είναι φορέσιμες [9], όπως ένα έξυπνο ρολόι. Έπειτα από μερικές συνεχόμενες μετρήσεις και την αποστολή αυτών στον διαδικτυακό λογαριασμό του χρήστη, έχουμε τα παρακάτω διαγράμματα.



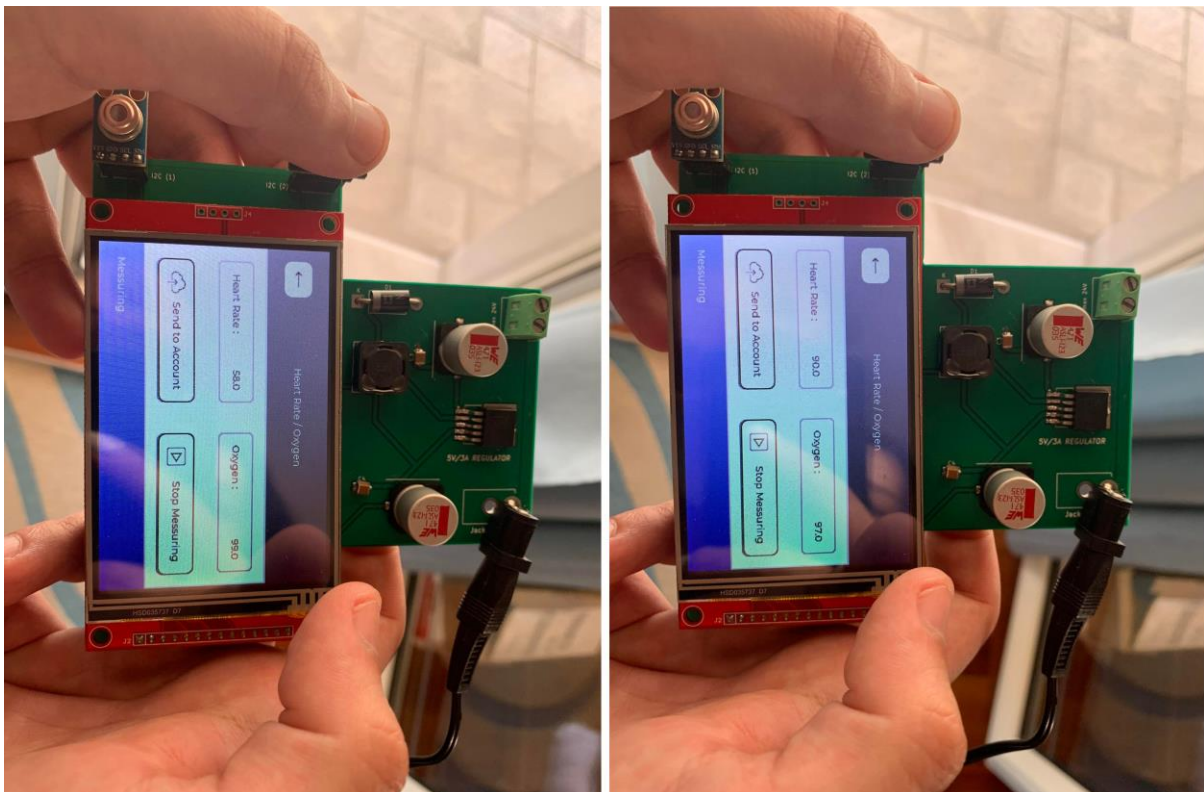
Εικόνα 4.38 – Διάγραμμα μετρήσεων καρδιακών παλμών

Oxygen Percetange

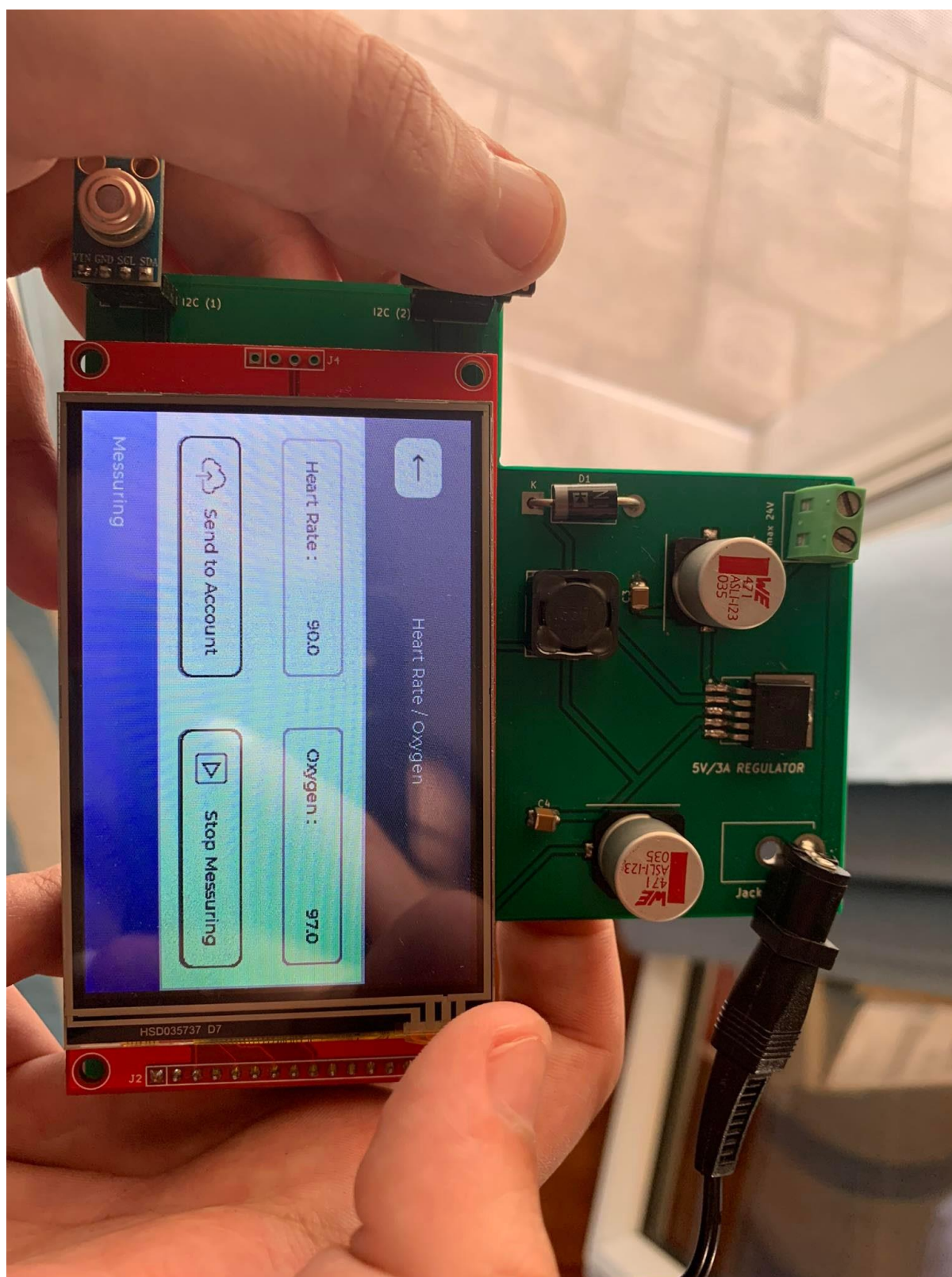


Εικόνα 4.39 – Διάγραμμα μετρήσεων ποσοστό οξυγόνου (Παλμική οξυμετρία)

Οι πρώτες δύο μετρήσεις διήρκεσαν περίπου 4 δευτερόλεπτα, ενώ οι τελευταίες δύο περισσότερο γύρω στα 20 δευτερόλεπτα. Το ιδανικό είναι ο αισθητήρας να προσαρμοστεί σε σημείο στο οποίο ο χρήστης πιέζει με σταθερή ισχύ. Στην συγκεκριμένη εφαρμογή εάν ο χρήστης ακουμπήσει απαλά και σταθερά με τον αντίχειρα του για 20-30 δευτερόλεπτα θα παρθούν αρκετά ακριβείς ενδείξεις. Η απόκλιση στους καρδιακούς παλμούς είναι $\pm 20\%$ από τους πραγματικούς.



Εικόνα 4.40 – Χρήση αισθητήρα οξυγόνου/καρδιακών παλμών



Εικόνα 4.41 - Χρήση αισθητήρα οξυγόνου/καρδιακών παλμών 2

4.7 Επίλογος

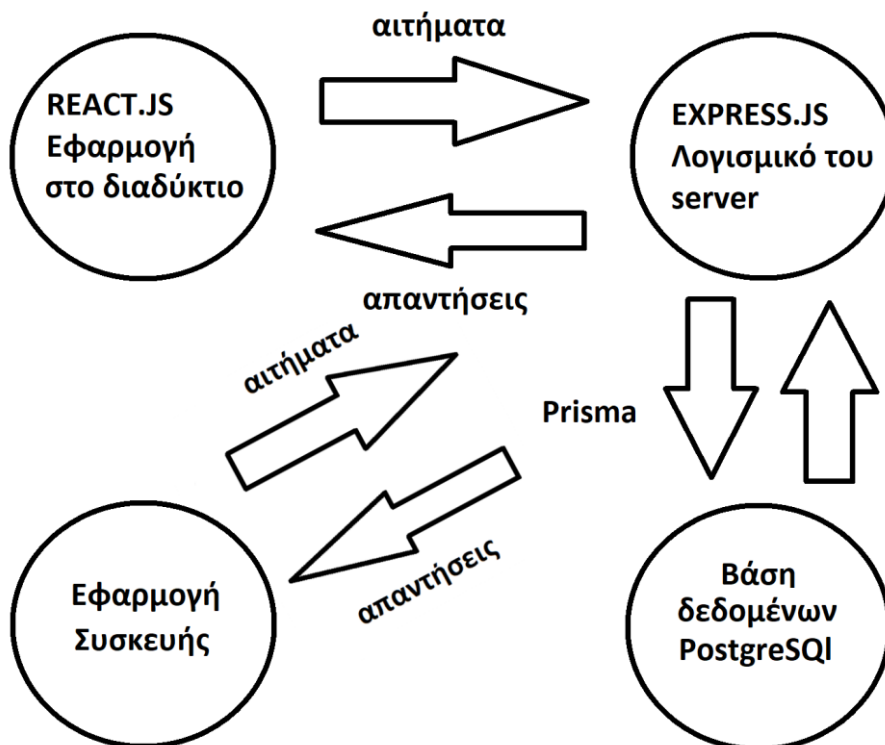
Η πλατφόρμα ανάπτυξης ηλεκτρονικής συσκευής που επιλέχθηκε είναι η ESP-32.Μας παρέχει περιφερειακά όπως WiFi, Bluetooth και υποστήριξη στην χρήση λειτουργικού συστήματος πραγματικού χρόνου, συγκεκριμένα το FreeRtos.Έγινε χρήση οθόνης με λειτουργία αφής για την

αλληλεπίδραση του χρήστη με το λογισμικό της συσκευής. Το λογισμικό αποτελείται από δύο κύρια κομμάτια. Το πρώτο είναι υπεύθυνο για την ανανέωση των pixel της οθόνης την στιγμή που ο χρήστης αλληλεπιδρά με αυτή και το δεύτερο για την επιτήρηση, την ρύθμιση και την εκτέλεση κρίσιμων εσωτερικών λειτουργιών, όπως αυτή της σύνδεσης της συσκευής στο διαδίκτυο. Η σύνδεση των αισθητήρων γίνεται εξωτερικά με την βοήθεια επαφών I2C. Η παροχή τροφοδοσίας μπορεί να γίνει με τρεις τρόπους. Με την σύνδεση καλωδίου USB απευθείας στο ESP32, με την χρήση της κλέμας ή με την χρήση του βύσματος DC-JACK. Η ρύθμιση τάσης επιτυγχάνεται με τη χρήση ολοκληρωμένου ρυθμιστή τάσεως με διακόπτη. Η τελική ψηφιακή σχεδίαση της ηλεκτρονικής πλακέτας (PCB) που περιέχει όλα τα παραπάνω στοιχεία πραγματοποιείται στο λογισμικό kiCad.

Κεφάλαιο 5ο: Η διαδικτυακή εφαρμογή

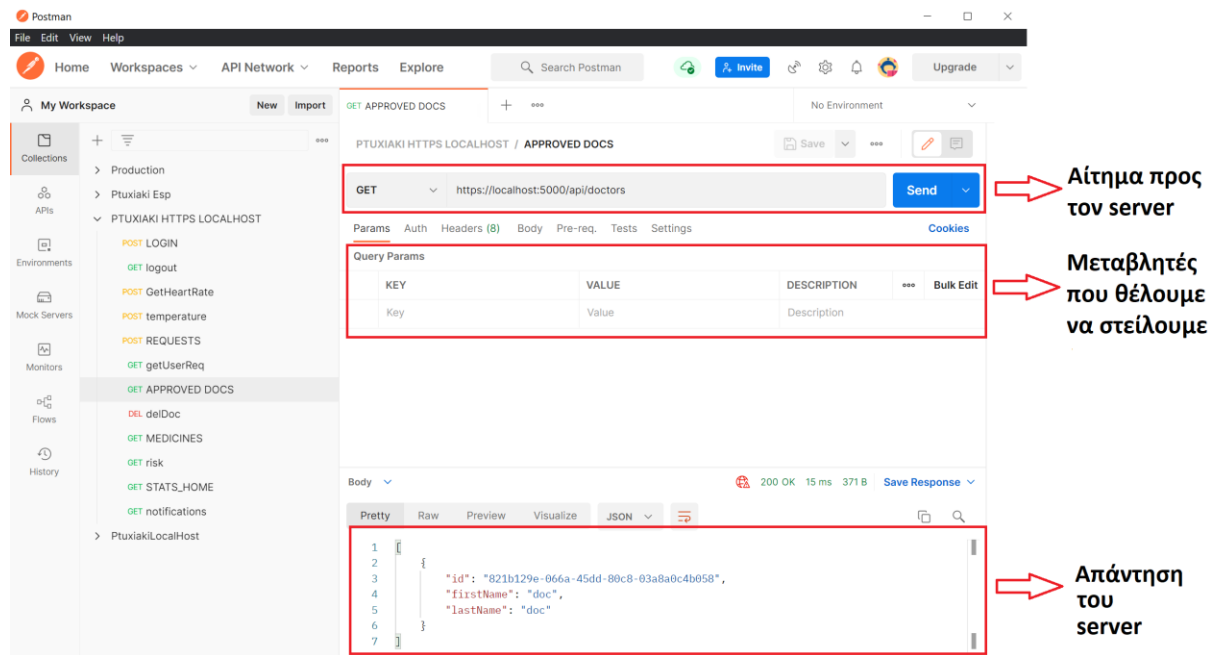
5.1 Εισαγωγή

Ο χρήστης μπορεί να επισκεφτεί την εφαρμογή στην σελίδα <https://ptuxiakiangstel.herokuapp.com/> . Στον οδηγό λογισμικού παρέχονται και στοιχεία για την σύνδεση σε δοκιμαστικούς λογαριασμούς. Το πιο σημαντικό κομμάτι της διαδικτυακής εφαρμογής είναι το λογισμικό του backend, δηλαδή αυτό του server. Αυτό είναι υπεύθυνο για την διατήρηση της σύνδεσης των λογαριασμών των χρηστών και για την ασφάλεια της μεταφοράς των δεδομένων. Όμως επειδή ο χρήστης δεν μπορεί εύκολα να στέλνει αιτήματα στον server, έπρεπε να φτιαχτεί κάποια εφαρμογή και να μπαίνει σε αυτή με την χρήση λογισμικού πλοήγησης διαδικτύου. Για αυτή την δουλειά επιλέχθηκε η βιβλιοθήκη React.



Σχήμα 5.1 - Αρχιτεκτονική λογισμικού διαδικτυακής εφαρμογής

Όπως φαίνεται στο σχήμα 5.1 το frontend της εφαρμογής είναι μια όμορφη παρουσίαση δεδομένων και διευκολύνει τον χρήστη να στείλει αιτήματα HTTP και να λαμβάνει απαντήσεις με το πάτημα μερικών κουμπιών, χωρίς να χρειάζεται να κάνει κάτι άλλο. Ακριβώς το ίδιο κάνει και η εφαρμογή που έχει η συσκευή μέτρησης των βιομετρικών. Σημειώνεται ότι ένας χρήστης που ασχολείται με την τεχνολογία, μπορεί να στείλει αιτήματα και να λάβει απαντήσεις από τον server χωρίς να κάνει χρήση την εφαρμογή του διαδικτύου ή την εφαρμογή της συσκευής. Με αυτόν τρόπο κιόλας αναπτύξαμε και δοκιμάσαμε το λογισμικό του server. Αυτό γίνεται με τρίτες εφαρμογές βοηθάνε να στείλει κάποιος αιτήματα με ή χωρίς μεταβλητές. Για παράδειγμα η εφαρμογή που χρησιμοποιήσαμε ονομάζεται POSTMAN. Αυτή η εφαρμογή βοηθάει στο να αποθηκεύεις τα αιτήματα έτσι ώστε να μην χρειάζεται να τα ξαναγράφεις, εμφανίζει την απάντηση του server και την κατάσταση της και γενικότερα επιτρέπει την ρύθμιση πολλών στοιχείων, όπως τον τύπο της απάντησης του server, δηλαδή να φαίνεται σαν JSON, XML κλπ. Μπορείς να αποστείλεις μεταβλητές στον server, να προσθέσεις ή να αφαιρέσεις διάφορες επικεφαλίδες (headers) που στέλνεις με τα HTTP αιτήματα, για παράδειγμα ο προγραμματιστής μπορεί να προσθέτει κάποιο cookie σε αυτές.



Εικόνα 5.2 - Στιγμιότυπο του λογισμικού Postman

Βλέποντας στην παραπάνω φωτογραφία η εφαρμογή αφού λάβει την απάντηση του server , μπορεί να την επεξεργαστεί και να την τοποθετήσει σε όποιο σημείο θέλει, αυτή είναι και η λογική που χρησιμοποιήθηκε.

5.2 Η εφαρμογή απο πλευρά ασθενή

5.2.1 Σελίδα σύνδεσης

Η αρχική σελίδα της εφαρμογής φαίνεται παρακάτω:

Welcome Back

Welcome back! Please enter your details...

Email :

Enter your Email

Password :

☐ Remember for 30 days

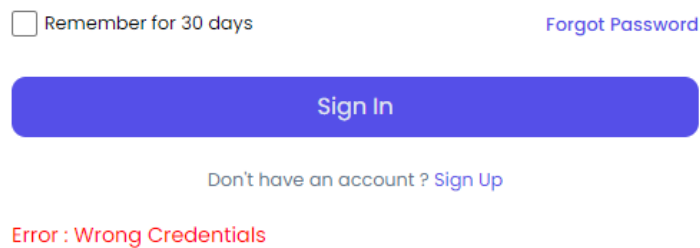
[Forgot Password](#)

Sign In

Don't have an account ? [Sign Up](#)

Εικόνα 5.3 - Σελίδα σύνδεσης

Σε αυτό το σημείο ο χρήστης μπορεί να πληκτρολογήσει το email του και τον κωδικό πρόσβασης για να συνδεθεί στην εφαρμογή. Επίσης, υπάρχει κουτάκι που μπορεί να επιλέξει για να παραμείνει συνδεδεμένος για 30 μέρες χωρίς να χρειαστεί να ξαναπληκτρολογήσει τα στοιχεία του. Η προεπιλεγμένη επιλογή που δεν φαίνεται είναι μία μέρα. Δίπλα υπάρχει επιλογή που μπορεί να πατήσει σε περίπτωση που έχει ξεχάσει τον κωδικό του πρόσβασης έτσι ώστε να τον επαναφέρει και να βάλει καινούριο. Από κάτω είναι το κουμπί «Sign In» και ουσιαστικά στέλνει τα στοιχεία στον server έτσι ώστε να συνδεθεί ο χρήστης. Τέλος τέρμα κάτω υπάρχει επιλογή που τον κατευθύνει στην σελίδα που μπορεί να εγγραφεί αν δεν έχει λογαριασμό. Σε περίπτωση που τα στοιχεία που πληκτρολόγησε ο χρήστης είναι λάθος εμφανίζεται το παρακάτω μήνυμα.



Εικόνα 5.4 - Σφάλμα σύνδεσης

5.2.2 Σελίδα εγγραφής

Η σελίδα για την εγγραφή νέου λογαριασμού :

Welcome !

Please enter your details for the registration...

First Name :

Last Name :

Email :

Password :

Confirm Password :

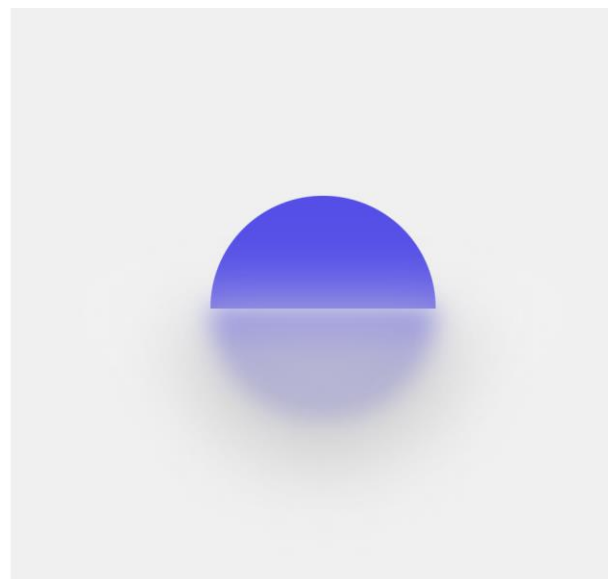
Role :

Patient

Patient

Doctor

Have an account? [Sign In](#)



Εικόνα 5.5 - Σελίδα σύνδεσης

Αρχικά ο χρήστης πρέπει να πληκτρολογήσει το όνομα και το επώνυμο του. Ακριβώς από κάτω στο πλαίσιο email πρέπει να πληκτρολογήσει το email που θα χρησιμοποιεί για την σύνδεση στον λογαριασμό του. Τα δύο επόμενα πλαίσια αφορούν τον κωδικό πρόσβασης. Αυτά τα δύο πλαίσια πρέπει να είναι ακριβώς ίδια για να είναι βέβαιο ότι ο χρήστης θέλει να χρησιμοποιήσει ακριβώς αυτόν τον κωδικό πρόσβασης. Το τελευταίο πλαίσιο αφορά την επιλογή ρόλου που θέλει να επιλέξει ο χρήστης, δηλαδή αν είναι ασθενής ή γιατρός. Τέλος υπάρχει κουμπί «Sign Up» για να ολοκληρωθεί η εγγραφή και από κάτω σε περίπτωση που ο χρήστης έχει ήδη λογαριασμό η επιλογή «Sign In» τον

πάει πίσω στην αρχική οθόνη που είναι για να κάνει σύνδεση στον λογαριασμό του. Όλα τα σφάλματα που μπορεί να εμφανιστούν στην συγκεκριμένη σελίδα είναι τα εξής:

- firstName: Please enter your First Name, αν το πλαίσιο του ονόματος είναι άδειο
- lastName: Please enter your Last Name, αν το πλαίσιο του επιθέτου είναι άδειο
- email: Must Be an Email, αν το πλαίσιο του email δεν είναι έγκυρο ή είναι άδειο
- password: Must be at least 5 chars, έχουμε βάλει σαν όρο ο κωδικός πρόσβασης να είναι τουλάχιστον 5 χαρακτήρες
- password: Doesn't Match : Αν τα δύο πλαίσια Password και Confirm Password δεν είναι ίδια.

firstName : Please enter you First Name
lastName : Please enter you Last Name
email : Must Be an Email
password : Must be at least 5 chars
password : Doesn't Match

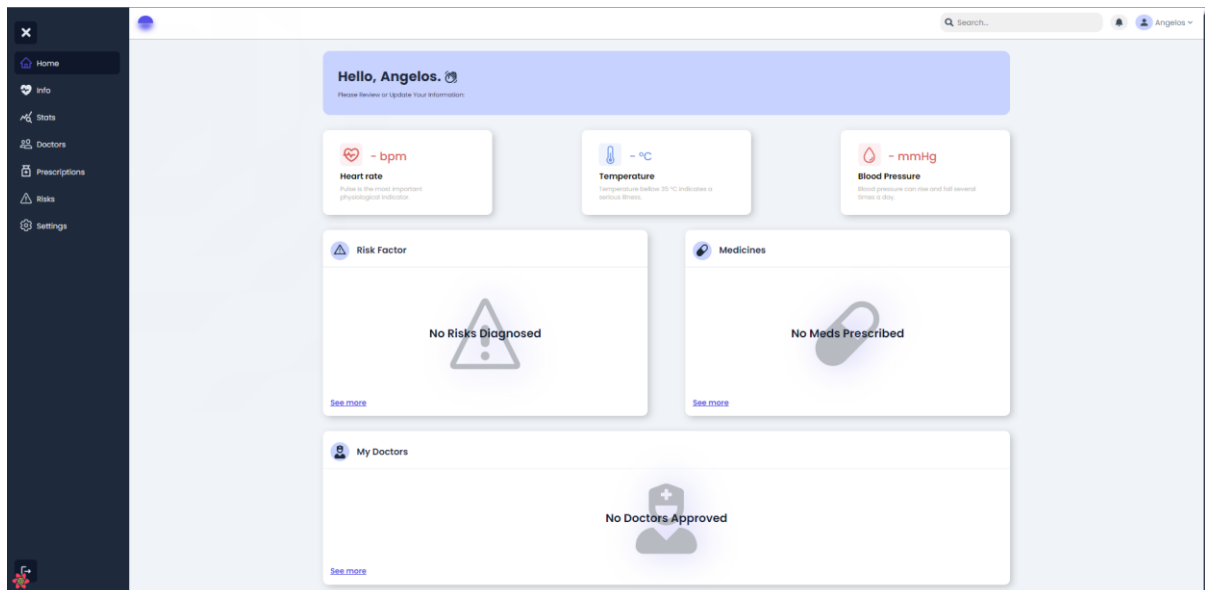
Sign Up

Have an account? [Sign In](#)

Εικόνα 5.6 - Σφάλματα εγγραφής

5.2.3 Αρχική σελίδα

Στην επόμενη φωτογραφία είναι η αρχική οθόνη που βλέπει ένας ασθενής όταν η σύνδεση του είναι επιτυχής:



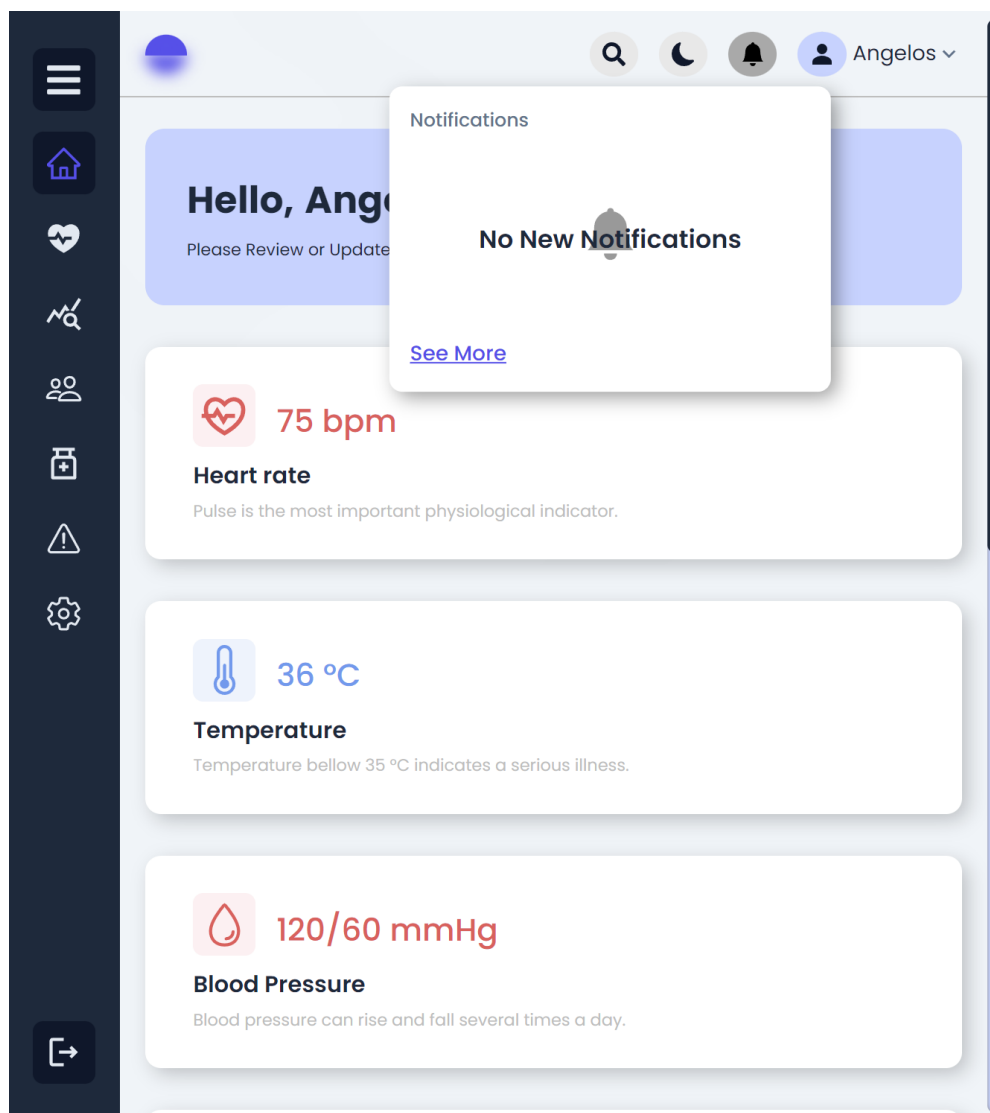
Εικόνα 5.7 - Αρχική σελίδα

Έχει γίνει σμίκρυνση της σελίδας έτσι ώστε να φαίνεται όλο το περιεχόμενο της ταυτόχρονα. Στην πραγματικότητα ανάλογα με την ανάλυση της συσκευής που χρησιμοποιεί ο χρήστης τα στοιχεία είναι μεγαλύτερα ή μικρότερα. Αρχικά στα αριστερά της σελίδας, δηλαδή στο πλαίσιο που είναι σκούρο μπλε είναι οι υποσελίδες της εφαρμογής. Εκείνο το σημείο μπορεί να χρησιμοποιηθεί για την γρήγορη πλοήγηση του χρήστη. Οι υποσελίδες θα αναλυθούν περισσότερο παρακάτω. Επιπλέον, υπάρχει δεύτερο μεγαλύτερο πλαίσιο, που αποτελείται από δύο μικρότερα υποπλαίσια, το άσπρο και

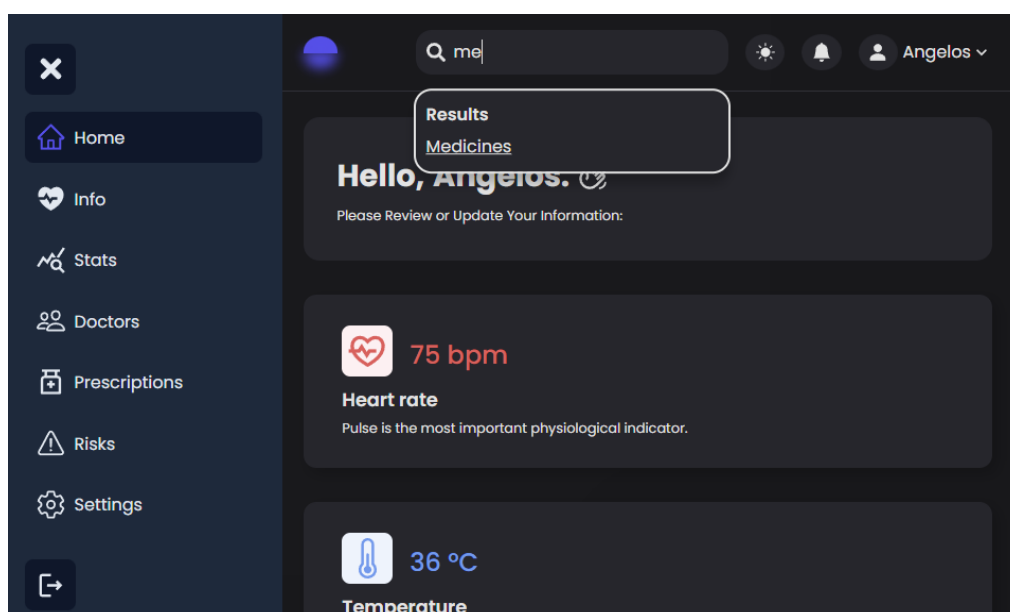
το γκρι.Ο χρήστης καθώς κατεβαίνει προς τα κάτω το αριστερό μενού δεν αλλάζει θέση, αλλά παραμένει στο ίδιο σημείο σταθερό. Ωστε αν ο χρήστης θελήσει να πάει σε άλλη σελίδα να μην χρειαστεί να ανεβεί πάλι τέρμα πάνω. Επίσης το σκούρο μπλε και το άσπρο μενού παραμένει ίδιο σε όλες τις σελίδες.Το περιεχόμενο που αλλάζει είναι μόνο αυτό που βρίσκεται στο γκρι υποπλαίσιο.

Το πρώτα τρία πλαίσια στο γκρι , είναι οι βασικές μετρήσεις που μπορεί να αποθηκεύσει ο ασθενής δηλαδή καρδιακούς παλμούς , θερμοκρασίες και οι μετρήσεις της πίεσης.Αυτή την στιγμή αντί για να νούμερα φαίνεται μόνο μία παύλα, διότι δεν υπάρχει καμία μέτρηση στον λογαριασμό.Αν όμως υπάρχουν μετρήσεις εμφανίζονται οι πιο πρόσφατες.Ακριβώς από κάτω είναι τα «ρίσκα» που αφορούν τον ασθενή. Για παράδειγμα σε εκείνο το σημείο μπορεί να γραφτεί από έναν γιατρό ότι ο ασθενής έχει πόνο στην πλάτη του πολύ καιρό και ότι λόγω αυτού δεν πρέπει να κάνει ορισμένες κινήσεις. Το δίπλα πλαίσιο έχει φάρμακα που έχουν γραφτεί από έναν γιατρό.Σημειώνεται ότι επειδή αυτή η σελίδα είναι η αρχική αν υπάρχουν πάνω από δύο ρίσκα ή δύο φάρμακα δεν θα εμφανιστούν όλα. Ο χρήστης θα πρέπει να πατήσει την επιλογή «See More» για να μεταβεί στην κατάλληλη υποσελίδα ή μπορεί να χρησιμοποιήσει το αριστερό σκούρο μπλε μενού.Τέλος , το τελευταίο πλαίσιο είναι οι γιατροί που έχει δώσει άδεια ο ασθενής και μπορούν να του στέλνουν αιτήματα για φάρμακα, ρίσκα και να έχουν την δυνατότητα να δουν τις μετρήσεις των βιομετρικών του.

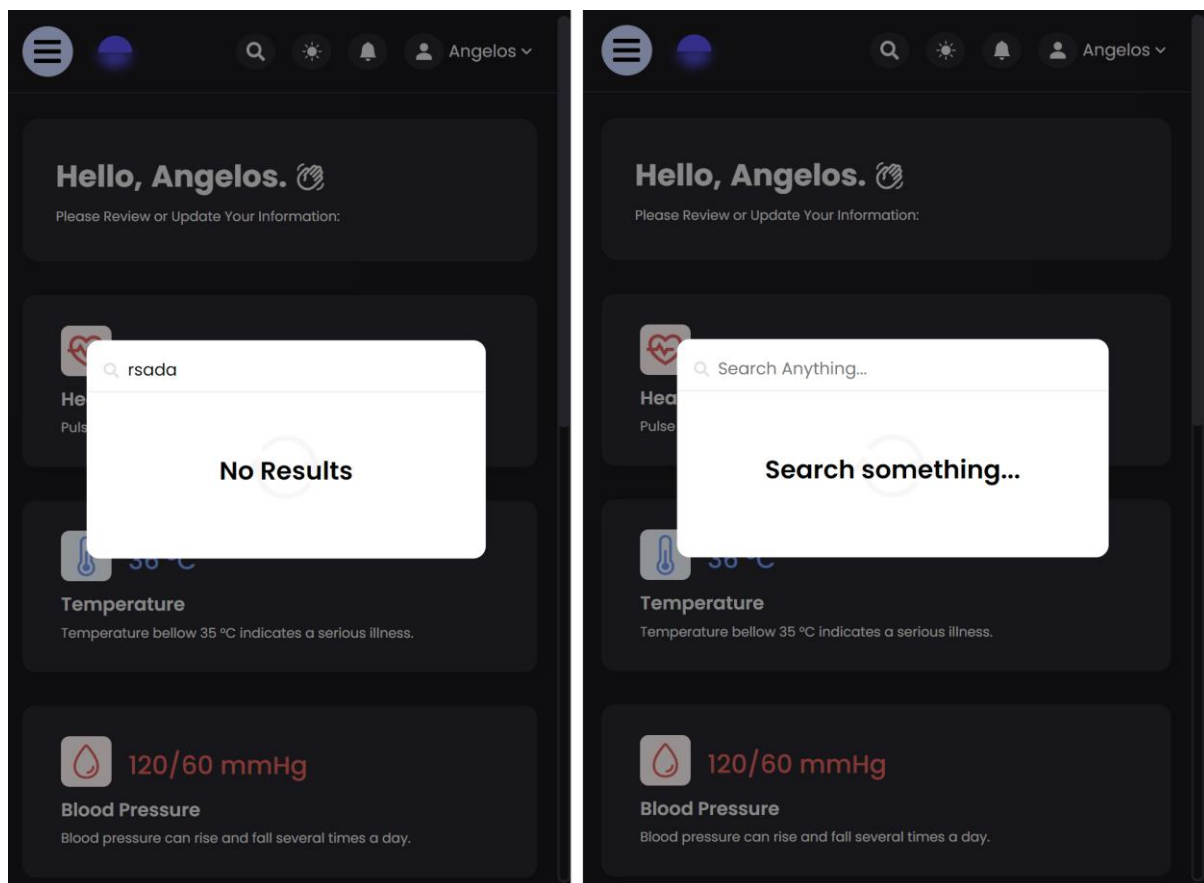
Σχετικά με το άσπρο πλαίσιο, αυτό αποτελείται από το λογότυπο της εφαρμογής, που αν το επιλέξει ο χρήστης θα μεταφερθεί στην αρχική σελίδα που βλέπουμε παραπάνω σε περίπτωση που δεν βρίσκεται ήδη εκεί. Στο δεξί μέρος υπάρχει μια μπάρα αναζήτησης που βοηθάει κι αυτή στην πλοήγηση, ένα κουμπί με καμπανάκι που εμφανίζει τις τελευταίες ειδοποιήσεις. Για παράδειγμα αν ο γιατρός έχει στείλει ένα αίτημα για να συνταγογραφήσει φάρμακο, θα δημιουργηθεί μια ειδοποίηση στον ασθενή.Το τελευταίο κουμπί που αναγράφει το όνομα του χρήστη, αν πατηθεί θα βγει υποπλαίσιο που περιέχει δύο επιλογές.Η πρώτη είναι να οδηγηθεί ο χρήστης στις ρυθμίσεις και η δεύτερη είναι να αποσυνδεθεί απο την εφαρμογή.



Εικόνα 5.8 - Αρχική σελίδα με τις ειδοποιήσεις ανοιχτές

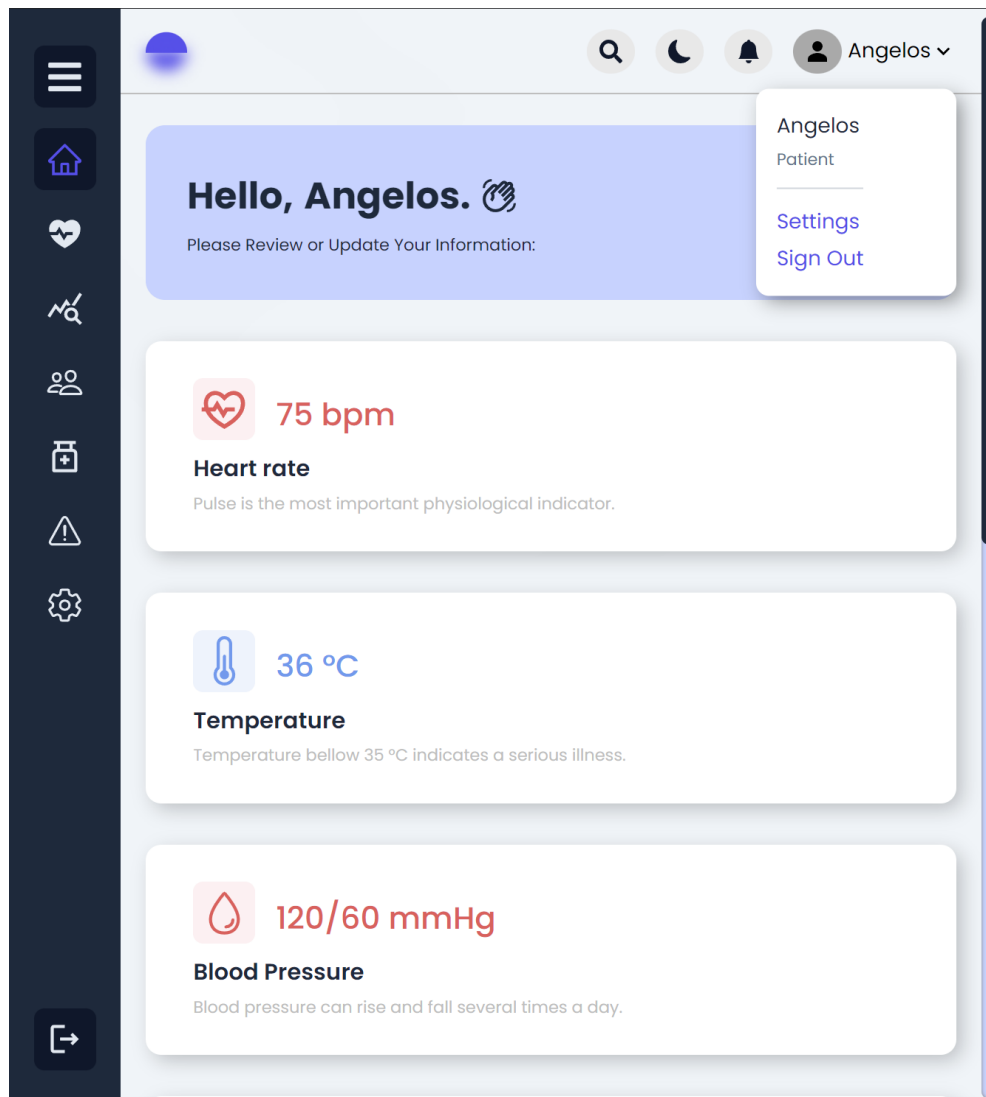


Εικόνα 5.9 - Αναζήτηση



Εικόνα 5.10 - Μπάρα αναζήτησης σε μικρή οθόνη

Σε αυτήν την φωτογραφία φαίνεται το υποπλαίσιο του κουμπιού για τις ειδοποιήσεις. Βλέπουμε ότι δεν υπάρχουν νέες ειδοποιήσεις και ότι υπάρχει επιλογή «see more» για την μετάβαση του χρήστη στην σελίδα με όλες τις ειδοποιήσεις. Επίσης, το παρακάτω στιγμιότυπο Ααφορά το πως φαίνεται η διαδικτυακή εφαρμογή σε μία οθόνη που έχει περίπου την ανάλυση ενός τάμπλετ. Υπάρχουν διάφορες αλλαγές, όπως ότι η αναζήτηση στο άσπρο πλαίσιο έχει μετατραπεί πλέον σε κουμπί και τα υποπλάισια του γκρι έχουν αλλάξει θέση. Αυτή η σχεδίαση έχει γίνει για να διευκολυνθεί ο χρήστης της εφαρμογής. Γενικότερα, όσο μικραίνει η ανάλυση της οθόνης τίνουμε να βάζουμε τα στοιχεία το ένα κάτω από το άλλο για να χωρέσουν στον χώρο που υπάρχει και να φαίνονται με ευκολία. Αν είναι μεγαλύτερη και υπάρχει χώρος τότε τα βάζουμε το ένα δίπλα από το άλλο, όπως φαίνεται στην φωτογραφία που έχει γίνει σμίκρυνση της σελίδας. Πάντοτε πρέπει να λαμβάνεται υπόψιν το κριτήριο της εύκολης αλληλεπίδρασης.



Εικόνα 5.11- Αρχική σελίδα με ανοιχτό του υποπλαισίου του προφίλ

Στο παραπάνω στιγμιότυπο φαίνεται το υποπλαίσιο που εμφανίζεται αν ο χρήστης πατήσει το τελευταίο εικονίδιο, το οποίο αναγράφει και το όνομα του. Στο υποπλαίσιο αυτό αρχικά αναγράφει το μικρό όνομα του χρήστη, ακριβώς από κάτω τον ρόλο του λογαριασμού. Στην συγκεκριμένη περίπτωση «Patient» δηλαδή ασθενής. Τέλος, όπως αναφέρθηκε, υπάρχει επιλογή μετάβασης στις ρυθμίσεις ή αποσύνδεσης από τον λογαριασμό.

5.2.4 Σελίδα μετρήσεων

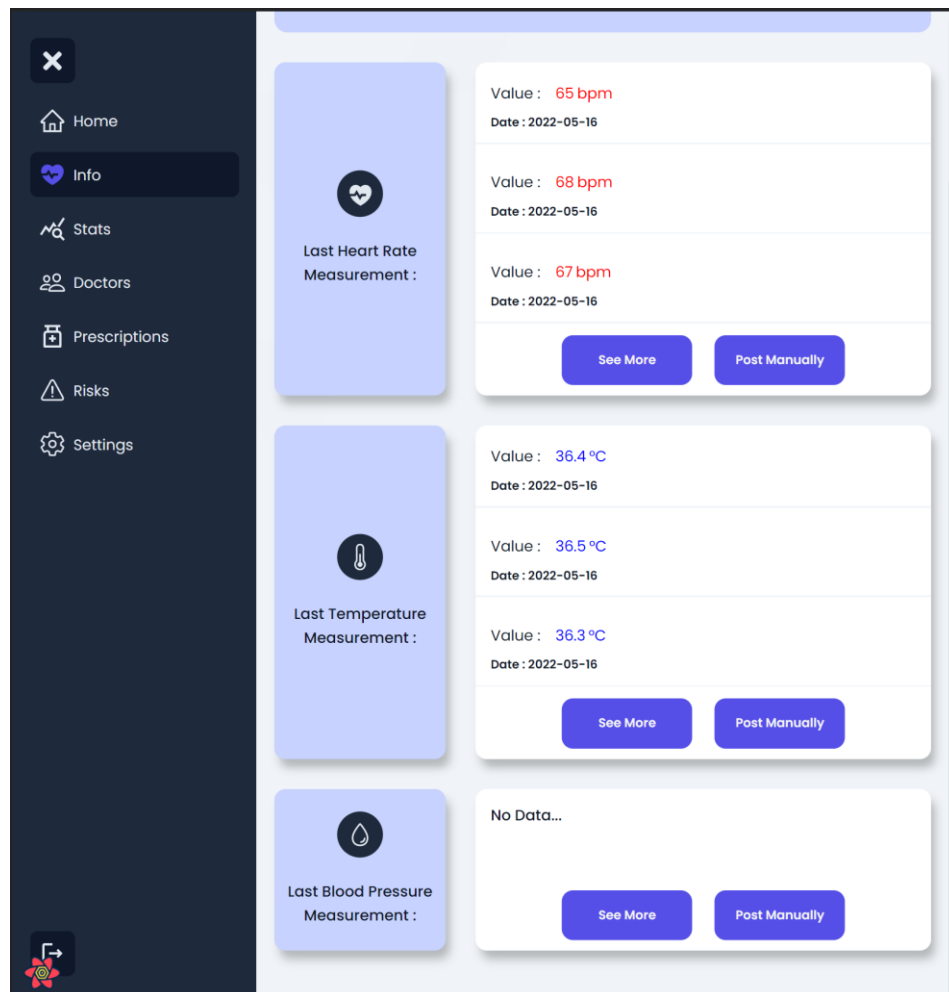
5.2.4.1 Πρόσφατες μετρήσεις

Η επόμενη βασική υποσελίδα της διαδραστικής εφαρμογής είναι οι πληροφορίες του ασθενή, δηλαδή τα βιομετρικά που ανεβάζει. Τα βιομετρικά αυτά αποτελούνται όπως φαίνεται και στην αρχική σελίδα από:

- Καρδιακούς παλμούς με μονάδα μέτρησης bpm (beats per minute)
- Θερμοκρασία σώματος με μονάδα °C (celsius)
- Αρτηριακή πίεση με μονάδα mmHg (millimetres of mercury), αναγράφεται και η Systolic blood pressure και η Diastolic blood pressure , δηλαδή συστολική και διαστολική.

- Επίπεδο ζυγόνο στο αίμα που αναφέρεται με ποσοστό (%)

Σε αυτή την σελίδα φαίνονται οι τελευταίες τρεις τιμές που έχει ανεβάσει ο ασθενής και από κάτω υπάρχουν κάποια κουμπία. Το πρώτο οδηγεί σε σελίδα που υπάρχουν όλες οι τιμές και δύναται να χρησιμοποιηθούν διάφορα φίλτρα, όπως ημερομηνία για να πάρουμε αποτελέσματα για συγκεκριμένο χρονικό πλαίσιο. Το δεύτερο κουμπί οδηγεί σε σελίδα όπου ο ασθενής μπορεί να προσθέσει νέες τιμές στην κατηγορία του βιομετρικού που θέλει αυτός, χωρίς την χρήση της εξωτερικής συσκευής. Πιο αναλυτικά:

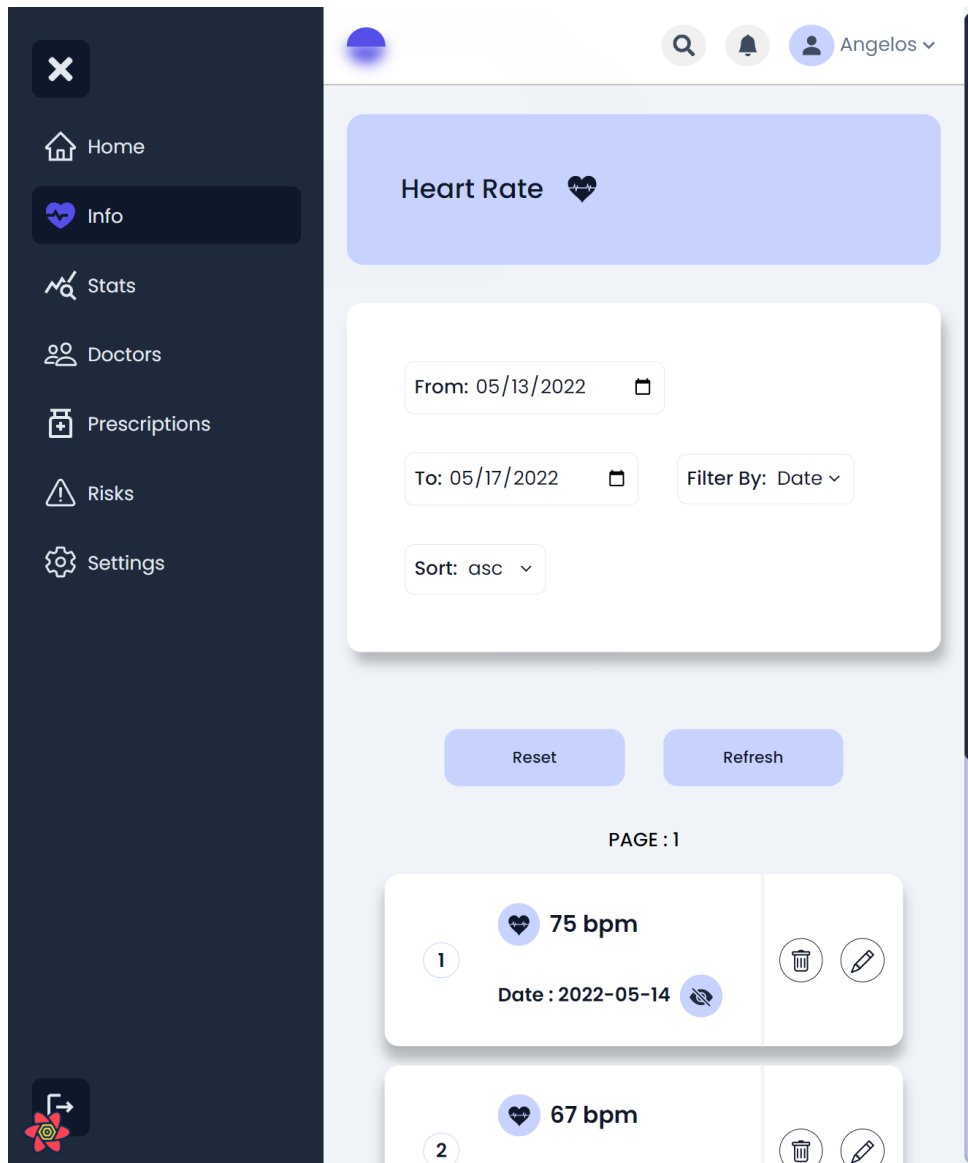


Εικόνα 5.12 - Σελίδα με τις πρόσφατες μετρήσεις

Το πρώτο άσπρο πλαίσιο, όπως φαίνεται, αφορά τις τιμές των καρδιακών παλμών. Για το συγκεκριμένο παράδειγμα έχουμε εισάγει τιμές, έτσι ώστε να φαίνεται η δομή των πληροφοριών. Στο πάνω μέρος υπάρχει η τιμή και στο κάτω μέρος αναγράφεται η ημερομηνία. Το δεύτερο άσπρο πλαίσιο, περίπου στην μέση της σελίδας, αφορά τις τιμές της θερμοκρασίας και το τελευταίο άσπρο πλαίσιο είναι για τις τιμές της πίεσης. Στην συγκεκριμένη κατηγορία μετρήσεων δεν υπάρχουν τιμές και για αυτή την περίπτωση αποφασίσαμε να αναγράφουμε «No Data...». Οι ημερομηνίες των παραπάνω παραδειγμάτων είναι τυχαίες. Όταν ο χρήστης θέλει να προσθέσει κάποια καινούρια τιμή έχει την δυνατότητα να βάλει ότι ημερομηνία θέλει, αν όμως στείλει μία τιμή από την συσκευή, η τιμή της ημερομηνίας θα είναι η ημερομηνία της μέρας που την ανέβασε.

5.2.4.2 Όλες οι μετρήσεις (See More)

Η παρακάτω σελίδα εμφανίζεται στην περίπτωση που ο χρήστης πατήσει «See More» στις τελευταίες μετρήσεις των καρδιακών παλμών, δηλαδή στο πρώτο άσπρο πλαίσιο.



Στιγμιότυπο 5.13 – Σελίδα με τις όλες τις μετρήσεις καρδιακών παλμών

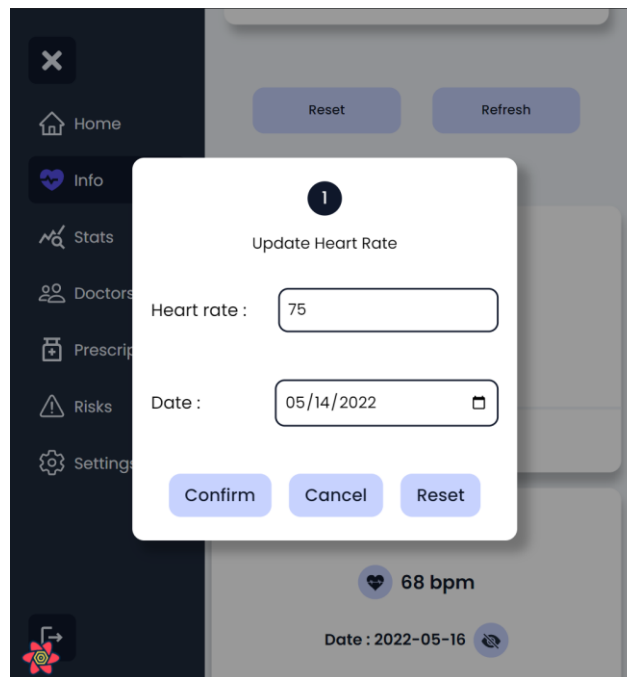
Το πρώτο πράγμα που αντικρίζει ο χρήστης είναι τα φίλτρα που μπορεί να χρησιμοποιήσει. Αυτά τα φίλτρα είναι:

- «From – To Date» : Δηλαδή «από – μέχρι» το οποίο αφορά το πλαίσιο των ημερομηνιών που θέλει να χρησιμοποιήσει για την αναζήτηση.
- «Filter By» : Οι επιλογές που δίνονται είναι Date και Rate
- «Sort» : Οι επιλογές που δίνονται είναι asc και desc (ascending και descending).

Για παράδειγμα αν ο χρήστης επιλέξει Filter By: Rate και Sort: desc, θα επιστραφούν αποτελέσματα τιμών στο χρονικό πλαίσιο που επιλέχθηκε και στα πρώτα αποτελέσματα θα εμφανίζονται μετρήσεις με τους μεγαλύτερους καρδιακούς παλμούς. Το τελευταίο αποτέλεσμα θα είναι ο μικρότερος καρδιακός παλμός. Κάτω από τα φίλτρα φαίνονται δύο κουμπιά. Το κουμπί «Reset» επαναφέρει τα

φίλτρα στις αρχικές τιμές και το κουμπί «Refresh» κάνει ανανέωση των αποτελεσμάτων. Ακριβώς κάτω από τα κουμπιά αναγράφεται ο αριθμός της σελίδας των αποτελεσμάτων. Ο λόγος που γίνεται αυτό είναι γιατί κάθε σελίδα περιέχει τέσσερα αποτελέσματα. Αυτό το κάναμε με σκοπό η σελίδα να είναι πιο απλή και οι απαντήσεις του server από τα αιτήματα να είναι μικρότερες, άρα και πιο γρήγορες. Γι αυτό τον λόγο δημιουργήσαμε και τα φίλτρα, έτσι ο χρήστης μπορεί να βρει εύκολα την πληροφορία που αναζητεί και δεν χρειάζεται να ψάχνει σε πίνακες με πάρα πολλά αποτελέσματα.

Σχετικά με το άσπρο πλαίσιο στο οποίο εμφανίζεται η μέτρηση, στα αριστερά βρίσκεται ο αριθμός του αποτελέσματος, στην μέση είναι η τιμή της μέτρησης και από κάτω βρίσκεται η ημερομηνία. Ο χρήστης έχει την δυνατότητα να πατήσει το κουμπί που βρίσκεται δίπλα από την ημερομηνία, το οποίο εμφανίζει πριν πόσο χρονικό διάστημα ανέβηκε η συγκεκριμένη μέτρηση. Για παράδειγμα θα εμφανίσει «2 days ago». Στην δεξιά πλευρά, η οποία χωρίζεται με μια γκρι γραμμή, υπάρχει κουμπί για την διαγραφή της μέτρησης το οποίο απεικονίζει ένα εικονίδιο κάδου και το κουμπί με το εικονίδιο του μολυβιού επιτρέπει στον χρήστη να αλλάξει τα στοιχεία του αποτελέσματος, δηλαδή στην συγκεκριμένη περίπτωση τους καρδιακούς παλμούς και την ημερομηνία. Αν ο χρήστης πατήσει το κουμπί αυτό τότε εμφανίζεται ξεχωριστό παράθυρο.

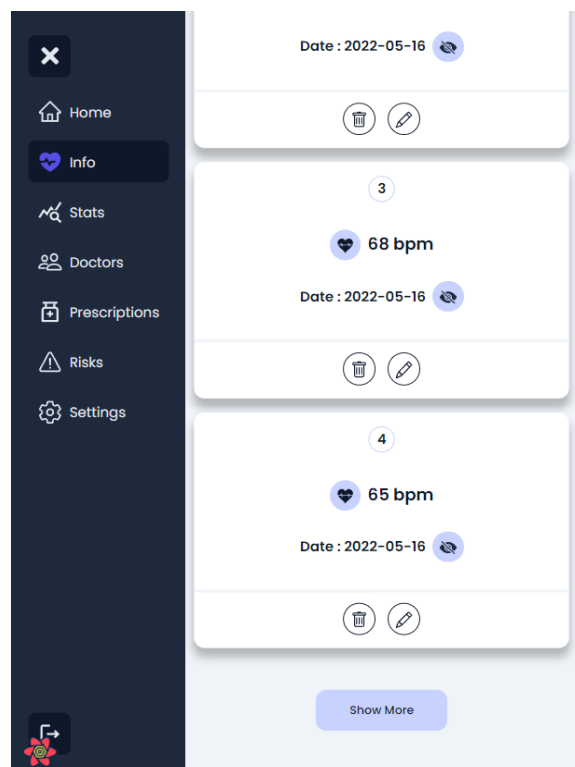


Εικόνα 5.13 - Αλλαγή στοιχείων μέτρησης

Σε αυτά τα πλαίσια μπορούν να αλλαχθούν οι τιμές. Αν ο χρήστης θέλει να επιβεβαιώσει την αλλαγή αυτή, πατάει το κουμπί «Confirm». Εάν θελήσει να κλείσει το παράθυρο χωρίς κάποια αλλαγή μπορεί να πατήσει το κουμπί «Cancel» ή να πατήσει οπουδήποτε στο μαύρο πλαίσιο πίσω από το παράθυρο. Τέλος το κουμπί «Reset» επαναφέρει τις αρχικές τιμές της μέτρησης. Βέβαια υπάρχει περιορισμός των τιμών που μπορεί να εισάγει ο χρήστης, διότι μια τιμή με 20 ή 320 παλμούς δεν θα έβγαζε νόημα. Έτσι αν ο χρήστης εισάγοντας για παράδειγμα «5» στο πλαίσιο Heart Rate και πατήσει «Confirm» στην ακριβώς κάτω σειρά από τα κουμπιά θα εμφανιστεί σφάλμα και έτσι δεν θα πραγματοποιηθεί η αλλαγή της τιμής.

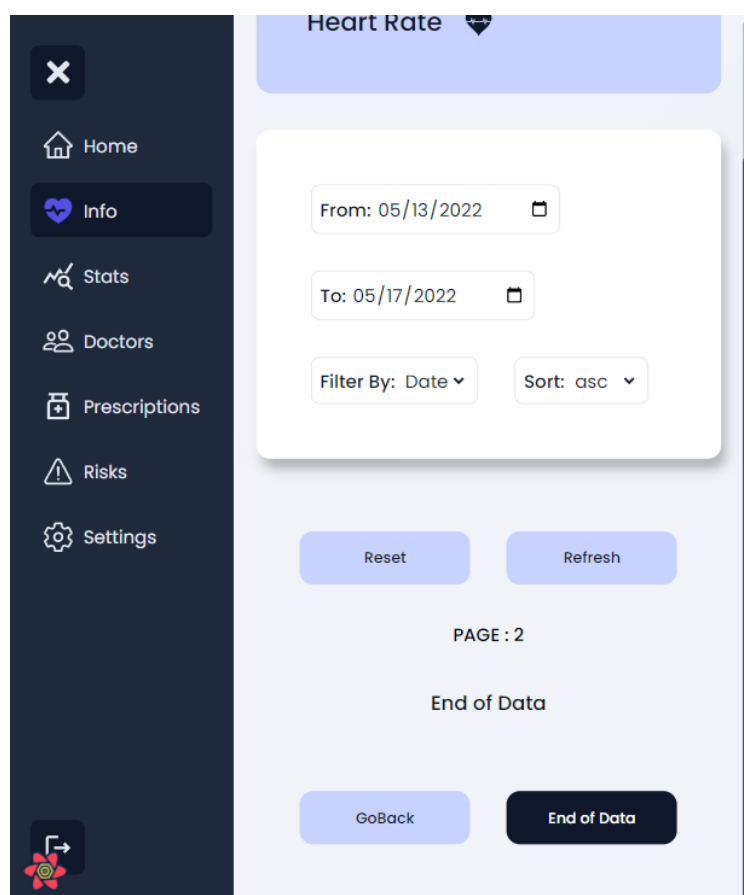
Εικόνα 5.14 - Σφάλμα αλλαγής στοιχείων μέτρησης

Τέλος κάτω απο την τελευταία μέτρηση της σελίδας εμφανίζεται κουμπί «Show More», έτσι ώστε να αλλάξει η σελίδα των αποτελεσμάτων.



Εικόνα 5.15 - Κουμπί για τις επόμενες μετρήσεις

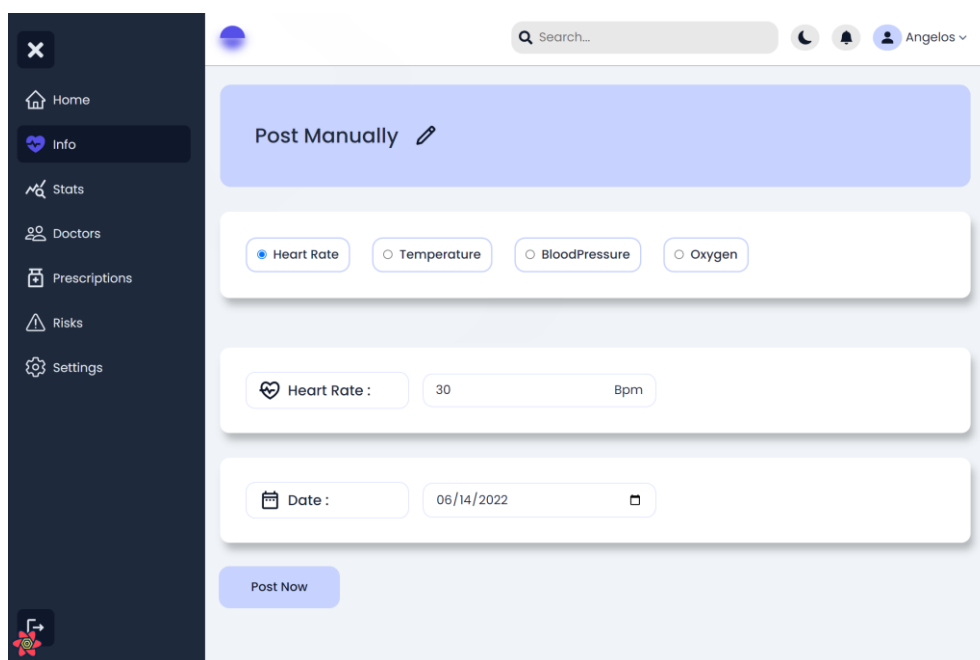
Στην περίπτωση που στην σελίδα που βρίσκεται ο χρήστης δεν εμφανιστούν τέσσερις μετρήσεις ή η επόμενη σελίδα είναι κενή, το κουμπί «Show More» δεν θα υπάρχει πλέον. Αντί αυτού εμφανίζεται ένα πλαίσιο που να φαίνεται σαν πατημένο κουμπί το οποίο αναγράφει «End of Data». Επίσης, αν ο χρήστης βρίσκεται σε μεγαλύτερη σελίδα από την πρώτη υπάρχει επιλογή να γυρίσει προς τα πίσω με το κουμπί «Go Back».



Εικόνα 5.16 - Τέλος μετρήσεων

5.2.4.3 Προσθήκη νέας μέτρησης (Post Manually)

Σχετικά με το κουμπί «Post Manually» στην σελίδα που αναγράφονται οι τελευταίες τρεις τιμές των μετρήσεων, οδηγεί τον χρήστη στην σελίδα Post.



Εικόνα 5.17 - Σελίδα προσθήκης νέων μετρήσεων

Στην σελίδα αυτή ο χρήστης ,όπως αναφέρθηκε, μπορεί να εισάγει νέες μετρήσεις.Στο πρώτο άσπρο πλαίσιο μπορεί να επιλέξει τι είδους μέτρησης θέλει .Στη συγκεκριμένη κατηγορία το δεύτερο άσπρο πλαίσιο αφορά την τιμή των καρδιακών παλμών.Οι ίδιοι περιορισμοί που ισχύουν και στην αλλαγή δεδομένων ισχύουν και στην δημιουργία νέων.Δηλαδή η τιμή των καρδιακών παλμών δεν μπορεί να είναι μικρότερη από 30.Το τελευταίο πλαίσιο είναι η ημερομηνία της μέτρησης.Ο χρήστης έχει την ελευθερία να βάλει όποια ημερομηνία θέλει.Τέλος, υπάρχει κουμπί «Post now» για να ανέβει η μέτρηση.Αν η τιμή ανέβει χωρίς πρόβλημα τότε το κουμπί «Post Now» γίνεται «Post Again».

Εικόνα 5.18 - Επιτυχή προσθήκη μέτρησης

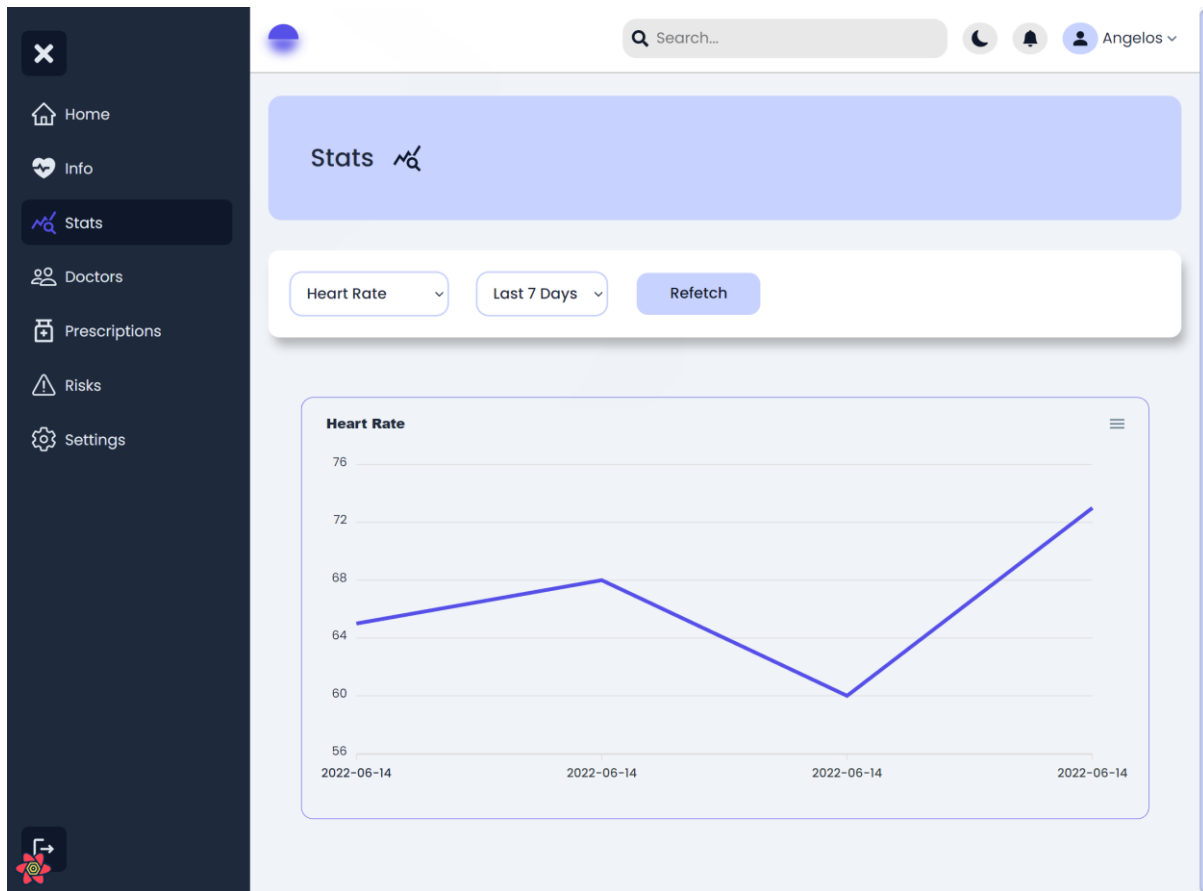
Με τον ίδιο τρόπο μπορούμε να προσθέσουμε θερμοκρασία και πίεση:

Εικόνα 5.19 - Σελίδες προσθήκης νέων μέτρησεων

5.2.5 Σελίδα γραφημάτων

Η επόμενη υποσελίδα που έχει το όνομα «Stats» στο πλάγιο αριστερό μενού μας εμφανίζει γράφημα με τις τιμές των μετρήσεων.Ο χρήστης μπορεί να επιλέξει ανάμεσα στους παλμούς , την θερμοκρασία ή την πίεση και να επιλέξει χρονικό διάστημα μιας ημέρας, 7 ημερών ή 30 ημερών.Στα γραφήματα στο Y άξονα αναγράφονται οι τιμές,δηλαδή οι παλμοί ή η θερμοκρασία ή διαστολική/συστολική πίεση, ενώ στον X άξονα οι ημερομηνίες.Ο χρήστης μπορεί να κάνει κλικ πάνω στην γραμμή του

γραφήματος και να δει ακριβώς τις τιμές που έχει το σημείο. Επιπλέον, δίνεται η δυνατότητα να κατεβάσει το γράφημα σαν εικόνα τύπου SVG ή PNG ή ακόμα να κάνει εξαγωγή των τιμών σε αρχείο CSV για ένα πρόγραμμα πινάκων όπως το Microsoft excel.



Εικόνα 5.20 - Σελίδα γραφημάτων

5.2.6 Σελίδα αλληπίδρασης με γιατρούς

Η υποσελίδα «Doctors» αφορά την επικοινωνία του γιατρού και του ασθενή. Σε αυτήν την υποσελίδα συμπεριλαμβάνονται τέσσερις καρτέλες.

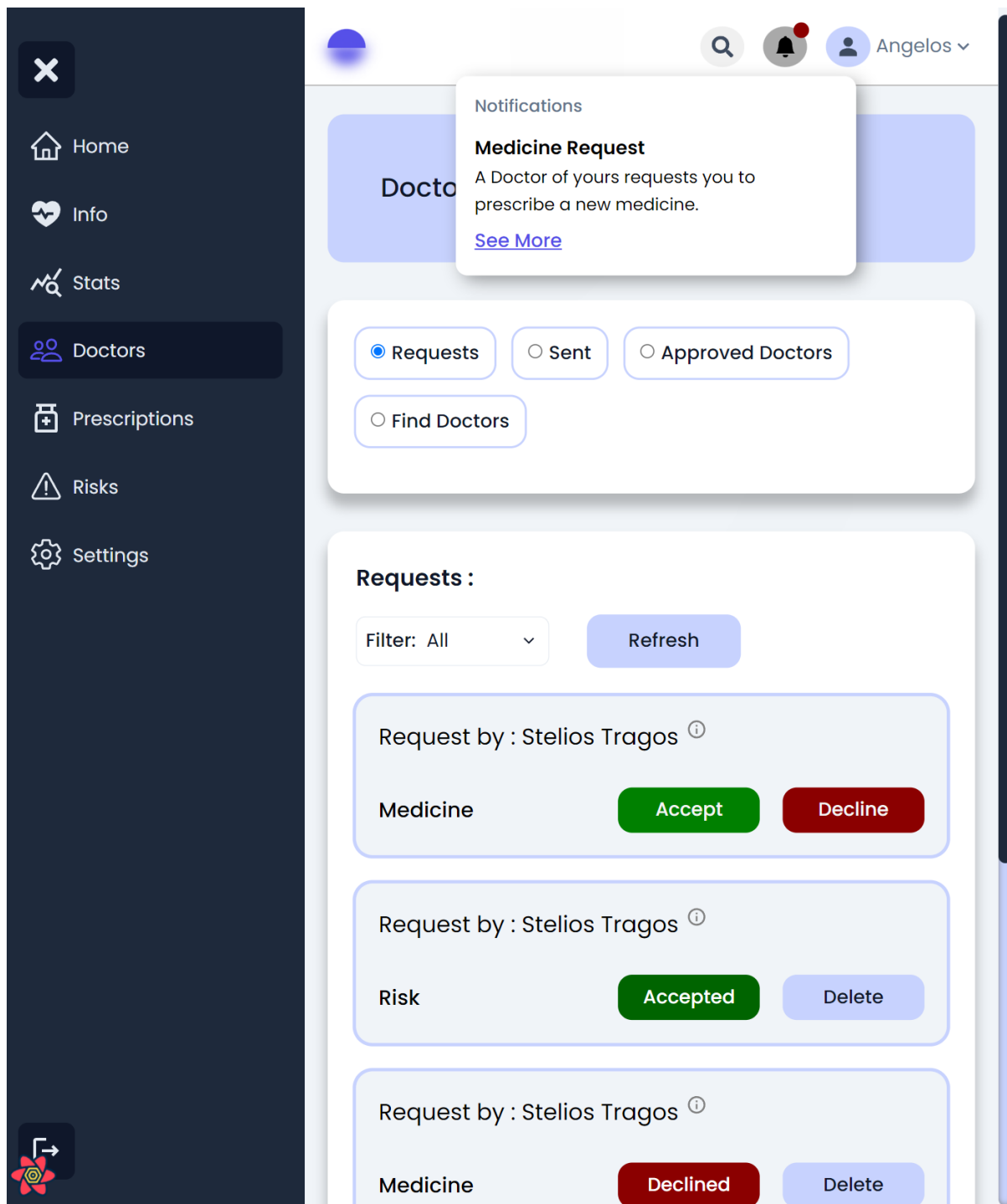
- Requests – Αιτήματα που έχει ο ασθενής από γιατρούς
- Sent – Τα αιτήματα που έχει στείλει ο ασθενής.
- Approved Doctors – Οι γιατροί που έχει εγκρίνει ο ασθενής
- Find Doctors – Αναζήτηση γιατρών

Η πρώτη καρτέλα που είναι και η προεπιλεγμένη έχει τα αιτήματα που δέχετε ο ασθενής από τους γιατρούς. Κάθε φορά που δέχεται ένα αίτημα, δημιουργείται και μια ειδοποίηση που φαίνεται στο κουμπί με το καμπανάκι. Αν ο χρήστης κάνει κλικ μία ειδοποίηση θα μεταβεί σε αυτή την καρτέλα. Αλλιώς μπορεί απλά να πατήσει το «Doctors» στο πλάγιο μενού. Ο χρήστης έχει την επιλογή να χρησιμοποιήσει φίλτρο για τα αιτήματα που έχει λάβει αλλά και για τα αιτήματα που έχει στείλει δηλαδή στις καρτέλες Requests και Sent. Το φίλτρο αυτό μπορεί να πάρει τέσσερις τιμές.

1. Unset – Τα αιτήματα που εκκρεμούν
2. Declined – Τα αιτήματα που έχουν απορριφθεί
3. Accepted – Τα αιτήματα που έχουν γίνει αποδεκτά

4. All – Όλα τα αιτήματα

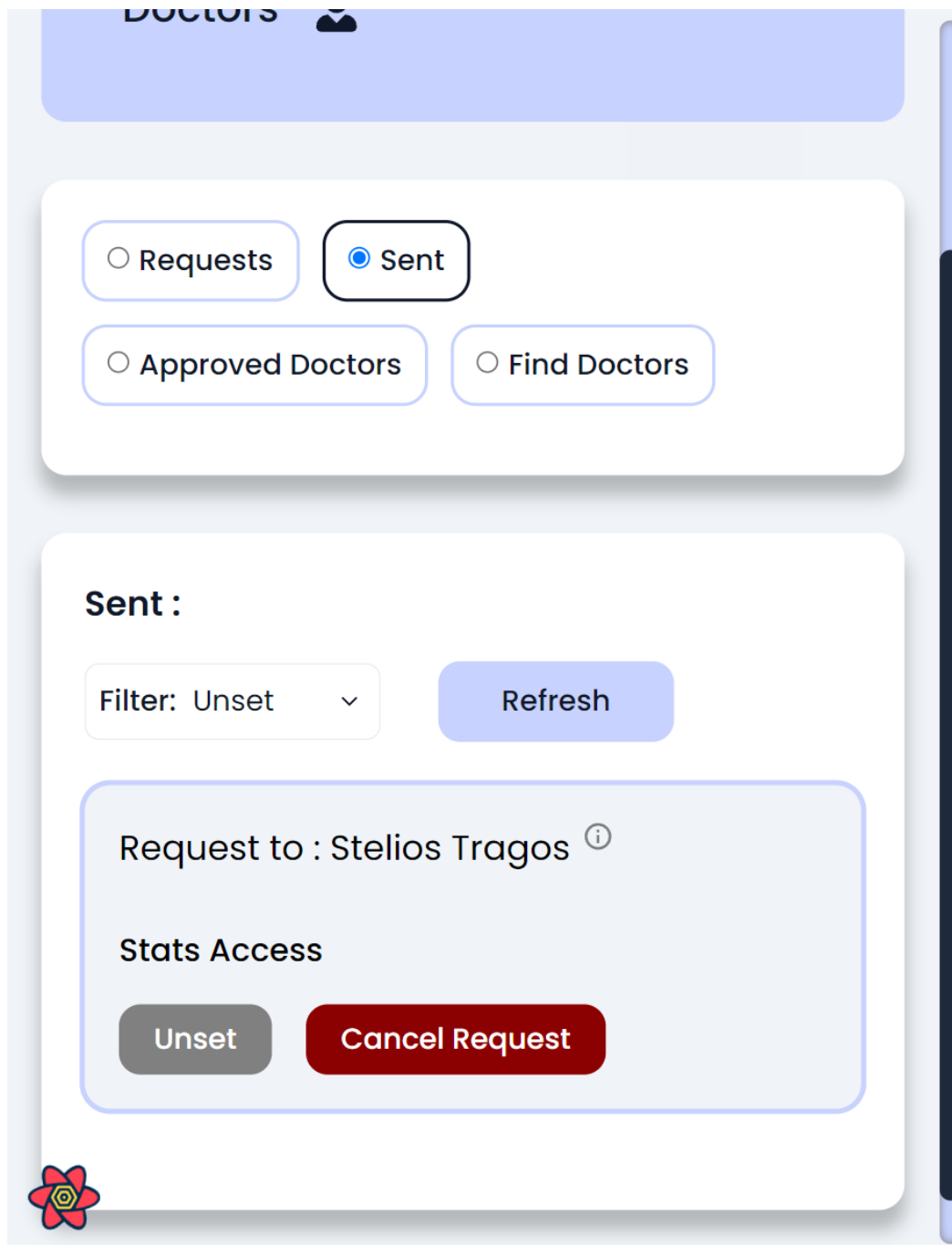
Ανάλογα με το φίλτρο που χρησιμοποιείται εμφανίζονται και τα αντίστοιχα αιτήματα.



Εικόνα 5.21 - Σελίδα αιτημάτων

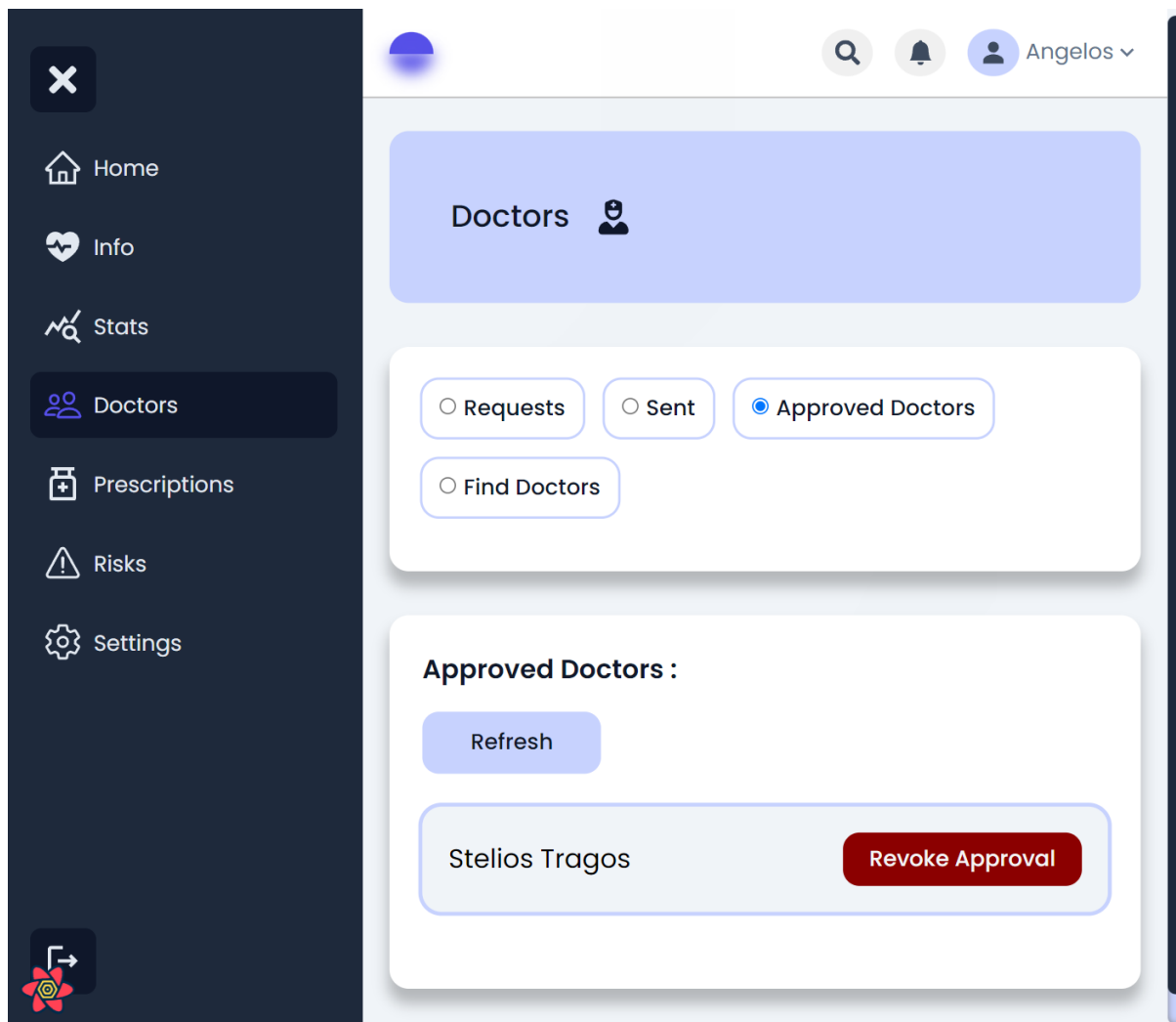
Στην παραπάνω φωτογραφία είναι επιλεγμένο το φίλτρο «All», άρα φαίνονται τα αιτήματα που έχει απαντήσει ο χρήστης, αλλά και τα αιτήματα που εκκρεμούν. Στην συγκεκριμένη περίπτωση το πρώτο αίτημα εκκρεμεί και υπάρχουν δύο επιλογές, «Accept» για την αποδοχή και «Decline» για την απόρριψη. Επίσης, τα αιτήματα που έχουν ήδη απαντηθεί μπορούν να διαγραφούν με το κουμπί «Delete». Ακόμη, τα αιτήματα που έχει στείλει ο χρήστης φαίνονται στην καρτέλα «Sent» και

βρίσκονται στην κατάσταση «Unset» που σημαίνει ότι δεν έχουν απορριφθεί ή εγκριθεί ακόμη και μπορούν να ακυρωθούν με το κουμπί «Cancel Request».



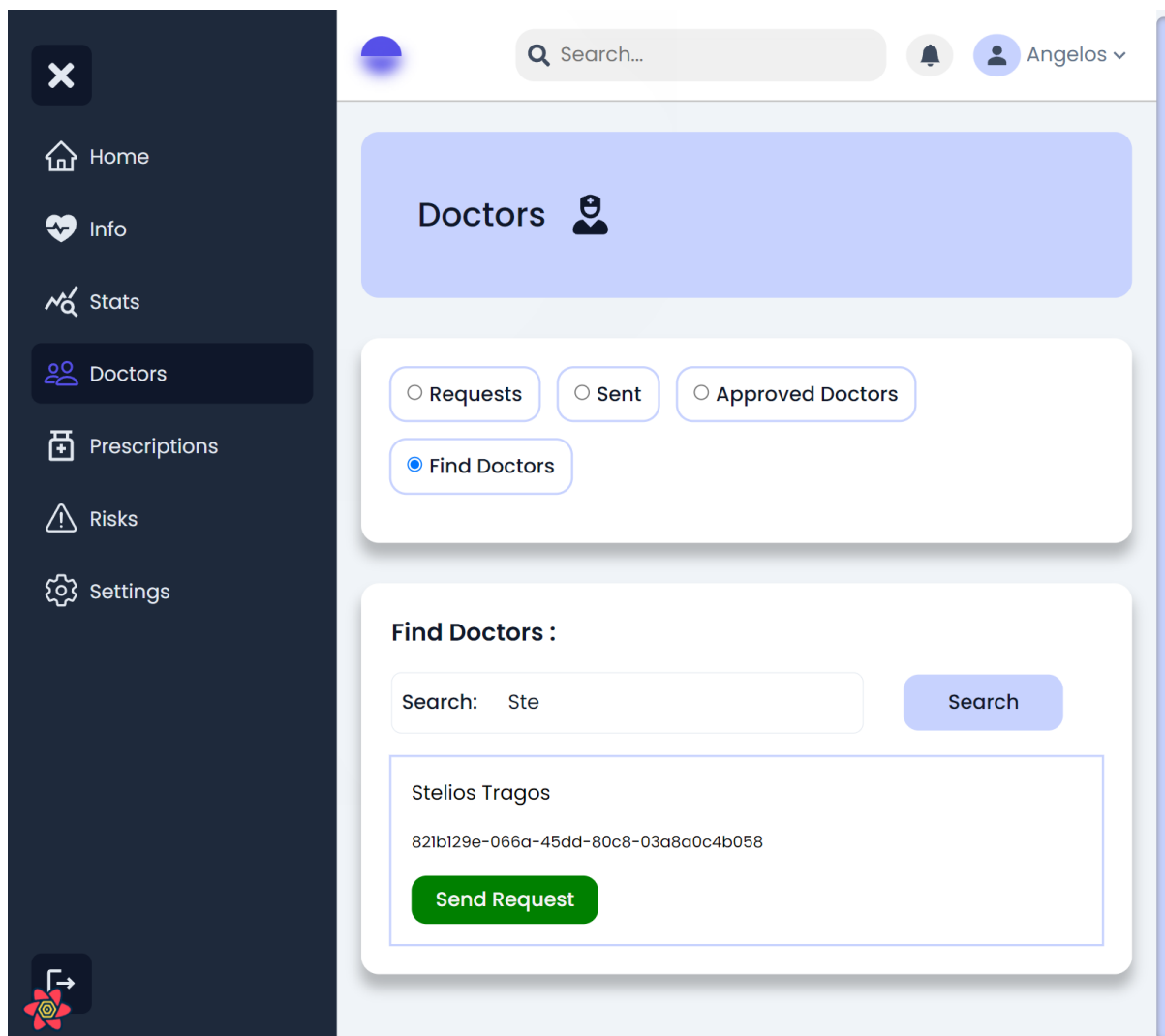
Εικόνα 5.22 - Σελίδα αιτημάτων που έχουν σταλθεί

Στην καρτέλα «Approved Doctors», όπως αναφέρθηκε, φαίνονται οι γιατροί που έχει εγκρίνει ο ασθενής. Σε αυτή την καρτέλα, ο χρήστης έχει την δυνατότητα να ανακαλέσει την έγκριση του με το κουμπί «Revoke Approval». Η ανάκληση της έγκρισης σημαίνει ότι πλέον ο γιατρός δεν θα μπορεί να στείλει αιτήματα για νέα φάρμακα, να γράψει νέα ρίσκα ή να δει τα βιομετρικά του ασθενή αυτού.

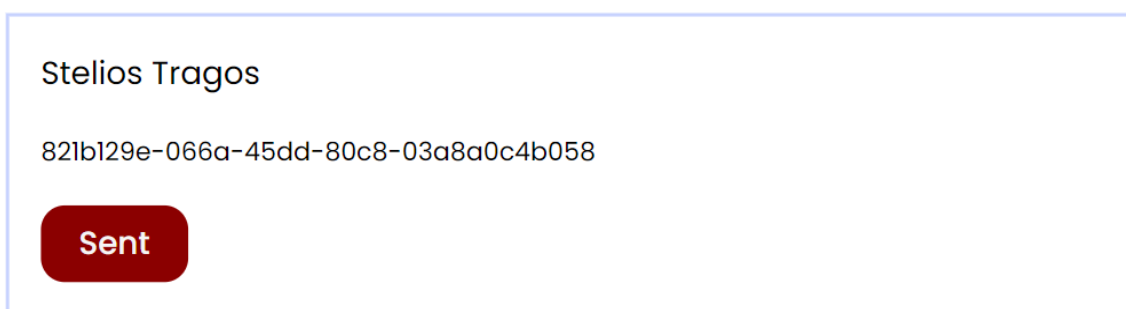


Εικόνα 5.23 - Σελίδα εγκεκριμένων γιατρών

Τέλος, η καρτέλα «Find Doctor» είναι για την αναζήτηση γιατρών. Η αναζήτηση μπορεί να γίνει με όνομα, επίθετο ή ακόμη και το μοναδικό ID. Αν έχει σταλθεί ήδη αίτημα σε έναν γιατρό δεν μπορεί να ξανασταλθεί και ο γιατρός αυτός δεν θα εμφανιστεί στην αναζήτηση αυτή. Ο ασθενής μπορεί να παρακολουθήσει την κατάσταση του αιτήματος στην καρτέλα «Sent». Αν όμως ο ασθενής ή ο γιατρός κάνει «Revoke Approval», δηλαδή ανάκληση της έγκρισης, ο γιατρός θα εμφανιστεί στην αναζήτηση και ο ασθενής θα μπορεί να του ξαναστείλει αίτημα για έγκριση. Το σύστημα λειτουργεί όπως ακριβώς λειτουργούν τα αιτήματα φιλίας στα μέσα κοινωνικής δικτύωσης. Ο χρήστης απλά γράφει αυτό που θέλει στην αναζήτηση και πατάει το κουμπί «Search» που βρίσκεται ακριβώς δίπλα ή πατάει το κουμπί «Enter» στο πληκτρολόγιο. Τα αποτελέσματα θα εμφανιστούν από κάτω. Σε κάθε αποτέλεσμα πρώτα αναγράφεται το όνομα του γιατρού, από κάτω το μοναδικό ID και τέλος από κάτω υπάρχει κουμπί για να σταλθεί το αίτημα και ονομάζεται «Send Request». Αν το κουμπί χρησιμοποιηθεί, από πράσινο θα γίνει κόκκινο και θα εμφανίσει «Sent» αντί για «Send Request», αν σταλθεί το αίτημα με επιτυχία, διαφορετικά θα γράψει «Error».



Εικόνα 5.24 - Σελίδα εύρεσης νέων γιατρών

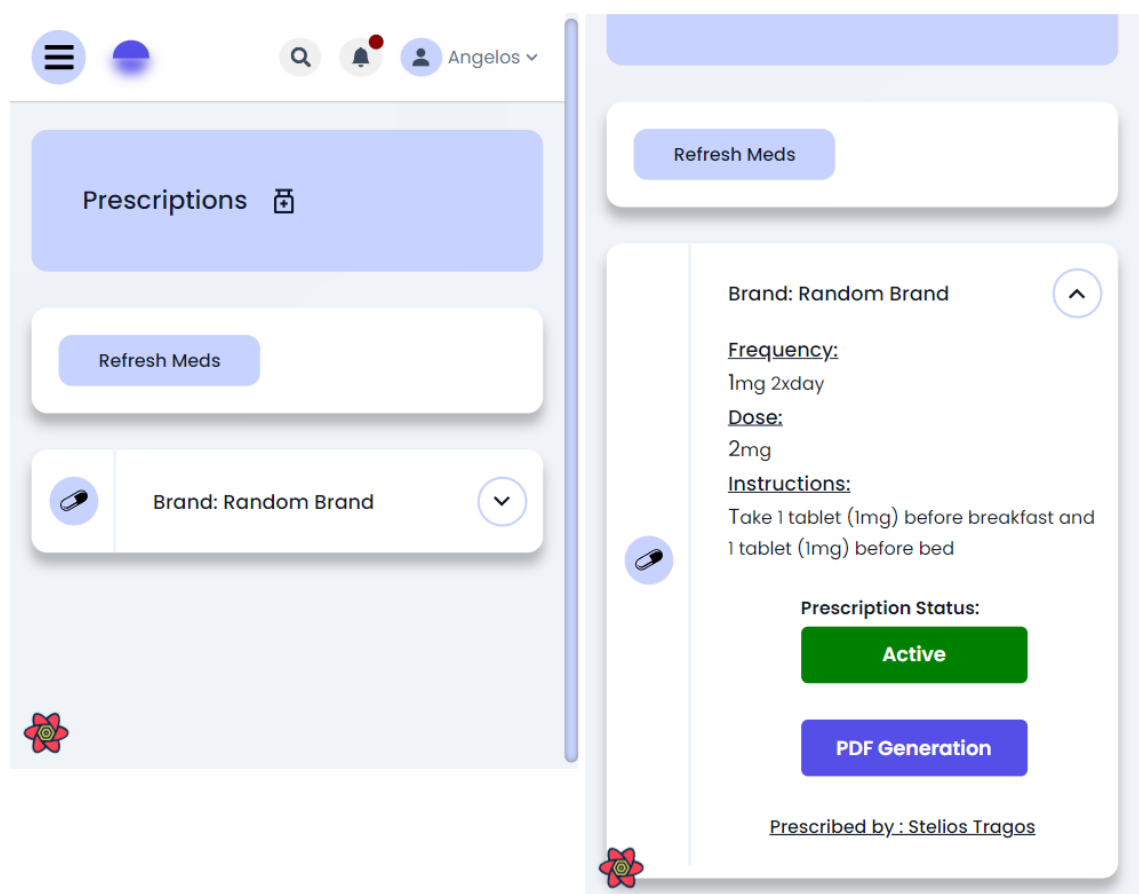


Εικόνα 5.25 - Επιτυχή αποστολή αιτήματος

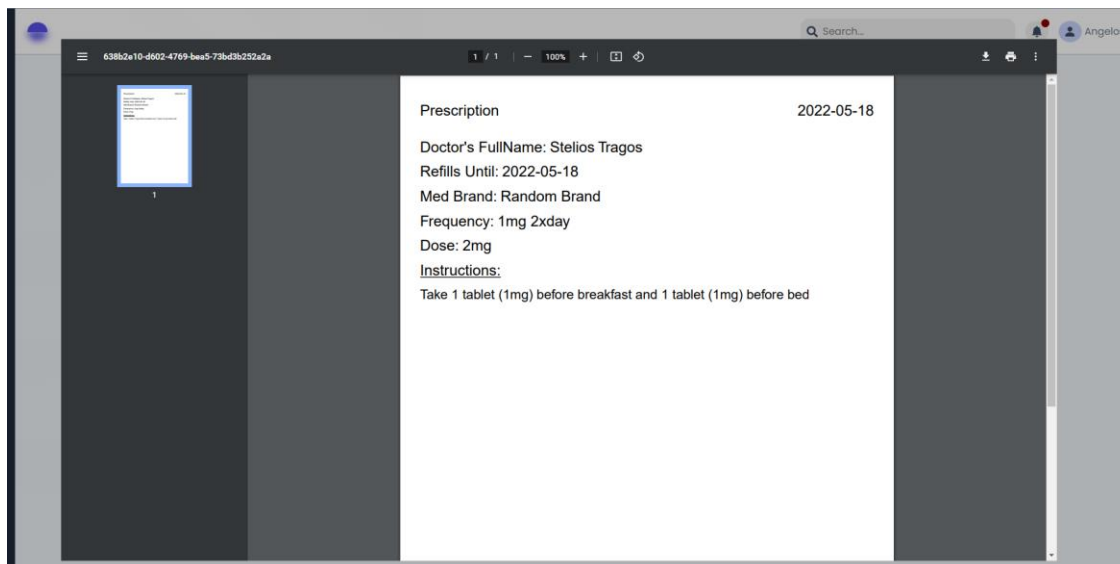
5.2.7 Σελίδα φαρμάκων

Η υποσελίδα Prescriptions είναι τα φάρμακα που έχουν συνταγογραφηθεί στον ασθενή από τους γιατρούς που έχει δώσει έγκριση. Σε αυτήν ο χρήστης μπορεί να δει πληροφορίες σχετικά με κάθε συνταγογράφηση, όπως την συχνότητα, την δοσολογία των φαρμάκων και γενικότερες κατευθυντήριες οδηγίες που έχει συμπεριλάβει ο γιατρός (Frequency, Dose, Instructions). Επιπλέον, αναγράφεται η κατάσταση της κάθε συνταγογράφησης, δηλαδή αν είναι ακόμη ενεργή ή αν έχει

λήξει. Τέλος, υπάρχει κουμπί για την δημιουργία αρχείου PDF με τις σχετικές πληροφορίες και βέβαια το όνομα του γιατρού που έχει κάνει την συνταγογράφηση. Έχουμε προσθέσει και επιπλέον λειτουργίες όπως κουμπί για ανανέωση των φαρμάκων σε περίπτωση που ο ασθενής έχει αποδεχτεί αίτημα νέου φαρμάκου και δεν έχει γίνει ανανέωση της λίστας αυτόματα, λειτουργία στην οποία αν ο χρήστης κάνει κλικ το όνομα του γιατρού, τον μεταφέρει στην σελίδα που περιέχει τους γιατρούς που έχει δώσει έγκριση (Approved Doctors). Στην αρχή δεν φαίνονται όλες οι πληροφορίες για το φάρμακο, αλλά μόνο το όνομα του (Brand). Αν ο χρήστης κάνει κλικ ένα κουμπί το οποίο ουσιαστικά είναι ένα βελάκι θα εμφανιστούν όλες οι πληροφορίες σχετικά με συγκεκριμένο φάρμακο. Αυτό έχει γίνει σε περίπτωση που ασθενής έχει αρκετά φάρμακα και απλά θέλει να δει ποιά είναι εύκολα, χωρίς να εμφανίζονται όλες οι πληροφορίες αυτόματα και καταλαμβάνουν πολύ χώρο στην οθόνη.



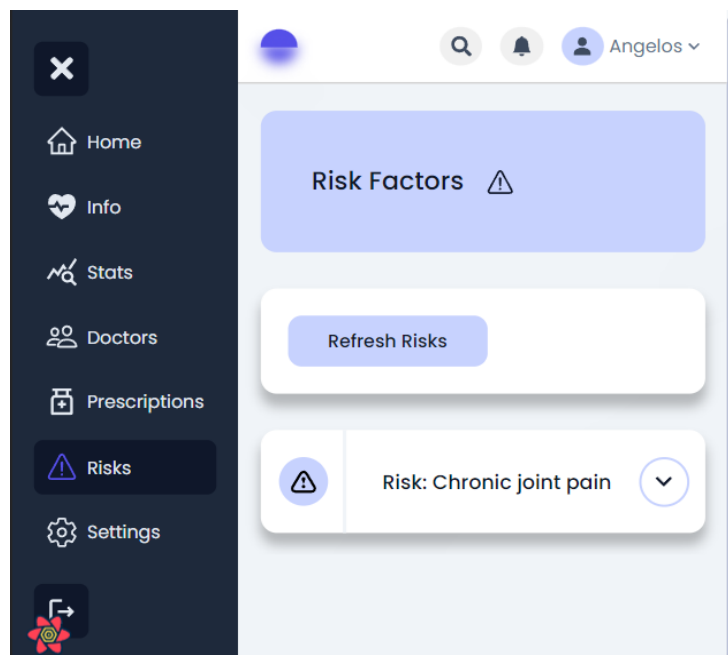
Εικόνα 5.26 - Σελίδα φαρμάκων



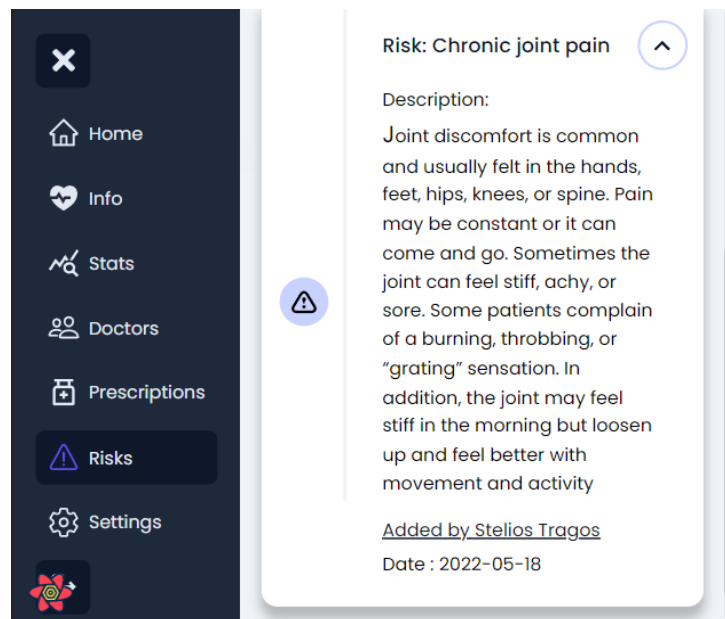
Εικόνα 5.27 - Δημιουργία αρχείου PDF

5.2.8 Σελίδα ρίσκων

Η επόμενη υποσελίδα είναι τα Risks(ρίσκα).Στα Risks ο γιατρός μπορεί να γράψει διάφορα πράγματα για τον ασθενή όπως συμπληρωματικές πληροφορίες που κρίνει σημαντικές, για παράδειγμα κάποια χρόνια πάθηση,πληροφορίες γι αυτή (συμπτώματα και τρόπους για ανακούφιση).Η συγκεκριμένη υποσελίδα μοιάζει με την Prescription, δηλαδή υπάρχει το ίδιο βελάκι για να εμφανιστούν όλες οι πληροφορίες, αλλιώς φαίνεται μόνο ο τίτλος και το κουμπί για την ανανέωση των πληροφοριών .Οι πληροφορίες που υπάρχουν είναι ο τίτλος, η περιγραφή, το όνομα του γιατρού που τις συνέταξε και η ημερομηνία.Όπως τα φάρμακα έτσι και τα ρίσκα είναι αιτήματα που πρέπει να αποδεχτεί ο ασθενής.



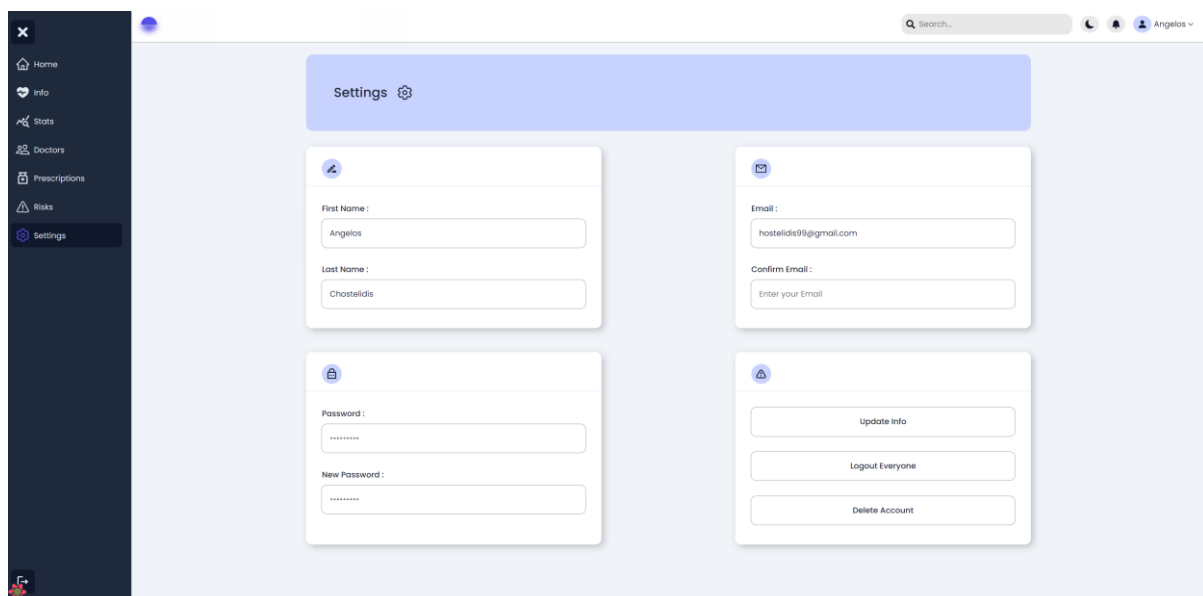
Εικόνα 5.28 - Σελίδα ρίσκων



Εικόνα 5.29 – Πληροφορίες ρίσκων

5.2.9 Σελίδα ρυθμίσεων

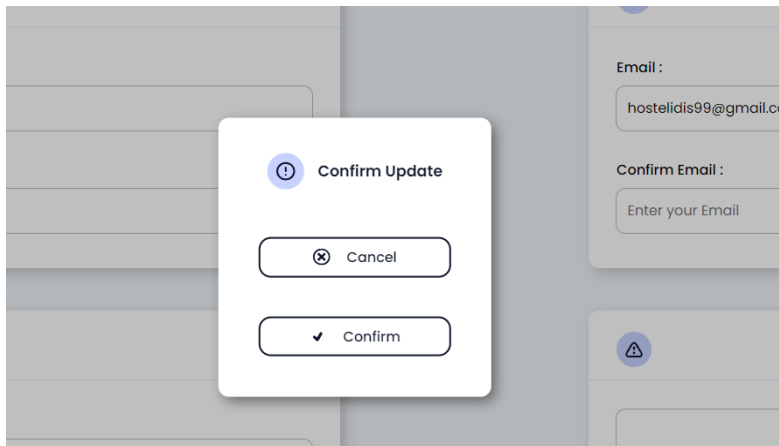
Η τελευταία υποσελίδα που βρίσκεται στο σκούρο μπλε μενού είναι τα Settings στα οποία ο χρήστης μπορεί να πάει είτε από αυτή είτε από το υποπλάισιο που εμφανίζεται αν πατήσει το όνομα του στο άσπρο μενού τέρμα πάνω δεξιά της σελίδας. Σε αυτή την υποσελίδα ο χρήστης μπορεί να δει και αλλάξει το ονοματεπώνυμο του, το email του και τον κωδικό πρόσβασης του. Επιπλέον δίνεται λειτουργία να αποσυνδέσει όλα τα άτομα που μπορεί να είναι συνδεδεμένα στον λογαριασμό του (Logout Everyone) και τέλος λειτουργία για την διαγραφή του λογαριασμού του.



Εικόνα 5.30 - Σελίδα ρυθμίσεων

Αν ο χρήστης πατήσει ένα από αυτά τα κουμπία επειδή οι λειτουργίες τους είναι σημαντικές εμφανίζεται ένα νέο μικρό παράθυρο στο οποίο ο χρήστης πρέπει να επιβεβαιώσει ή να απορρίψει τις ενέργειές του. Για παράδειγμα, αν ο χρήστης θέλει να αλλάξει το όνομα του και πατήσει «Update

Info» θα εμφανιστεί παράθυρο «Confirm Update» με δύο κουμπιά, το ένα είναι «Confirm» και το άλλο «Cancel».



Εικόνα 5.31 - Ερώτημα αλλαγής ρυθμίσεων

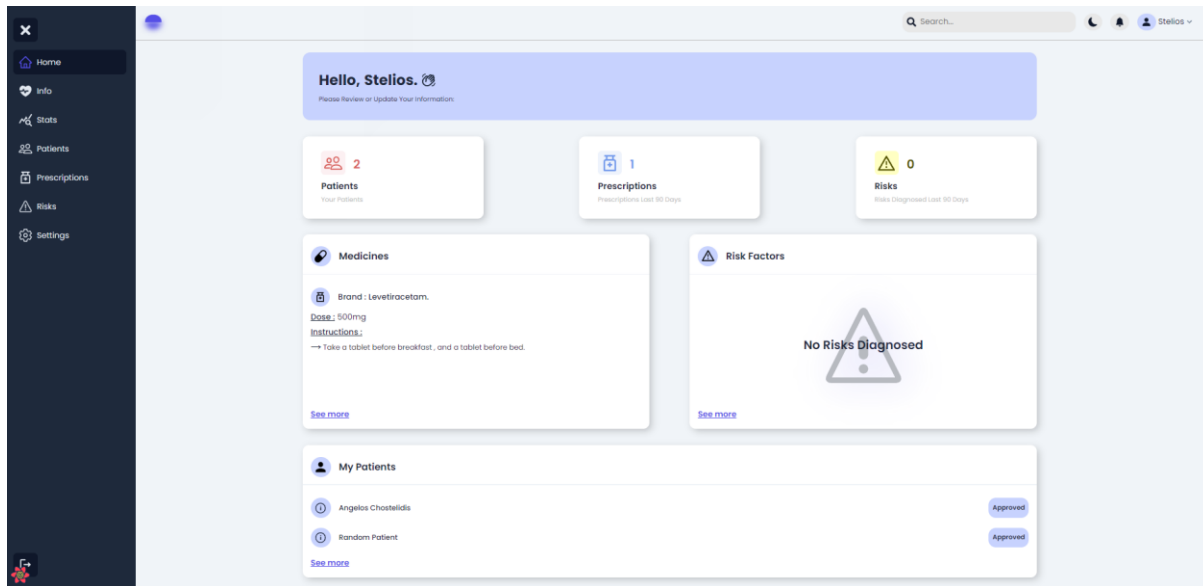
Ακόμη, μπορούν να εμφανιστούν σφάλματα. Αν για παράδειγμα ο χρήστης πάει να αλλάξει το email του και στο πλαίσιο επιβεβαίωσης δεν το πληκτρολογήσει σωστά ή αν προσπαθήσει να αλλάξει το κωδικό πρόσβασης και ο κωδικός δεν είναι ο σωστός.

5.3 Η εφαρμογή από πλευρά Γιατρού

5.3.1 Αρχική σελίδα

Εάν ο χρήστης συνδεθεί στην εφαρμογή και έχει τον ρόλο του γιατρού η εφαρμογή είναι διαφορετική, διότι ο γιατρός μπορεί να κάνει ενέργειες που δεν μπορεί να κάνει ο ασθενής. Για παράδειγμα, μπορεί να συνταγογραφήσει φάρμακα ή να γράψει πληροφορίες για άλλους χρήστες. Η εφαρμογή όμως έχει την ίδια δομή, τα ίδια χρώματα, τα ίδια υποπλάισια και γενικότερα το ύφος της εφαρμογής είναι παρόμοιο.

Η αρχική σελίδα της εφαρμογής αποτελείται, όπως και του ασθενή, από υποπλάισια που αφορούν τα φάρμακα, τα ρίσκα. Η μόνη διαφορά που υπάρχει είναι τα πρώτα τρία υποπλάισια που αφορούν το νούμερο των ασθενών που έχουν εγκριθεί, το νούμερο των φαρμάκων που έχουν συνταγογραφηθεί τις τελευταίες 30 ημέρες και τα νούμερα των ρίσκων που έχουν γραφτεί και αποδεχτεί τις τελευταίες 30 ημέρες, ενώ στην εφαρμογή από την πλευρά ασθενή υπάρχουν διάφορα βιομετρικά.

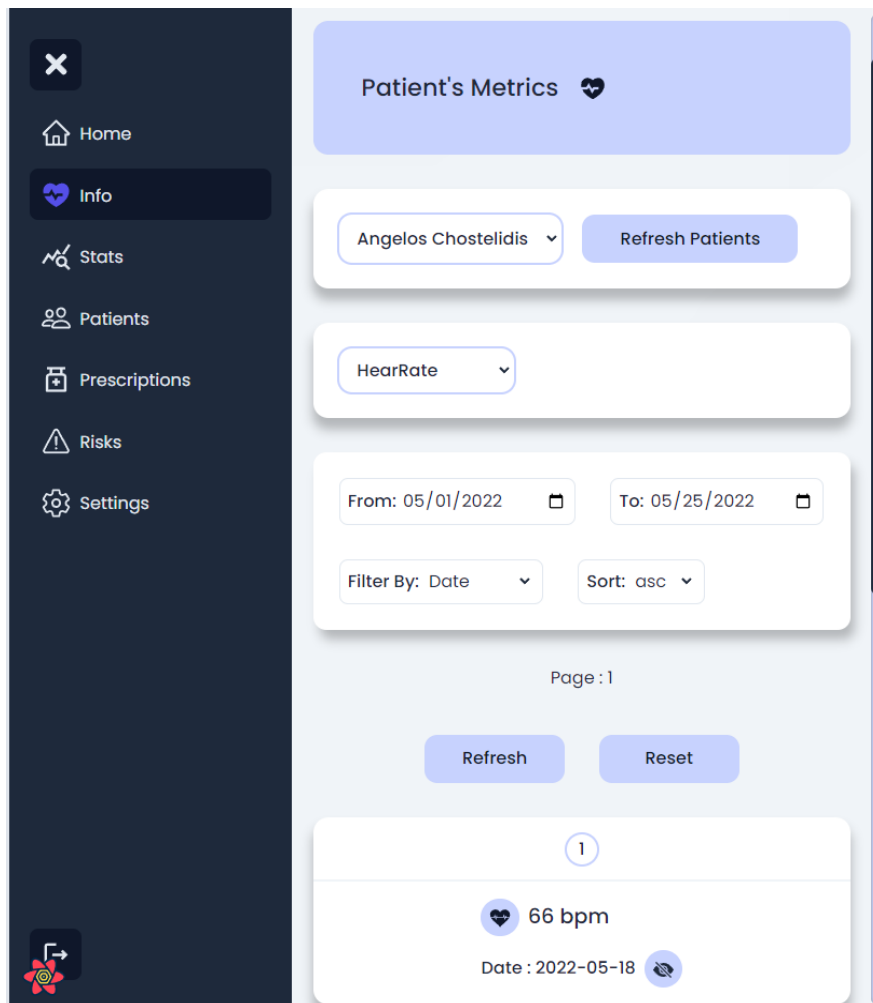


Εικόνα 5.32 - Αρχική σελίδα ιατρού

Το σκούρο μπλε μενού στα αριστερά είναι ακριβώς το ίδιο, αλλά το περιεχόμενο των υποσελίδων είναι διαφορετικό. Το άσπρο μενού στην κορυφή κάθε υποσελίδας είναι ακριβώς το ίδιο με αυτό του ασθενή. Για την ακρίβεια, είναι ακριβώς το ίδιο κομμάτι κώδικα. Στο υποπλαίσιο των φαρμάκων εμφανίζονται τα δύο τελευταία φάρμακα που έχει γράψει ο γιατρός και στο υποπλαίσιο των ρίσκων τα δύο τελευταία ρίσκα που έχει γράψει ο γιατρός. Στο τελευταίο υποπλαίσιο εμφανίζονται οι τελευταίοι δύο ασθενείς που έχουν εγκριθεί. Αν ο χρήστης κάνει κλικ στο κουμπί «See More» θα κατευθυνθεί στην αντίστοιχη υποσελίδα που αφορά το πλαίσιο.

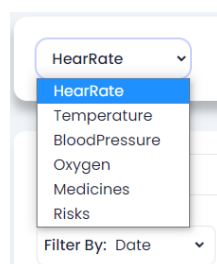
5.3.2 Σελίδα μετρήσεων ασθενών

Η επόμενη υποσελίδα ονομάζεται «Info» και αφορά τα στοιχεία των ασθενών. Ο γιατρός φυσικά έχει πρόσβαση σε αυτά εφόσον ο ασθενής δώσει έγκριση στο αίτημα «Stat Access». Ο γιατρός έχει την επιλογή να επιλέξει έναν από τους ασθενείς που έχει εγκρίνει και κατόπιν να δει τα βιομετρικά του, τα φάρμακα του και τα ρίσκα. Όμως, δεν έχει την δυνατότητα να δει από ποιους γιατρούς έχουν γραφτεί τα παραπάνω.



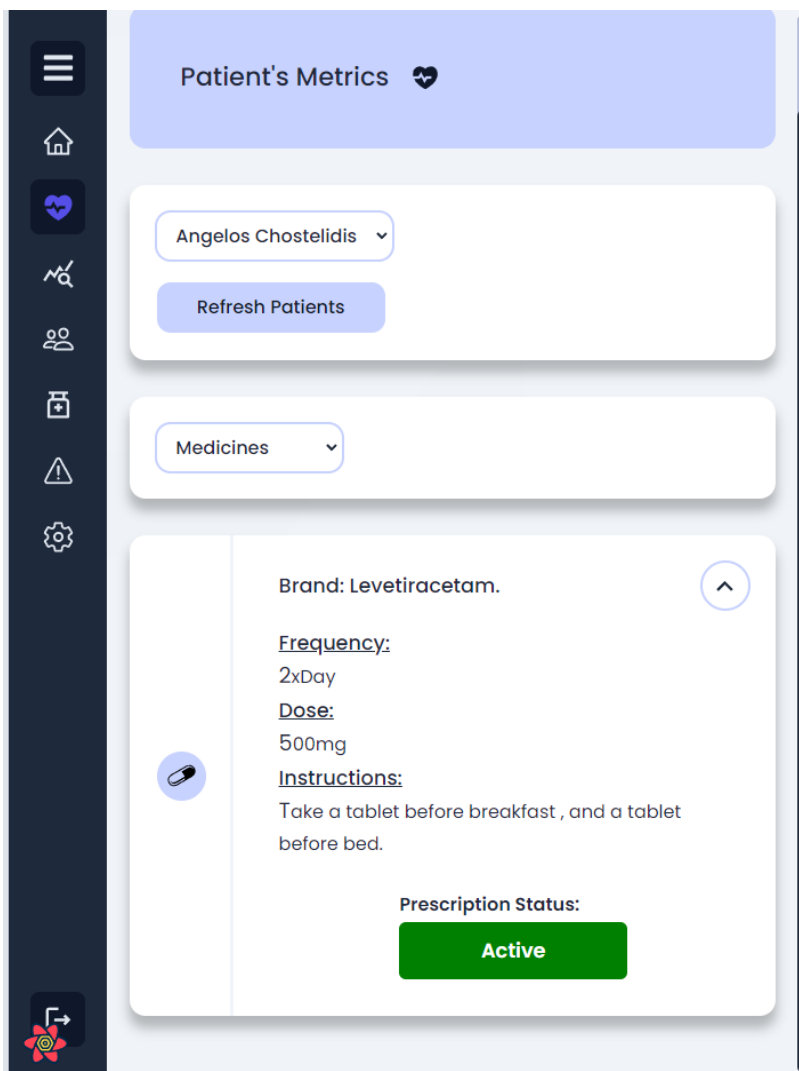
Εικόνα 5.33 - Σελίδα πληροφοριών ασθενών

Ο γιατρός στο πρώτο υποπλαίσιο επιλέγει τον ασθενή που θέλει. Στο δεύτερο υποπλαίσιο επιλέγει τις πληροφορίες που θέλει να δει, στο τρίτο υπάρχουν τα φίλτρα που αναφέρθηκαν παραπάνω, όπως και στην πλευρά του ασθενή. Δηλαδή, υπάρχει δυνατότητα ορισμού χρονικού πλαισίου και κατάταξης αύξουσας ή φθίνουσας σειράς ανάλογα με την ημερομηνία της μέτρησης. Τέλος, από κάτω εμφανίζονται τα αποτελέσματα. Υπάρχει ο αριθμός της σελίδας των αποτελεσμάτων, κουμπί για την ανανέωση, κουμπί για την επαναφορά των φίλτρων στις αρχικές τιμές και κουμπί για την επόμενη ή την προηγούμενη σελίδα. Κάθε σελίδα περιέχει μέχρι τέσσερις μετρήσεις. Η σελιδοποίηση έχει υλοποιηθεί μόνο για τις μετρήσεις και όχι για τα φάρμακα ή τα ρίσκα, διότι οι μετρήσεις μπορεί να είναι πάρα πολλές, ενώ τα φάρμακα και τα ρίσκα όχι.



Εικόνα 5.34 - Επιλογή κατηγορίας πληροφοριών

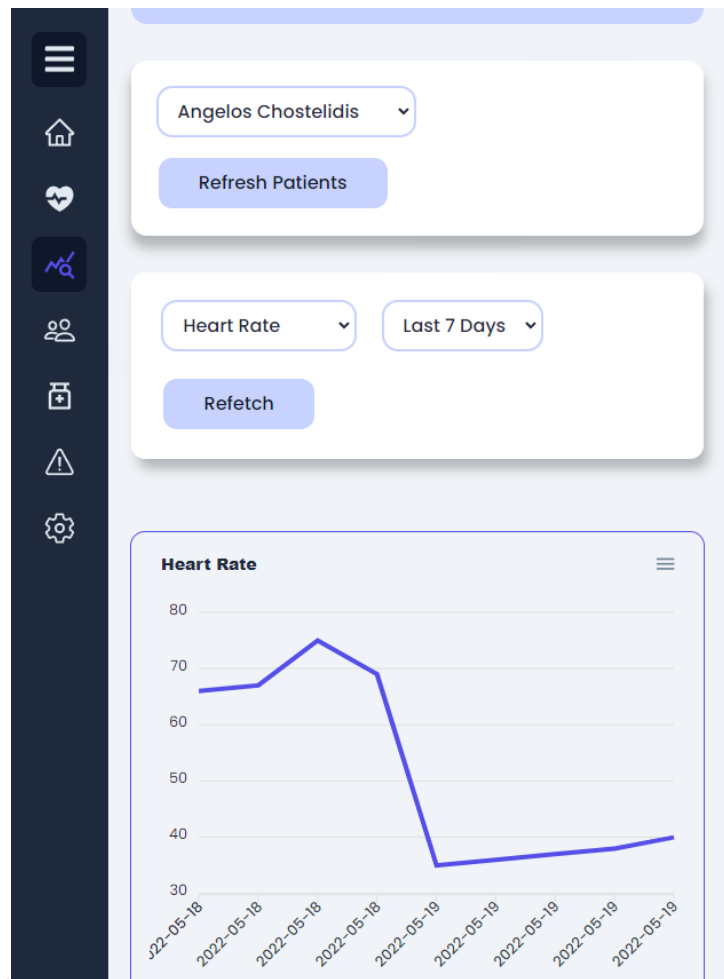
Σε αυτήν την υποσελίδα ο γιατρός δεν έχει την δυνατότητα να κάνει κάποια αλλαγή, αλλά μόνο μπορεί να δει πληροφορίες . Επομένως τα φάρμακα και τα ρίσκα εμφανίζονται όπως εμφανίζονται και στον ασθενή.



Εικόνα 5.35 - Φάρμακα ασθενή

5.3.3 Σελίδα γραφημάτων με μετρήσεις ασθενών

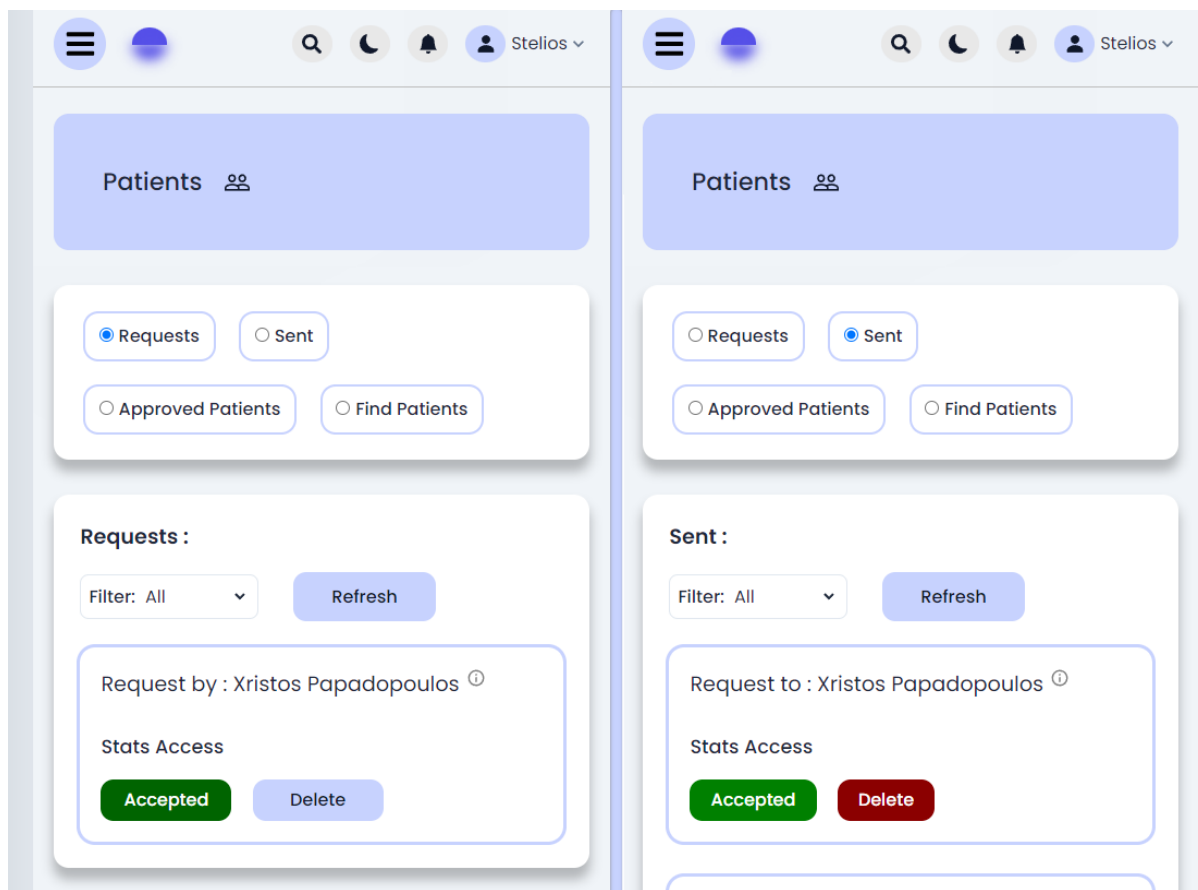
Η επόμενη υποσελίδα «Stats» είναι αυτή που δείχνει τα γραφήματα με βιομετρικά και τις ημερομηνίες. Η μόνη διαφορά με αυτή του ασθενή είναι ότι ο γιατρός έχει την επιλογή να επιλέξει ποιανού χρήστη τα βιομετρικά θέλει να δει. Υπάρχει κουμπί για την ανανέωση των ασθενών, κουμπί για την ανανέωση των αποτελεσμάτων, επιλογή για τις μετρήσεις που θέλει να δει (μετρήσεις καρδιακών παλμών , θερμοκρασίας κτλπ). Τέλος τα φίλτρα παραμένουν ίδια, δηλαδή μιας ημέρας, επτά ημερών και ενός μήνα. Ακόμη, όπως και στον ασθενή, δίνεται η δυνατότητα στον γιατρό να κατεβάσει το γράφημα σαν εικόνα τύπου SVG ή PNG, όπως επίσης να κάνει εξαγωγή των τιμών σε αρχείο CSV για ένα πρόγραμμα πινάκων όπως το Microsoft excel.



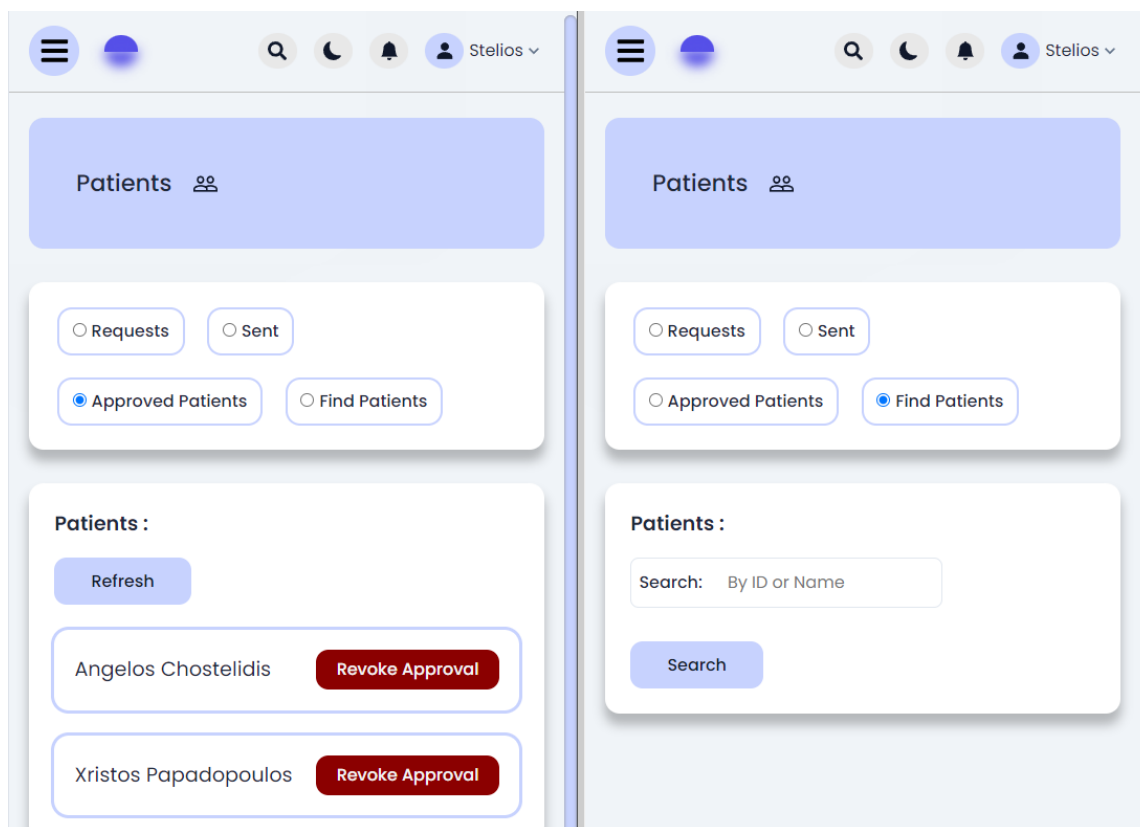
Εικόνα 5.36 - Σελίδα γραφημάτων

5.3.4 Σελίδα αλληλεπίδρασης με ασθενείς

Η υποσελίδα «Patients» είναι η αντίστοιχη της «Doctors» του ασθενή. Όπως αναφέρθηκε, περιέχει τα αιτήματα (requests & sent) τα οποία στέλνονται από ασθενείς. Ο γιατρός τα αποδέχεται ή τα απορρίπτει. Στην καρτέλα sent είναι τα αιτήματα που έχει στείλει ο γιατρός, και όπως ο ασθενής, μπορεί να δει την κατάσταση τους ή να τα ακύρωσει πριν απαντήσει ο χρήστης που τα λαμβάνει. Η καρτέλα «Approved Patient» είναι οι εγκεκριμένοι ασθενείς, όπου υπάρχει και η δυνατότητα να διαγραφούν. Τέλος η καρτέλα «Find Patients» είναι για την αναζήτηση ασθενών, όπως ο ασθενής κάνει αναζήτηση γιατρών για να στείλει αίτημα.



Εικόνα 5.37 - Σελίδα γραφημάτων



Εικόνα 5.38 – Ασθενείς και εύρεση ασθενών

5.3.5 Σελίδα φαρμάκων

Η επόμενη υποσελίδα «Prescriptions» αφορά την συναγογράφηση των φαρμάκων σε ασθενείς. Υπάρχουν δύο καρτέλες: η πρώτη «Prescribe» είναι η φόρμα που ο γιατρός συμπληρώνει τις πληροφορίες για τα φάρμακα και η δεύτερη «Prescriptions» είναι όλες οι συνταγογραφήσεις του γιατρού.

Εικόνα 5.39 – Σελίδα προσθήκης νέου φαρμάκου

Στο δεύτερο άσπρο πλαίσιο ο γιατρός μπορεί να επιλέξει έναν από τους ασθενείς που έχει εγκρίνει και υπάρχει κουμπί για ανανέωση των ασθενών. Στο τρίτο άσπρο πλαίσιο βρίσκεται η φόρμα που συμπληρώνει ο γιατρός. Οι πληροφορίες που βάλαμε είναι η μάρκα του φαρμάκου, η δόση, η συχνότητα, οδηγίες όπως για παράδειγμα πότε πρέπει να λαμβάνεται το φάρμακο και πότε λήγει η συνταγή. Τέλος, υπάρχει κουμπί για να δημιουργηθεί το αίτημα. Εάν το αίτημα γίνει σωστά, τότε θα εμφανιστεί το «Success».

Expires : 07/24/2022

Success Submit Again

Εικόνα 5.40 - Επιτυχία αίτησης για νέο φάρμακο

Εάν λείπει κάποια πληροφορία τότε θα εμφανιστεί σφάλμα με κόκκινα γράμματα «Missing Inputs».

Expires : 07/24/2022

Missing Inputs

Submit

Εικόνα 5.41- Σφάλμα αιτήματος για νέο φάρμακο

Στην δεύτερη καρτέλα «Prescriptions» ο γιατρός μπορεί να δει ποια φάρμακα έχει γράψει σε ποιον ασθενή του ή γενικά όλα τα φάρμακα που έχει γράψει. Μπορεί να δει πληροφορίες για αυτά και να τα διαγράψει.

Prescriptions

Prescribe Prescriptions

To: Angelos Chostelidis Refresh Patients

All To: Angelos Chostelidis To: Xristos Papadopoulos

Brand: Levetiracetam.

Frequency: 2xDay

Dose: 500mg

Instructions: Take a tablet before breakfast, and a tablet before bed.

Status: Active

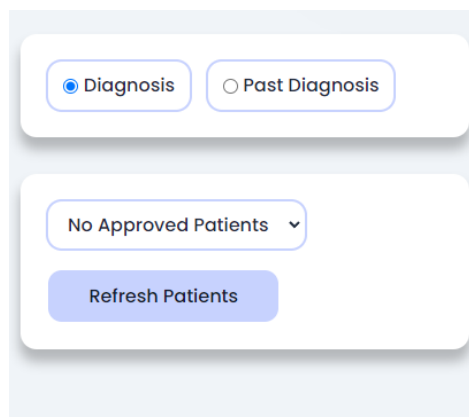
Delete

Εικόνα 5.42 - Λίστα συνταγογραφούμενων φαρμάκων

5.3.6 Σελίδα ρίσκων

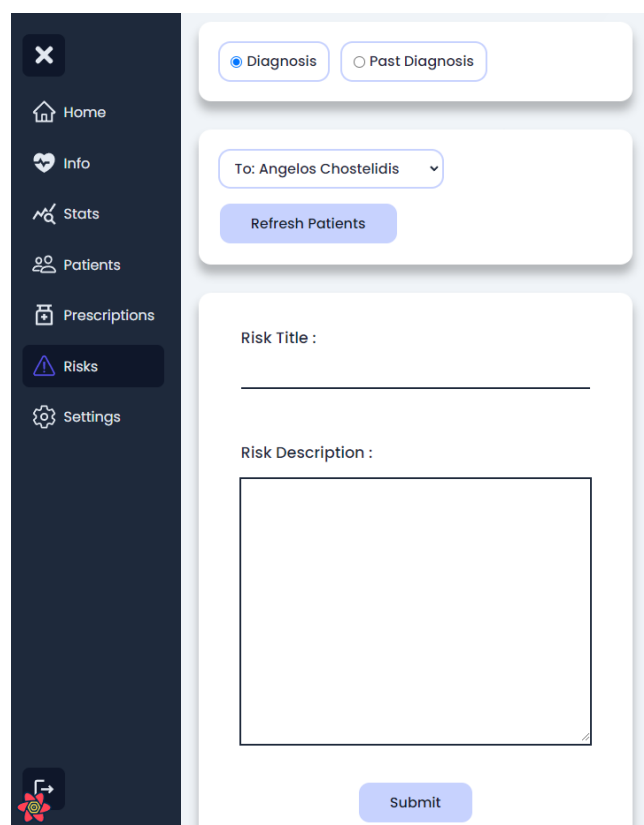
Η επόμενη υποσελίδα «Risks», αφορά τα ρίσκα που μπορεί να γράψει ο γιατρός στους ασθενείς του. Όπως και στην προηγούμενη υποσελίδα και αυτή αποτελείται από δύο καρτέλες. Μία καρτέλα που περιέχει μία φόρμα για να γράψει τα στοιχεία του ρίσκου και μία καρτέλα που μπορεί να δει ποια ρίσκα έχει γράψει και σε ποιον ασθενή. Στην πρώτη καρτέλα «Diagnosis» το δεύτερο άσπρο πλαίσιο

αφορά την επιλογή του ασθενή που θέλει ο γιατρός να στείλει αίτημα για το νέο ρίσκο. Εάν δεν υπάρχουν ασθενείς τότε σε εκείνο το πλαίσιο αναγράφεται «No approved patients» και δεν εμφανίζεται η φόρμα για την συμπλήρωση του ρίσκου.



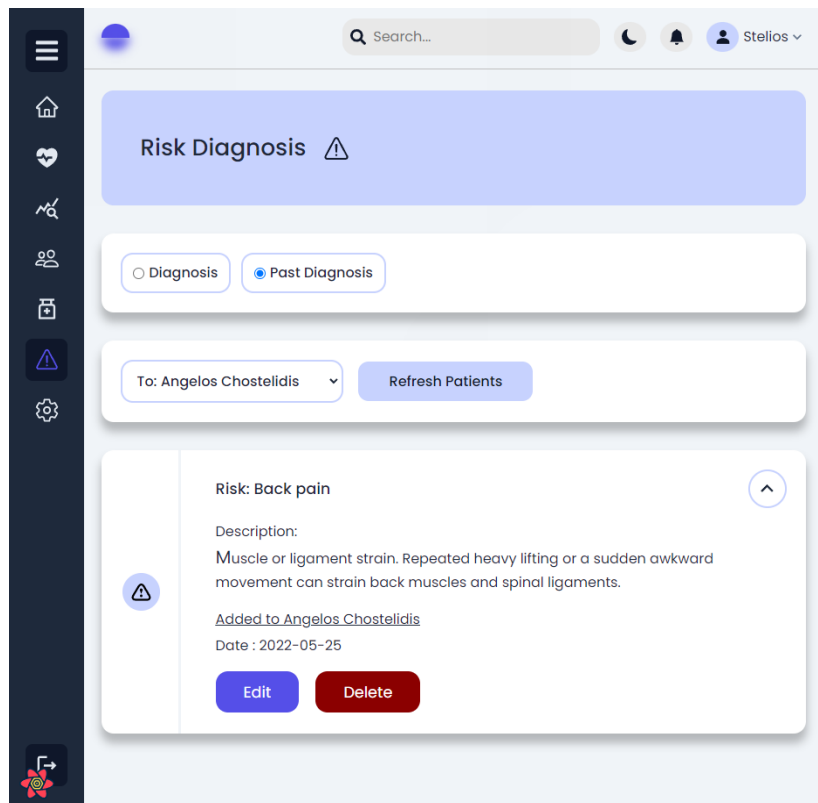
Εικόνα 5.43 - Λίστα ασθενών

Η φόρμα του ρίσκου αποτελείται από τον τίτλο και ένα πλαίσιο που μπορούν να συμπληρωθούν πληροφορίες γι' αυτό.



Εικόνα 5.44 - Φόρμα ρίσκων

Στην καρτέλα «Past Diagnosis» ,όπως αναφέρθηκε,ο γιατρός μπορεί να δει τα ρίσκα που έχει γράψει στους ασθενείς του, να τα διαγράψει ή να αλλάξει τις πληροφορίες.



Εικόνα 5.45 - Λίστα ρίσκων

Η τελευταία υποσελίδα αφορά τις ρυθμίσεις και είναι ακριβώς ίδια με αυτή του ασθενή. Δηλαδή, υπάρχουν οι ρυθμίσεις για τον κωδικό πρόσβασης, το e-mail, το ονοματεπώνυμο κτλπ.

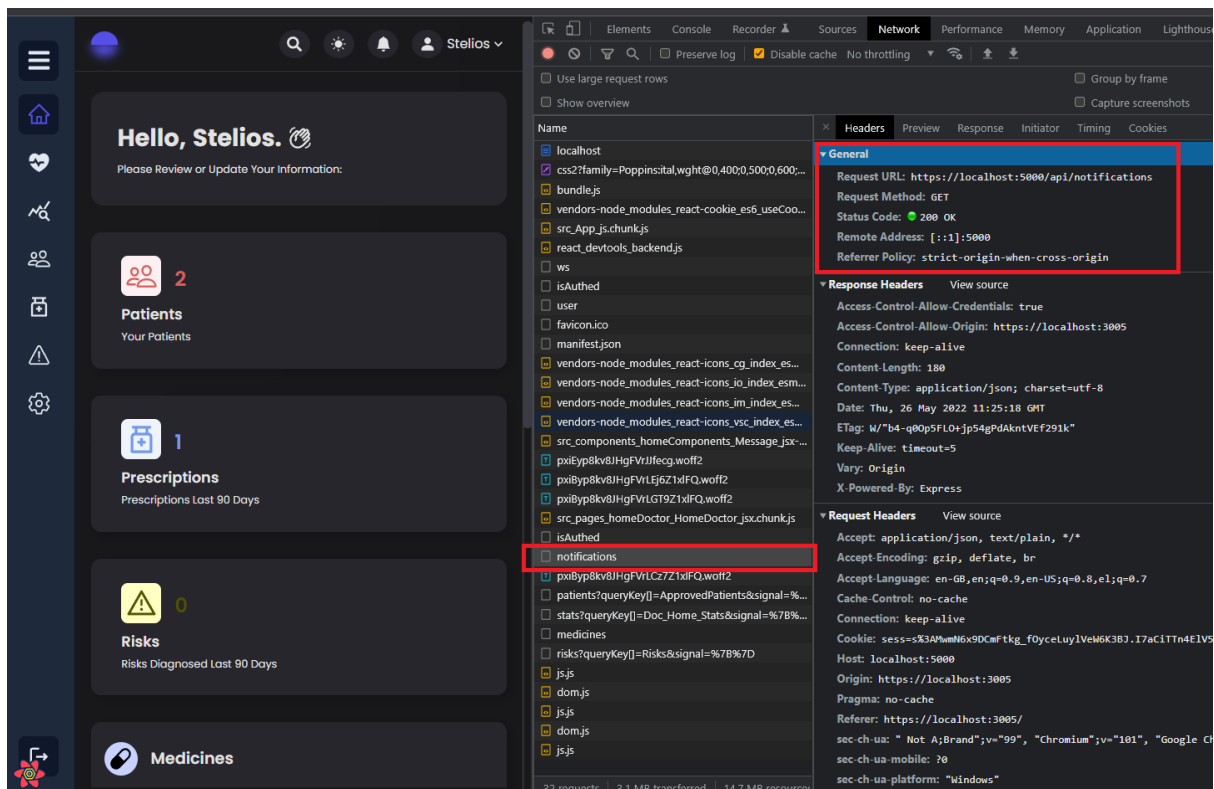
5.4 Restful API

Όπως έχει αναφερθεί το λογισμικό του server έχει δημιουργηθεί με το framework express.js. Ουσιαστικά έχει φτιαχτεί ένας HTTP(S) server. Η εφαρμογή της συσκευής, όπως και η εφαρμογή στο διαδίκτυο, επικοινωνεί με το API (Application Programming Interface) του server. Το RESTFUL API είναι ένα API το οποίο είναι δομημένο σύμφωνα με την αρχιτεκτονική REST. Το API είναι ουσιαστικά ο τρόπος που αλληλεπιδρούν δύο εφαρμογές μεταξύ τους. Το RESTFUL API χρησιμοποιεί HTTP αιτήματα για να διαχειρίζεται τις πληροφορίες. Αυτές οι πληροφορίες μπορούν να αξιοποιηθούν μέσω των αιτημάτων GET, PUT, POST και DELETE τα οποία κάθε ένα από αυτά εκτελεί και διαφορετική λειτουργία. Το RESTFUL API στέλνει τα HTTP αιτήματα στον server και αυτός απαντάει αντίστοιχα [22].

Τα HTTP αιτήματα [22]:

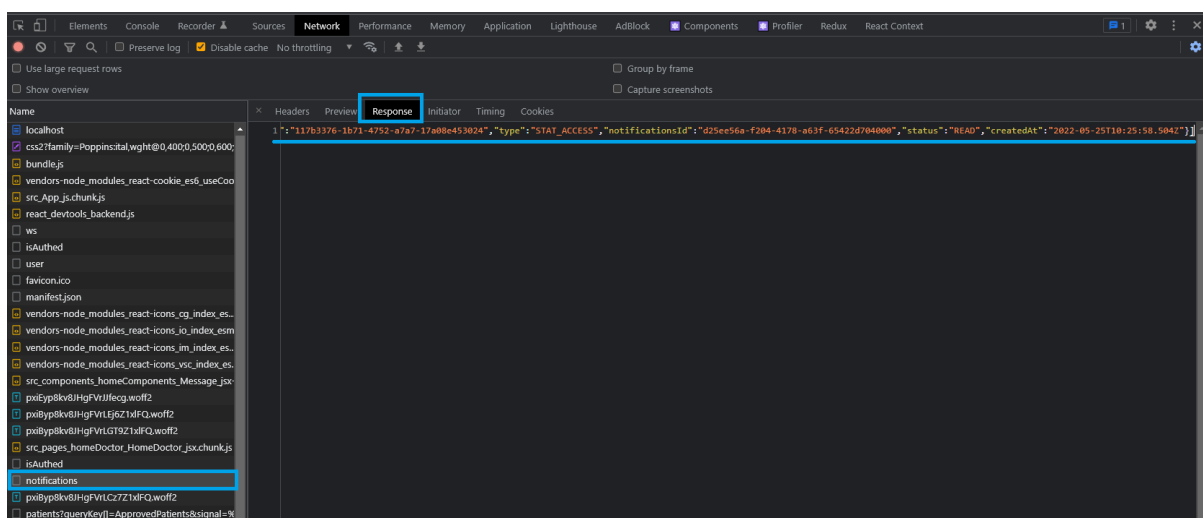
- Το GET χρησιμοποιείται για να ζητήσουμε και να πάρουμε μια πληροφορία από τον server.
- Το PUT είναι για να αλλάξουμε μια ήδη υπάρχουσα πληροφορία του server.
- Το POST υπάρχει για την δημιουργία μιας καινούργιας πληροφορίας στον server.
- Το DELETE χρησιμοποιείται για να διαγράψει μια πληροφορία στον server.

Τα HTTP αιτήματα που στέλνονται στον server μπορούμε εύκολα να τα δούμε αν πάμε στις ρυθμίσεις για προγραμματιστές σε έναν browser, όπως το google Chrome.



Εικόνα 5.46 - μέθοδοι HTTP

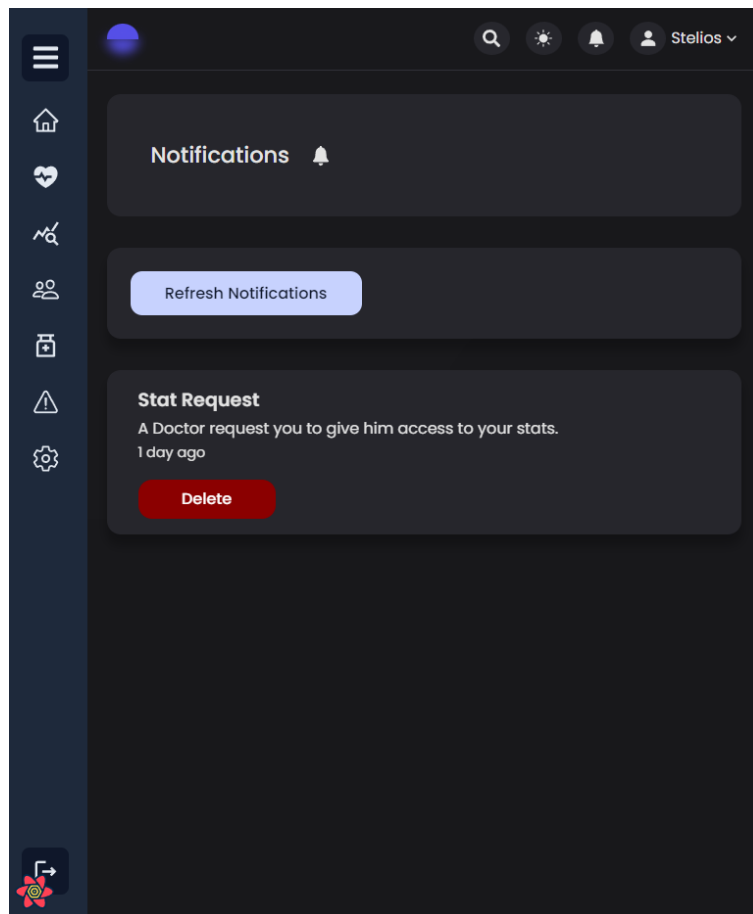
Για παράδειγμα στο παραπάνω στο πρώτο κόκκινο πλαίσιο βλέπουμε ένα HTTP request που ονομάζεται notifications, όπου ουσιαστικά είναι οι ειδοποιήσεις αυτού του λογαριασμού. Στο δεύτερο μεγαλύτερο κόκκινο πλαίσιο φαίνονται οι γενικές πληροφορίες του αιτήματος, δηλαδή από που ζητούνται οι πληροφορίες, τον τύπο του αιτήματος και τον κωδικό απάντησης, δηλαδή GET request και 200 OK αντίστοιχα.



Εικόνα 5.47 - Απάντηση αιτήματος HTTP

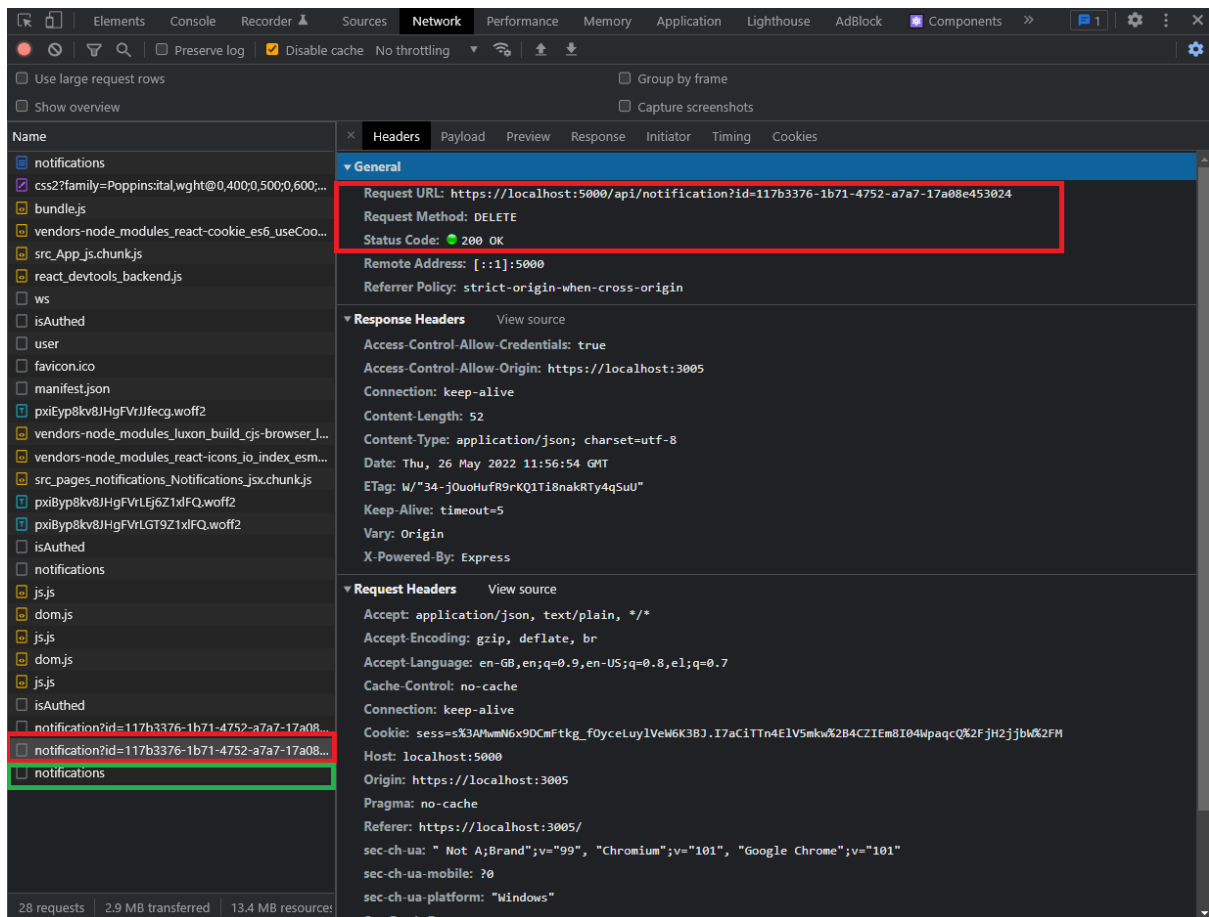
Στην καρτέλα response μπορούμε να δούμε την απάντηση του αίτηματος. Στην συγκεκριμένη περίπτωση είναι μια ειδοποίηση με τα στοιχεία της δηλαδή το id, type, status κλπ. Έτσι με βάση την απάντηση απεικονίζουμε στην εφαρμογή τα πράγματα που θέλουμε να δείξουμε. Ξέρουμε ότι η

ειδοποίηση είναι διαβασμένη από το «status:read», τον τύπο του αιτήματος-ειδοποίηση, «type:STAT_ACCESS» και την ημερομηνία που δημιουργήθηκε από το «createdAt : ...».



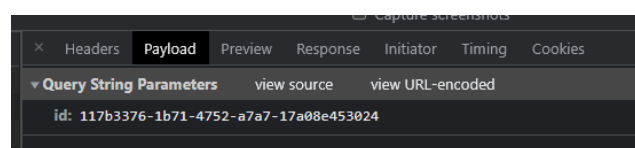
Εικόνα 5.48 - Ειδοποιήσεις

Έτσι με τα παραπάνω στοιχεία δείχνουμε και την συγκεκριμένη ειδοποίηση. Το κουμπί «Delete» ουσιαστικά αν πατηθεί θα στείλει νέο αίτημα HTTP τύπου delete.



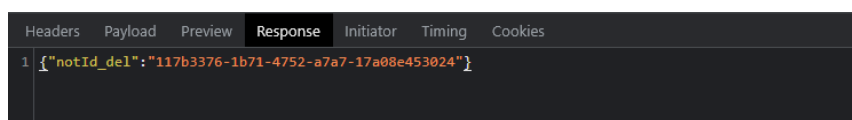
Εικόνα 5.49 - HTTP αίτημα για τις ειδοποιήσεις

Αφού πατηθεί το κουμπί «Delete» μπορούμε να δούμε το αίτημα HTTP-Delete που στέλνεται στον express Server. Αφού ληφθεί απάντηση γι' αυτό το αίτημα ξανακάνουμε αίτημα GET notifications για να πάρουμε τον νέο πίνακα με τις ειδοποιήσεις, ο οποίος στην συγκεκριμένη περίπτωση θα είναι άδειος αφού είχαμε μόνο μία ειδοποίηση.



Εικόνα 5.50 - Μεταβλητή HTTP αιτήματος

Στην καρτέλα Payload μπορούμε να δούμε τι μεταβλητές έχουμε στείλει μαζί με το αίτημα, Στην συγκεκριμένη περίπτωση έχουμε στείλει την «id», έτσι ώστε ο server να ξέρει ποια ειδοποίηση να διαγράψει.



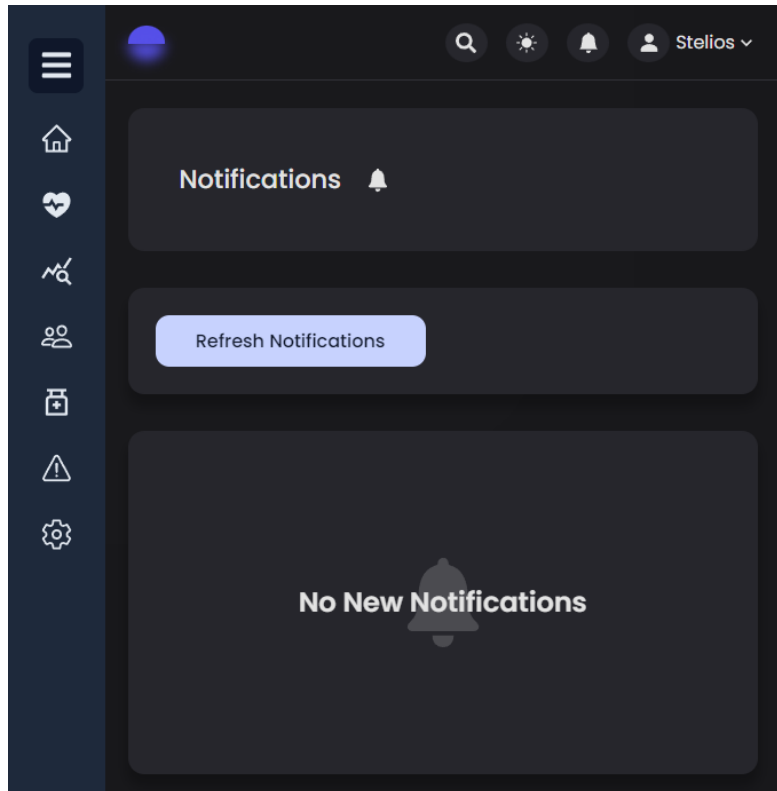
Εικόνα 5.51- Απάντηση DELETE HTTP αιτήματος

Στην καρτέλα Response βλέπουμε την απάντηση του server η οποία μας επιστρέφει μία μεταβλητή με όνομα notId_del που περιέχει το «id» της ειδοποίησης που διαγράφηκε. Επειδή το καινούριο αίτημα

HTTP-GET notifications επιτρέπει έναν κενό πίνακα απεικονίζουμε «No New Notifications» στην εφαρμογή.

Headers	Preview	Response	Initiator	Timing	Cookies
1		{} []			

Εικόνα 5.52 - Απάντηση GET HTTP αιτήματος



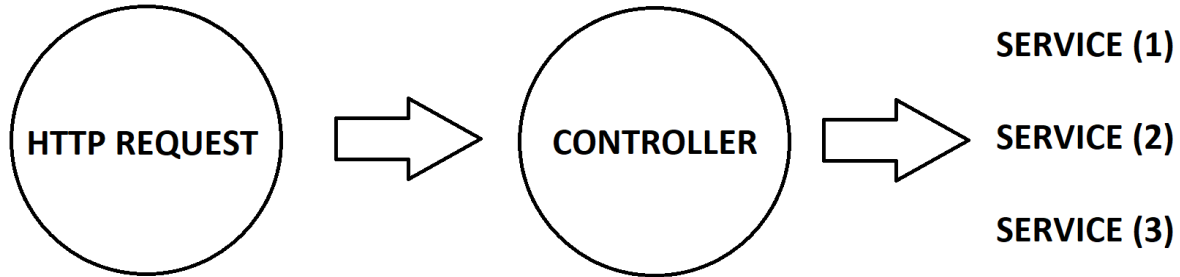
Εικόνα 5.53 - Νέα λίστα ειδοποιήσεων

Το κουμπί «Refresh Notifications» ουσιαστικά ξαναστέλνει αίτημα HTTP-GET notifications στον server, κι αν υπάρχουν νέες ειδοποιήσεις θα εμφανιστούν στην εφαρμογή. Γενικότερα με αυτόν τον τρόπο λειτουργεί και όλη η εφαρμογή από τις μετρήσεις μέχρι και την αποδοχή-απόρριψη αιτημάτων που στένλει γιατρός ή ασθενής. Όλες οι παραπάνω πληροφορίες φυσικά αποθηκεύονται σε βάση δεδομένων και ανάλογα με τα αιτήματα HTTP, τις επεξεργαζόμαστε.

5.5 Λογική HTTP αιτημάτων

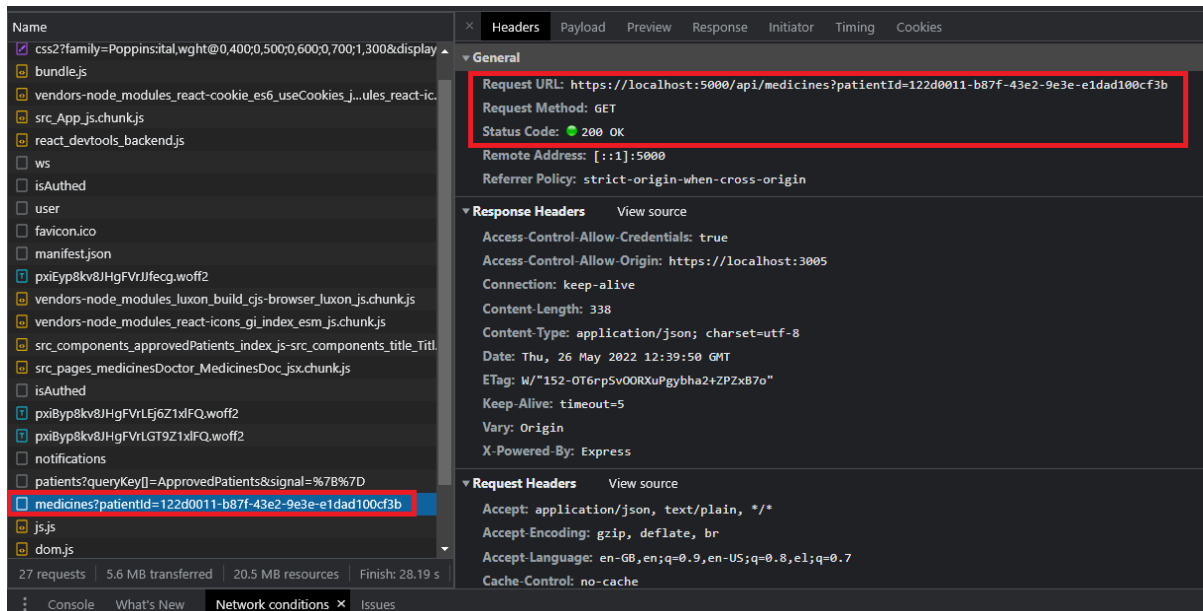
Φυσικά στέλοντας αίτηματα HTTP δεν κάνουν την δουλειά που θέλουμε αυτόματα και έχει προγραμματιστεί λογική που αφορά τι πρέπει να κάνει ο server για κάθε ένα από αυτά τα αιτήματα. Δηλαδή, θα μπορούσαμε με ένα αίτημα GET να γίνεται διαγραφή στοιχείων, καθώς αυτή η συμπεριφορά δεν είναι προβλέψιμη, ακολουθούμε ορισμένους κανόνες, δηλαδή για ένα αίτημα GET Πρέπει απλά να δεχόμαστε πληροφορίες χωρίς να γίνεται κάποια αλλαγή στην βάση δεδομένων. Για την αρχιτεκτονική του RESTFUL API χρησιμοποιήθηκε το MVC (Model View Controller). Στο Controller κομμάτι βρίσκεται όλος ο προγραμματισμός για κάθε αίτημα. Στο παραπάνω παράδειγμα για την αποκτήση των ειδοποιήσεων καλείται ο controller, δηλαδή ο ελεγχτής του notifications, όπου θα περιείχε τον προγραμματισμό για την μέθοδο GET, DELETE κλπ. Όμως επειδή έχουμε πολλά προγραμματισμένα αιτήματα και επειδή με ένα αίτημα DELETE μπορεί να θέλουμε να διαγράψουμε

για παράδειγμα δύο διαφορετικά πράγματα ταυτόχρονα, χωρίσαμε τα αιτήματα σε ξεχωριστά αρχεία που τα αποκαλούμε SERVICES. Επομένως, χρησιμοποιούμε αρχιτεκτονική MVCS. Έτσι με αυτόν τον τρόπο ο controller βρίσκεται εκεί απλά για να καλέσει τις κατάλληλες υπηρεσίες (Services). Οπότε αποφεύγουμε να ξαναγράψουμε τον ίδιο κώδικα παράπανω από μία φορά και μπορούμε να καλέσουμε την ίδια υπηρεσία σε διαφορετικούς ελεγχτές.



Εικόνα 5.54 - Αλληλεπίδραση ελεγκτών με εσωτερικές υπηρεσίες

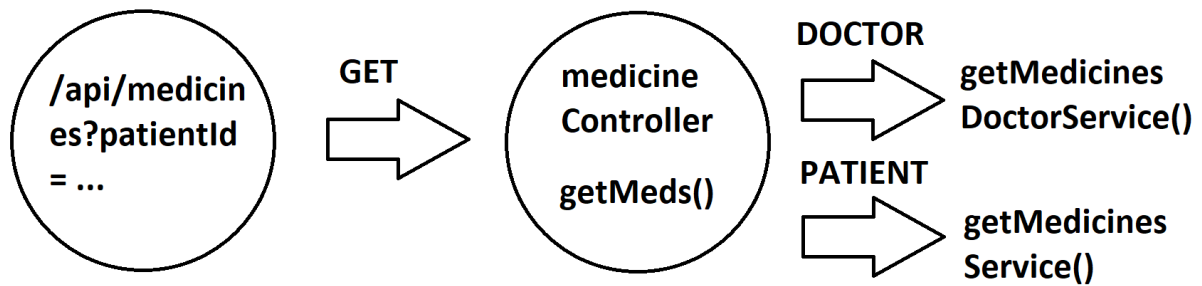
Για παράδειγμα μπορούμε να πάρουμε, στην εφαρμογή της πτυχιακής, τον ελεγχτή που έχει να κάνει με τα φάρμακα.



Εικόνα 5.55 - HTTP αίτημα για φάρμακα

Το αίτημα HTTP-GET, το οποίο φαίνεται παραπάνω, καλεί το URL localhost:5000/api/medicines με μία μεταβλητή patientID, δηλαδή ζητάει τα φάρμακα του ασθενή με το συγκεκριμένο ID. Το συγκεκριμένο URL καλεί τον Controller που ονομάζεται MedicineController, ο οποίος με την σειρά του για το αίτημα GET καλεί την συνάρτηση που έχουμε ονομάσει GetMeds. Η συνάρτηση αυτή, ανάλογα με τον ρόλο του χρήστη, δηλαδή αν είναι γιατρός ή ασθενής, καλεί διαφορετική υπηρεσία. Για παράδειγμα, αν ο χρήστης είναι γιατρός και έχει εγκρίνει τον ασθενή με το συγκεκριμένο ID θα του επιστραφούν τα φάρμακα που του έχει γράψει. Όλη αυτή η λογική έχει οριστεί στην υπηρεσία που έχει να κάνει με τον γιατρό. Αν όμως ο χρήστης είναι ασθενής θα κληθεί διαφορετική υπηρεσία η οποία δεν έχει ορίσει την μεταβλητή patientID, οπότε θα αγνοηθεί και η συγκεκριμένη υπηρεσία απλά θα επιστρέψει τα τρία τελευταία φάρμακα του. Φυσικά, ο μόνος τρόπος να γίνει κλήση του συγκεκριμένου URL με αυτή την μεταβλητή είναι εκτός εφαρμογής με κάποια άλλη εξωτερική, διότι

μέσα στην εφαρμογή δεν έχει ορισθεί κουμπί που να στέλνει ακριβώς αυτό το αίτημα από την πλευρά του ασθενή, διότι δεν χρειάζεται αφού η μεταβλητή δεν λειτουργεί.



Εικόνα 5.56 - Αλληλεπίδραση του ελεγκτή φαρμάκων με εσωτερικές υπηρεσίες

Έτσι χωρίζουμε όλα τα αιτήματα μεταξύ γιατρού και ασθενή. Δηλαδή, αν πάει ένας ασθενής να γράψει φάρμακα σε κάποιον άλλο ασθενή μέσω εξωτερικής εφαρμογής, η αντίστοιχη υπηρεσία δεν θα κληθεί ποτέ γιατί ο χρήστης δεν θα είναι γιατρός και αντί αυτού ο ελεγκτής θα επιστρέψει σφάλμα.

5.6 Επίλογος

Η διαδικτυακή εφαρμογή επικοινωνεί και λαμβάνει τα δεδομένα που χρειάζεται από τον server, τα οποία στην συνέχεια απεικονίζονται με εύκολο και κατανοητό για τον χρήστη. Η επικοινωνία αυτή επιτυγχάνεται με την χρήση μεθόδων HTTP. Η διαδικτυακή εφαρμογή παρέχει διαφορετικές λειτουργίες για τον ασθενή και διαφορετικές για τον γιατρό. Ο ασθενής μπορεί να προσθέσει, να αφαιρέσει, να διαγράψει ή να τροποποιήσει τα βιομετρικά του δεδομένα. Ο γιατρός, αφού του δοθεί πρόσβαση από τον ασθενή, μπορεί οποιαδήποτε στιγμή να ελέγχει τα δεδομένα αυτά. Επιπλέον, ο γιατρός έχει την δυνατότητα να συνταγογραφήσει φάρμακα ή να προσθέσει παρατηρήσεις για την κατάσταση της υγείας του ασθενή. Ο ασθενής από πλευράς του αποδέχεται ή απορρίπτει τα αιτήματα αυτά. Για κάθε νέο αίτημα που στέλνει γιατρός ή ασθενής δημιουργείται νέα ειδοποίηση στο λογαριασμό τους. Οι ειδοποιήσεις υπενθυμίζουν τον χρήστη για νέες υποψήφιες αλλαγές δεδομένων. Ακόμη, υπάρχει δυνατότητα απεικόνισης των βιομετρικών δεδομένων σε γραφήματα. Παρέχονται φίλτρα σε γιατρό και ασθενή για την αναζήτηση αυτών με συγκεκριμένα κριτήρια, όπως η ημερομηνία δημιουργίας.

Κεφάλαιο 6ο: Ασφάλεια

6.1 Εισαγωγή

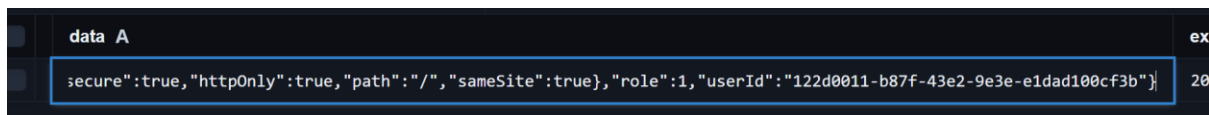
Το πιο σημαντικό κομμάτι της εργασίας όσο αναφορά την ασφάλεια είναι η επικύρωση της ταυτότητας του χρήστη. Δηλαδή το πως γνωρίζει το server αν ο χρήστης είναι αυτός που λέει. Για παράδειγμα πως το λογισμικό επιστρέφει μόνο τα φάρμακα που πρέπει να επιστρέψει σε κάποιον χρήστη. Αυτή η αναγνώριση γίνεται με την χρήση των Cookies.

6.2 Διαδικασία αναγνώρισης χρήστη με cookies

Όταν ο χρήστης στέλνει τα στοιχεία του στον server, δηλαδή την ηλεκτρονική του διεύθυνση και τον κωδικό πρόσβασης, το server τα συγκρίνει με τα αποθηκευμένα στοιχεία στην βάση δεδομένων. Βέβαια, ο αποθηκευμένος κωδικός πρόσβασης είναι κωδικοποιημένος. Για το συγκεκριμένο θέμα θα γίνει αναφορά παρακάτω. Αφού αποκωδικοποιηθεί ο κωδικός, συγκρίνεται με τον κωδικό που έχει στείλει ο χρήστης. Αντίστοιχα συγκρίνεται η ηλεκτρονική διεύθυνση με αυτή που είναι αποθηκευμένη. Εάν όλα τα στοιχεία είναι σωστά, τότε δημιουργείται ένα cookie που περιέχει διάφορες πληροφορίες και οι ίδιες αυτές πληροφορίες αποθηκεύονται και στην βάση δεδομένων. Το cookie αποστέλλεται σαν απάντηση από τον server στον χρήστη. Επίσης, το ίδιο το cookie που στέλνεται είναι και αυτό κωδικοποιημένο. Η λέξη κλειδί για την αποκωδικοποίηση του βρίσκεται στο server και δεν την γνωρίζει κανένας. Εάν ο χρήστης πάει στα εργαλεία του browser του και βρει τα cookie, θα δει κάτι σαν αυτό :

```
"s%3AfuVvTNkHPcA5L81e8BDRw75QkKv-  
Sk0a.bWi5zVXM1Zr5ZainwF%2BEWG5Nl6kJNoHA9Susa3FSWOc"
```

Ο λόγος είναι, όπως αναφέρθηκε, ότι είναι κωδικοποιημένες οι πληροφορίες. Στην βάση δεδομένων μπορούμε να δούμε τις πραγματικές πληροφορίες του cookie



Εικόνα 6.1 - Αποκωδικοποιημένες πληροφορίες Cookie

Οι σημαντικές πληροφορίες που έχουμε προσθέσει στο cookie είναι ο ρόλος και το μοναδικό ID του χρήστη, δηλαδή όπως φαίνεται το userId. Έτσι αφού ο χρήστης συνδεθεί στην εφαρμογή και αρχίζει να στέλνει αιτήματα, μαζί με κάθε αίτημα στέλνεται και το cookie που έχει από την σύνδεση, το οποίο αποκωδικοποιείται από τον server και συγκρίνεται με τις αποθηκευμένες πληροφορίες. Εάν οι πληροφορίες ταιριάζουν με τις αποθηκευμένες, τότε με το userID, επειδή κάθε λογαριασμός έχει μοναδικό, ο server μπορεί να ξέρει ποιος είναι ο χρήστης και τον ρόλο του.

6.3 Δικλείδες ασφαλείας των cookies

Μπορούμε να πάρουμε διάφορα μέτρα έτσι ώστε να κάνουμε το Cookie όσο πιο ασφαλές γίνεται. Αρχικά, όπως αναφέρθηκε το Cookie είναι κωδικοποιημένο και ο χρήστης δεν μπορεί να καταλάβει τις πληροφορίες που έχει. Εάν όμως με κάποιον τρόπο αποκωδικοποιηθεί ή απλά αν ο χρήστης πειράξει το Cookie, όταν σταλθεί στον server, δεν θα είναι έγκυρο διότι συγκρίνεται με τις πληροφορίες που είναι αποθηκευμένες στην βάση δεδομένων όπου είναι ίδιες με αυτές του αρχικού Cookie που δημιουργήθηκε. Επιπλέον, χρησιμοποιώντας το Cookie, υπάρχει η δυνατότητα να

παρακολουθείται ο χρήστης, όπως για παράδειγμα πόσες φορές έχει επισκεπτεί την ιστοσελίδα ή κάτι άλλο χρησιμοποιώντας μεταβλητές. Δημιουργώντας ένα HTTP Cookie μας δίνεται η δυνατότητα να χρησιμοποιήσουμε κάποιες ειδικές σημαίες ή χαρακτηριστικά (flags ή attributes) για να το κάνουμε πιο ασφαλές. Αυτά που έχουμε χρησιμοποιήσει στην πτυχιακή εργασία είναι τα εξής:

- httpOnly
- secure
- sameSite

6.3.1 HTTP ONLY

Το httpOnly είναι μια ιδιότητα που αποτρέπει προγράμματα ή αλγορίθμους που τρέχουν στην πλευρά του χρήστη (Client Side) να έχουν πρόσβαση στο Cookie. Έτσι, αποτρέπεται η μέθοδος cross-site scripting (XSS), η οποία στέλνει δεδομένα του χρήστη στην διαδικτυακή ιστοσελίδα αυτού που επιτίθεται [24].

6.3.2 SECURE

Η ιδιότητα secure είναι μία επιλογή που ορίζεται από τον server καθώς στέλνει ένα καινούριο cookie στον χρήστη μέσα στην απάντηση HTTP. Ο σκοπός αυτής της επιλογής είναι να αποτρέψει τους ανεξουσιοδοτημένους χρήστες να παρακολουθήσουν το cookie αυτό. Έτσι με αυτή την επιλογή οι browser θα στείλουν το cookie μόνο σε HTTPS σελίδα και όχι σε ένα μη κρυπτογραφημένο HTTP κανάλι [24].

6.3.3 SAMESITE

Η επιλογή samesite έχει οριστεί με την πιο αυστηρή του ιδιότητα, η οποία ονομάζεται strict. Αυτή η ιδιότητα αποτρέπει τα cookie να στέλνονται σε τρίτα website. Έτσι το cookie θα στέλνεται μόνο σε αιτήματα που προέρχονται από το site που δημιουργήθηκε. Με αυτό τον τρόπο αποτρέπουμε μία από τις πιο δημοφιλείς επιθέσεις στους χρήστες, η οποία ονομάζεται Cross-Site Request Forgery (CSRF) [24].

6.4 HTTPS

Ένα ακόμη μέτρο που πήραμε για την ασφάλεια είναι η χρήση το HTTPS. Η χρήση του ήταν αναγκαία, γιατί όπως αναφέρθηκε χρησιμοποιήθηκε και η ιδιότητα secure στα cookies. Το HTTPS είναι ένα πρωτόκολλο το οποίο υπάρχει για να είναι ασφαλής η επικοινωνία μεταξύ ενός περιηγητή και μιας διαδικτυακής εφαρμογής. Το HTTPS χρησιμοποιεί κρυπτογράφηση κειμένου προκειμένου να αυξήσει την ασφάλεια σε πληροφορίες οι οποίες είναι σημαντικές, όπως είναι για παράδειγμα οι κωδικοί πρόσβασης. Το πρωτόκολλο που χρησιμοποιεί το HTTPS για την κρυπτογράφηση ονομάζεται TLS.

Για να γίνει η κρυπτογράφηση υπάρχει ένα σύστημα με δύο κλειδιά. Το πρώτο κλειδί είναι δημόσιο και ουσιαστικά μπορεί να χρησιμοποιηθεί από όποιον θέλει να επικοινωνήσει με την διαδικτυακή εφαρμογή με ασφάλεια. Όμως, τα κρυπτογραφημένα δεδομένα μπορούν να αποκωδικοποιηθούν μόνο από αυτούς που έχουν το δεύτερο κλειδί [25].

6.5 Κρυπτογραφημένη αποθήκευση κωδικών πρόσβασης

Σαν τελευταίο μέτρο ασφαλείας πήραμε την κρυπτογραφημένη αποθήκευση κωδικών πρόσβασης. Οι κωδικοί πρόσβασης των χρηστών πρέπει να είναι κρυπτογραφημένοι για την ασφάλεια τους, γιατί

πολλοί από αυτούς χρησιμοποιούν τους ίδιους για πολλές υπηρεσίες. Οι κωδικοί πρόσβασης στην βάση δεδομένων φαίνονται στην παρακάτω φωτογραφία:

lastName A	email A	password A
Chostelidis	hostelidis99@gmail.com	\$2a\$10\$eFdIxGg8UXPzuF1XRZtnc.9vOuH7FVCqqQpBTrxaEVRRoKnaQgggi
Tragos	doc@doc.com	\$2a\$10\$mpb8UX94NfvZdp3ve6Kb7eEVptZdKswzXbkdVLx1tNkXIgDBP04se

Εικόνα 6.2 - Κωδικοί πρόσβασης στην βάση δεδομένων

Για να το πετύχουμε αυτό χρησιμοποιήσαμε τις τεχνικές hashing και salting.

6.6 Hashing και Salting

Το Hashing είναι μια διαδικασία κατά την οποία τα δεδομένα κωδικοποιούνται με έναν συγκεκριμένο τρόπο. Ανάλογα με την μέθοδο που θα χρησιμοποιηθεί η λέξη που θα παρθεί θα αποτελείται από συγκεκριμένο αριθμό γραμμάτων και αριθμών ανεξαρτήτως της πληροφορίας που θα του δώσουμε να μετατρέψει. Κάθε πληροφορία αντιστοιχίζεται σε μία μόνο λέξη και όσες φορές και να την μετατρέψουμε θα είναι πάντα η ίδια. Επίσης, η διαδικασία του Hashing είναι μονόδρομη. Αυτό σημαίνει ότι δεν σου επιτρέπει να επαναφέρεις μια κωδικοποιημένη λέξη στην αρχική της μορφή. Επειδή όμως κακόβουλοι χρήστες είναι πιθανό να αποκτήσουν πρόσβαση στην βάση δεδομένων μπορούν αρχικά να κλέψουν το κωδικοποιημένο μήνυμα και έπειτα μέσω ορισμένων εργαλείων μπορούν να το αποκωδικοποιήσουν. Για αυτόν τον λόγο εφευρέθηκε η τεχνική salting. Στην τεχνική αυτή μπαίνει ουσιαστικά μια λέξη στο τέλος του μηνύματος και μετά γίνεται η κωδικοποίηση του, έτσι ώστε ακόμα και κάποιος να αποκτήσει πρόσβαση στην κωδικοποιημένη λέξη να μην μπορεί να καταλάβει ποιο είναι το μήνυμα και ποια η λέξη που προστέθηκε [26].

6.7 Επίλογος

Η ασφάλεια του συστήματος επιτυγχάνεται με διαφορετικούς τρόπους ταυτόχρονα. Η αναγνώριση της ταυτότητας του χρήστη από τον server είναι δυνατή με την χρήση Cookies. Η ασφάλεια των Cookies ενισχύεται με την χρήση ιδιοτήτων που μας προσφέρουν. Αυτές οι ιδιότητες είναι η httpOnly, η secure, και η samesite. Επιπλέον, η χρήση του πρωτοκόλλου HTTPS προσφέρει ασφαλή και κρυπτογραφημένη επικοινωνία μεταξύ διακομιστή και χρήστη. Τέλος, σε περίπτωση διαρροής στοιχείων από την βάση δεδομένων, οι κωδικοί πρόσβασης των λογαριασμών δεν θα μπορούν να διαβαστούν από κακόβουλους χρήστες γιατί είναι κρυπτογραφημένοι. Η κρυπτογράφηση αυτή υλοποιήθηκε για την προστασία των χρηστών της εφαρμογής, διότι παρατηρείται ότι γίνεται χρήση ίδιων κωδικών πρόσβασης σε πολλές και διαφορετικές διαδικτυακές υπηρεσίες, όπως για παράδειγμα σε ηλεκτρονικές συναλλαγές ή σε μέσα κοινωνικής δικτύωσης.

Κεφάλαιο 7ο: Συμπεράσματα και προτάσεις βελτίωσης

Συμπερασματικά, η παρούσα εργασία αφορά μία διαδικτυακή εφαρμογή, η οποία έχει δημιουργηθεί από την αρχή με σκοπό την προσαρμογή της στις απαιτήσεις και στις προσδοκίες που συμβάδιζαν με τις λειτουργίες της. Την εφαρμογή αυτή συμπληρώνει και μία εξωτερική συσκευή, της οποίας η χρήση είναι να συλλέγει βιομετρικά δεδομένα από τον χρήστη με την βοήθεια αισθητήρων. Τα δεδομένα αυτά στέλνονται στην παραπάνω εφαρμογή διαδικτυακά. Όπως πολλές εφαρμογές, έτσι και η συγκεκριμένη, αποτελείται κυρίως από δύο κομμάτια, το frontend και το backend. Τα δύο αυτά κομμάτια είναι ανεπτυγμένα ξεχωριστά και επικοινωνούν με το πρωτόκολλο http(s). Είναι γεγονός πως υπάρχει πληθώρα τεχνολογιών για την ανάπτυξη των διαδικτυακών εφαρμογών. Στην συγκεκριμένη περίπτωση αξιοποιήθηκαν η React.js (για το παρουσιαστικό κομμάτι) και η Express.js (για το λογισμικό του διακομιστή). Αναφορικά με τις λειτουργίες της εφαρμογής, αυτές ποικίλλουν ανάλογα με το είδος του χρήστη (ασθενής- γιατρός). Τέλος, σημαντικό κομμάτι της εφαρμογής που δημιουργήθηκε αποτέλεσε η ασφάλεια των δεδομένων και των στοιχείων του χρήστη. Για την επίτευξη αυτής χρησιμοποιήθηκαν 3 μέθοδοι: cookies, κρυπτογράφηση επικοινωνίας μεταξύ διακομιστή και χρήστη (https) και η κωδικοποιημένη αποθήκευση κωδικών πρόσβασης στην βάση δεδομένων.

Όσο αναφορά τις προτάσεις βελτίωσης σχετικά με την ανάπτυξη της ηλεκτρονικής συσκευής, αυτές είναι οι εξής: Πρώτον, με σκοπό την εξοικονόμηση του χώρου θα μπορούσαμε να σχεδιάσουμε την ηλεκτρονική πλακέτα (pcb) με τέτοιο τρόπο ώστε τα ηλεκτρονικά της στοιχεία να έχουν μικρότερη απόσταση μεταξύ του. Επιπλέον, για αυτόν τον σκοπό θα μπορούσαμε να εκμεταλλευτούμε μόνο το μέρος της πλατφόρμας (ESP32) που αφορά τον μικροελεγκτή. Δεύτερον, η προσθήκη κυκλώματος που θα μετρά της τάση θα μπορούσε να προβλέπει και να ενημερώνει τον χρήστη για τον πιθανό εναπομείναντα χρόνο χρήσης της. Τέλος, μια ακόμα πρόταση βελτίωσης που θα μπορούσαμε να αξιοποιήσουμε είναι η προσθήκη διαφορετικού τύπου βυσμάτων με σκοπό την χρήση εναλλακτικών αισθητήρων. Σχετικά με την διαδικτυακή εφαρμογή, και λόγω του ότι αποτελεί ένας άμεσος τρόπος αλληλεπίδρασης μεταξύ γιατρού και ασθενή, θα μπορούσε να προσθεθεί μια συνομιλία πραγματικού χρόνου. Επιπρόσθετα, θα μπορούσαμε να προσθέσουμε την δυνατότητα ανταλλαγής αρχείων μεταξύ γιατρού και ασθενή. Με αυτόν τον τρόπο ο ασθενής θα μπορεί να ανεβάσει αρχεία εξετάσεων του και ο γιατρός να έχει άμεση πρόσβαση σε αυτά, να τα μελετά και ακόμη να τα συγκρίνει με τυχόν προηγούμενες εξετάσεις.

Βιβλιογραφία

Data Sheet

- [1] Recton Semiconductors, “Shottky barrier rectifier voltage range 20 to 40 Volts Current 3.0 Ampere”, 1N5822-T.
- [2] Taiwan semiconductor , “3A / 150kHz Step-Down DC-DC Converter”, TS2596CM550 RNG.
- [3] Cui Devices , “DC POWER JACK”, PJ-002BH.
- [4] Espressif , Esp32-DevKitc.
- [5] Wurth Elektronik, “Aluminium Electrolytic Capacitors - SMD WCAP-ASLI 470uF 35V 20%”, 865080562016.
- [6] Kemet, “Multilayer Ceramic Capacitors MLCC - SMD/SMT 50Vo 0.1uF C0G 1812 5% AEC-Q200”, C1210C104MBRACAUTO.
- [7] Bourns, “Fixed Inductors 33uH 20% SMD 1260”, SRR1260-330M.
- [8] MLX90614, contactless IR temperature sensor.
- [9] MAX30102, High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health

Internet Site

- [10] React.js , “A JavaScript library for building user interfaces”, 2022. Available: <https://reactjs.org/>
- [11] Angular.js , “Web developer’s platform”, 2010. Available: <https://angular.io/>
- [12] Vue.js “Progressive JavaScript Framework”, 2014. Available: <https://vuejs.org/>
- [13] Express.js , “Web framework for Node.js”, 2017. Available: <https://expressjs.com/>
- [14] Spring Boot, “Java Platform”, 2022. Available: <https://spring.io/projects/spring-boot>
- [15] Flask, “Python Web Framework”, 2010. Available: <https://flask.palletsprojects.com/en/2.1.x/>
- [16] FreeRTOS, “Real-time operating system for microcontrollers”, 2022. Available: <https://www.freertos.org/>
- [17] Zephyr, “Zephyr RTOS”, 2022. Available: <https://www.zephyrproject.org/>
- [18] LVGL , “Graphics Library”, 2022. Available: <https://lvgl.io/>

Application Note

- [19] STMicroelectronics, “LSM6DSOX: Finite State Machine”, AN5273, 2019.

[20] Analog Devices, Miguel Usach, “SPI Interface”, AN-1248 Application Note, 2013.

[21] NXP Jean-Marc Irazabal and Steve Blozis, “I2C MANUAL”, AN10216-01.

Articles

[22] Mdn web docs, “An overview of HTTP”, May 20, 2022.

[23] Mdn web docs, “REST” , May 2, 2022.

[24] Mdn web docs, “Using HTTP cookies”, Apr 27, 2022.

[25] Cloudflare, “What is HTTPS?”.

[26] auth0 Dan Arias, “Adding Salt to Hashing: A Better Way to Store Passwords”, February 25, 2021.

Εικόνες

[27] MVC pattern

[28] Structure of RTOS

[29] ESP-32 Pins

[30] ESP-32 Architecture

Οδηγός Λογισμικού

Η διαδικτυακή εφαρμογή: <https://ptuxiakiangstel.herokuapp.com/>

Δοκιμαστικά στοιχεία για σύνδεση ως ασθενής: Email: patient@patient.com , Password: 12345

Δοκιμαστικά στοιχεία για σύνδεση ως γιατρός Email: doc@doc.com, Password: 12345