



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**ΔΙΑΔΙΚΤΥΑΚΗ ΠΛΑΤΦΟΡΜΑ ΓΙΑ ΕΞ' ΑΠΟΣΤΑΣΕΩΣ  
ΥΠΟΣΤΗΡΙΞΗ ΑΣΘΕΝΩΝ ΚΑΙ ΗΛΙΚΙΩΜΕΝΩΝ  
ΜΕ ΧΡΗΣΗ RASPBERRY**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

ΣΤΑΜΑΤΙΟΣ-ΠΑΝΑΓΙΩΤΗΣ Ν. ΚΟΡΡΕΣ

**Επιβλέπων :** Τσανάκας Παναγιώτης  
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2017





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

## ΔΙΑΔΙΚΤΥΑΚΗ ΠΛΑΤΦΟΡΜΑ ΓΙΑ ΕΞ' ΑΠΟΣΤΑΣΕΩΣ ΥΠΟΣΤΗΡΙΞΗ ΑΣΘΕΝΩΝ ΚΑΙ ΗΛΙΚΙΩΜΕΝΩΝ ΜΕ ΧΡΗΣΗ RASPBERRY

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΤΑΜΑΤΙΟΣ-ΠΑΝΑΓΙΩΤΗΣ Γ. ΚΟΡΡΕΣ

Επιβλέπων : **Τσανάκας Παναγιώτης**  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25 Σεπτεμβρίου 2017.

.....  
Παναγιώτης Τσανάκας  
Καθηγητής Ε.Μ.Π.

.....  
Κιαμάλ Πεκμετζή  
Καθηγητής Ε.Μ.Π.

.....  
Δημήτριος Κουτσούρης  
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2017

.....  
Σταμάτιος-Παναγιώτης Γ. Κορρές

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

Copyright © Σταμάτιος-Παναγιώτης Ν. Κορρές, 2017.  
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

## Περίληψη

Η υψηλού επιπέδου υγειονομική περίθαλψη αποτελεί σημαντική ανάγκη για τον ανθρώπινο πληθυσμό. Παρόλο που ανάπτυξη της Ιατρικής τις τελευταίες δεκαετίες είναι ραγδαία, το πρόβλημα της επαρκούς και αποτελεσματικής περίθαλψης παραμένει. Οι κοινωνικές ομάδες που παρουσιάζουν τη μεγαλύτερη δυσκολία πρόσβασης σε αυτές τις υπηρεσίες είναι οι ηλικιωμένοι άνθρωποι, οι κάτοικοι απομακρυσμένων περιοχών και νησιών καθώς και οι άνθρωποι χαμηλότερου εισοδήματος. Η πρόοδος των κλάδων των Τηλεπικοινωνιών και της Πληροφορικής παρέχει μία πολύ δραστική λύση στα παραπάνω προβλήματα. Μία από τις συναφείς, με τα θέματα αυτά, εφαρμογές τους είναι η Τηλεϊατρική. Η τελευταία έχει τη δυνατότητα να επιτρέπει τη διαθεσιμότητα των σχετικών υπηρεσιών ανεξαρτήτως των γεωγραφικών αποστάσεων, να εκμηδενίζει τον χρόνο επικοινωνίας, να συμβάλλει στην έγκυρη πρόληψη και αντιμετώπιση έκτακτων καταστάσεων μέσω της απομακρυσμένης και συστηματικής παρακολούθησης ασθενών, αλλά και να μειώνει δραστικά τα έξοδα για όλα τα παραπάνω.

Στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός διαδικτυακού συστήματος το οποίο θα παρέχει απομακρυσμένη παρακολούθηση του ιστορικού ιατρικών βιοσημάτων των ασθενών, την προβολή των μετρήσεων σε διαδραστικά διαγράμματα, την ασύρματη και αυτόματη αποθήκευση μετρήσεων, την υπενθύμιση δραστηριοτήτων ή/και λήψης φαρμάκων, καθώς και ένα ολοκληρωμένο κοινωνικό δίκτυο για εύκολη επικοινωνία του ασθενή με τρίτους σε πραγματικό χρόνο, τόσο με εικόνα – βίντεο όσο και με ανταλλαγή γραπτών μηνυμάτων. Οι ασθενείς θα μπορούν να χρησιμοποιούν τις υπηρεσίες μέσω ενός συστήματος Raspberry-Pi, έναν φορητό υπολογιστή εξοπλισμένο με τις απαραίτητες περιφερειακές συσκευές. Ιδιαίτερη έμφαση δόθηκε στην ευκολία χρήσης ολόκληρου του συστήματος, ώστε να μην απαιτείται κάποια προηγούμενη γνώση από τους χρήστες. Όσον αφορά τις τεχνολογίες που χρησιμοποιήθηκαν, ο κύριος και οι επιμέρους εξυπηρετητές είναι υλοποιημένοι με nodejs, οι βάσεις του συστήματος με mongoDB, ενώ για την διεπαφή του χρήστη - συστήματος έγινε χρήση του πλαισίου AngularJS .

### Λέξεις Κλειδιά :

Τηλεϊατρική, Απομακρυσμένη παρακολούθηση, mongoDb, HTML5, JavaScript, Nodejs, WebRTC, Restful Architecture, Εφαρμογή μονής σελίδας, AngularJS, Bluetooth, Raspberry, Κοινωνικό Δίκτυο, OAuth2orize, Web server.

## **Abstract**

The need of high level medical care is very crucial for people. Although the progress of medicine is rapid the last decades, the problem of sufficient and efficient health care still remains. The social groups that have the most difficulty accessing those services are the elderly, those who live at remotes areas or islands and also those who have lower incomes. The development of Telecommunications and Computer Science can offer a very drastic solution to the above problems. Their main application is Telehealth. Telehealth is able to provide accessibility, which is independent of the geographic distances, significantly reduces the communication time, contributes to punctual prevention and confrontation of emergency situations through constant and remote monitoring and also is very cost-efficient.

The scope of this thesis is the design and implementation of a web based platform which provides remote monitoring of health records and biosignals, projection of those biosignals in interactive diagrams, wireless and automated biosignals storage, reminders for activities or/and medication and also an integrated social network which eases the user's real-time communication, allowing not only video calls but also written text. The patients of the platform will access it through a Raspberry Pi, which is a complete portable computer equipped with the necessary peripherals. Great emphasis was put on the ease of use, so no previous knowledge is required from users. Regarding the technology used, the Servers of the system are implemented with Nodejs, the database used is mongoDb and AngularJS was deployed as the front-end framework.

### Keywords:

Telehealth, Remote monitoring, mongoDb, HTML5, JavaScript, Nodejs, WebRTC, Restful Architecture, Single Page Application(SPA), AngularJS, Bluetooth, Raspberry, Social Network, Oauth2orize, Web server.

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Π. Τσανάκα για τις συμβουλές και την παρότρυνση πάνω στο συγκεκριμένο αντικείμενο, καθώς και την εμπιστοσύνη που έδειξε προς το πρόσωπό μου αναθέτοντάς μου την παρούσα διπλωματική εργασία. Επίσης θα ήθελα να ευχαριστήσω το μεταδιδακτορικό ερευνητή του Ε.Π.Μ. κ. Ανδρέα Μενύχτα ο οποίος με την εμπειρία, την καθοδήγηση και την υπομονή του, βοήθησε να ολοκληρωθεί αυτή η διπλωματική εργασία. Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συμπαράσταση και την υπομονή της καθ' όλη τη διάρκεια των σπουδών μου στο Ε.Μ.Π.

## Table of Contents

1	Εισαγωγή.....	12
1.1	Σκοπός της Διπλωματικής.....	12
1.2	Οργάνωση Κειμένου.....	13
2	Θεωρητικό και Τεχνολογικό Υπόβαθρο – Βασικές έννοιες.....	14
2.1	Θεωρητικό υπόβαθρο – Βασικές έννοιες.....	14
2.1.1.1	Ορισμός των Βιοσημάτων και η Χρησιμότητα τους.....	14
2.1.1.2	Ιστορικά Στοιχεία.....	15
2.1.2	Η καρδιά και ο Καρδιακός Ρυθμός.....	16
2.1.3	Το Οξυγόνο και η Οξυγόνωση του Αίματος.....	16
2.1.4	Παλμικό Οξύμετρο, και οι Φυσικές Παράμετροι που Μετρά, Κορεσμός σε Οξυγόνο .....	17
2.2	Ερευνητικό Υπόβαθρο.....	23
2.2	Τεχνολογικό Υπόβαθρο – Βασικές Έννοιες.....	24
2.2.1	Φυλλομετρητές Ιστού (Web Browsers).....	24
2.2.2	Web Applications ως Ολοκληρωμένες Εφαρμογές.....	26
2.2.3	Web Browser Συμβατότητα.....	28
2.2.4	CORS.....	28
2.2.5	JavaScript.....	28
2.2.5.1	Δυναμικό Σύστημα Τύπων και Ταυτοχρονισμός.....	29
2.2.5.2	Ο Βρόχος Συμβάντων (event loop).....	29
2.2.5.3	Run-to-Completion Logic.....	30
2.2.6	Single Page Applications and Frameworks.....	31
2.2.6.1	Τα Προγράμματα Πελάτες Γίνονται πιο Σύνθετα.....	31
2.2.6.2	AJAX (Asynchronous JavaScript and XML).....	31
2.2.6.3	SPA και το Μοντέλο MVC.....	32
2.2.8	AngularJS ως Framework.....	34
2.2.9	WebRTC.....	34
2.2.9.1	Σύντομη Ιστορική Προσέγγιση.....	34
2.2.9.2	Η Τεχνολογία WebRTC.....	35
2.2.9.3	(Σηματοδοσία) Signaling.....	35
2.2.10	Node.js.....	36
2.2.11	Διακομιστής Μεσολάβησης και Πρωτόκολλο HTTPS.....	38
	.....	38
2.2.11.1	Πρωτόκολλο HTTPS.....	38



2.2.11.2 Αντίστροφος Διακομιστής Μεσολάβησης (Reverse Proxy).....	39
2.2.11.3 Εξυπηρετητής Apache ως Reverse Proxy Server.....	39
2.2.12 Υπηρεσίες Rest (Representational State Transfer).....	40
2.2.13 JSON Objects.....	41
2.2.14 MongoDB.....	42
2.2.15 OAuth 2.0 και Access Tokens.....	42
2.2.15.1 Ροή του Πρωτοκόλλου.....	43
2.2.15.2 Authorization Grant.....	44
2.2.15.3 Σύμβολο Πιστοποίησης (Access token).....	44
2.2.15.4 Resource Owner Password Credential.....	44
2.2.16 Raspberry-Pi.....	46
2.2.17 Bluetooth Low Energy (Ble).....	47
3 Ανάλυση και Σχεδιασμός του Συστήματος.....	48
3.1 Γενική Περιγραφή.....	48
3.2 Απαιτήσεις.....	49
3.2.1 Λειτουργικές Απαιτήσεις.....	49
3.2.2 Μη Λειτουργικές Απαιτήσεις.....	51
3.3 Οι Χρήστες του συστήματος.....	51
3.4 Αρχιτεκτονική.....	52
3.4.1 Κεντρική Υποδομή και οι Υπηρεσίες της.....	52
3.4.1.1 Δημιουργία λογαριασμού – Σύνδεση στο σύστημα.....	52
3.4.1.2 Αιτήματα φιλίας (Friend Requests) και φιλίες (Friendships).....	52
3.4.1.3 Πολλαπλοί Χρήστες Συνδεδεμένοι σε Ίδιο Λογαριασμό.....	53
3.4.1.4 Online Χρήστες.....	53
3.4.1.5 Ασύγχρονη Ενημέρωση του Συστήματος.....	54
3.4.1.6 Μηνύματα Κειμένου (Text Messages).....	54
3.4.1.7 Σηματοδότηση (Signaling) και WebRTC.....	54
3.4.1.8 Μεταφορά (Αποστολή) Βιοσημάτων.....	58
3.4.2 Υποδομή του Raspberry.....	60
3.4.2.1 Σύνδεση στο Δίκτυο.....	60
3.4.2.2 Υπενθυμίσεις.....	61
3.4.2.3 Βιοσήματα.....	61
3.4.2.3.1 Εισαγωγή βιοσημάτων στο Σύστημα.....	61
3.4.2.3.2 Προβολή και επεξεργασία βιοσημάτων.....	62
3.4.2.3.3 Αποστολή Βιοσημάτων.....	62

3.4.2.4 Ρυθμίσεις Λογαριασμού.....	63
3.4.3 Επικοινωνία Μεταξύ των Υποσυστημάτων.....	64
4 Υλοποίηση του Συστήματος.....	65
4.1 SPA Layout.....	66
4.1.2 SPA Layout των Χρηστών Ασθενών.....	67
4.2 Ο Κεντρικός Εξυπηρετητής (Cloud Server).....	67
4.2.1 Τρόπος επικοινωνίας Μεταξύ SPA και των Servers.....	67
4.2.1.1 AJAX – HTTP Αιτήματα.....	68
4.2.1.2 Websockets.....	72
4.3 Υλοποίηση των Απαιτήσεων Απλών Χρηστών του Συστήματος.....	72
4.3.1 Χρήστες και Σύνδεση στο Σύστημα.....	72
4.3.2 Ασύγχρονη Ενημέρωση του Συστήματος και Angular Services.....	73
4.3.3 Αιτήματα φιλίας.....	74
4.3.4 Αυτόματη Συμπλήρωση Ονομάτων Φίλων.....	75
4.3.5 Φίλοι και Κατάσταση Σύνδεσης των Φίλων ενός Χρήστη (State of Users – Online / Offline).....	76
4.3.6 Μηνύματα Γραπτά Κειμένου.....	78
4.3.7 Σηματοδότηση – Singalling και WebRTC (Σχεδιαστικές Αποφάσεις ).....	79
4.3.8 Μεταφορά και Απεικόνιση Βιοσημάτων.....	82
4.4 Υλοποίηση των απαιτήσεων χρηστών – ασθενών του.....	83
συστήματος.....	83
4.4.1 Σύνδεση στο Δίκτυο.....	83
4.4.2 Υπενθυμίσεις.....	84
4.4.3 Διαχείριση Βιοσημάτων.....	88
5. Επίλογος - Συμπεράσματα.....	94
5.1 Το Τελικό Σύστημα.....	94
5.2 Παρατηρήσεις – Προσφερόμενες Υπηρεσίες.....	94
5.3 Θέματα για Περαιτέρω Διερεύνηση.....	95
Βιβλιογραφία.....	97



## Εικόνες - Σχήματα

Σχήμα 1: Η γενική Αρχιτεκτονική του Συστήματος .....	13
Σχήμα 2: Λήψη Βιοσήματος (a) μέσω απλής παρατήρησης (b) μέσω τοποθέτησης κάποιου αισθητήρα.....	15
Σχήμα 3: Ψηλάφηση ενός ασθενή.....	16
Σχήμα 4: Καμπύλη της Οξυγόνωσης του αίματος.....	19
Σχήμα 5: Παλμικό Οξύμετρο Τοποθετημένο στον Καρπό.....	20
Σχήμα 6: Ποσοστά χρήσης των web browsers παγκοσμίως.....	27
Σχήμα 7: Απόδοση Διάφορων Web Browsers.....	28
Σχήμα 8: Γραφική μορφή ενός απλού HTML κειμένου.....	29
Σχήμα 9: Document Object Model μιας web σελίδας.....	29
Σχήμα 10: Οι τεχνολογίες που σε συνδυασμό δημιουργούν εφαρμογές δικτύου.....	30
Σχήμα 11: Event loop της JavaScript.....	32
Σχήμα 12: Διαφορές μεταξύ της υπάρχουσας λογικής του διαδικτύου και της SPA.....	35
Σχήμα 13: Διαφορές μεταξύ της υπάρχουσας λογικής του διαδικτύου και της SPA.....	36
Σχήμα 14: Η διαδικασία Σηματοδότησης (Signaling) για την εγκαθίδρυση επικοινωνίας πραγματικού χρόνου.....	39
Σχήμα 15: Η λογική εξυπηρέτησης πολλών ταυτόχρονων αιτημάτων χωρίς παραλληλισμό.....	40
Σχήμα 16: Διακομιστής Μεσολάβησης.....	41
Σχήμα 17: Αντίστροφος διακομιστής μεσολάβησης.....	43
Σχήμα 18: RESTful αρχιτεκτονική μιας εφαρμογής.....	45
Σχήμα 19: Αναπαράσταση δεδομένων με σήμανση JSON.....	46
Σχήμα 20: Ροή Πρωτοκόλλου του OAuth 2.0.....	47
Σχήμα 21: Πιστοποίηση τύπου Resource Owner Password Credential.....	49
Σχήμα 22: Raspberry Pi.....	50
Σχήμα 23: Το Προτεινόμενο Σύστημα.....	52
Σχήμα 24: Διαφορετικές τοπολογίες ενός δικτύου.....	53
Σχήμα 25: Ροή ελέγχου για πραγματοποίηση κλήσης.....	60
Σχήμα 26: Ροή ελέγχου τερματισμού κλήσης.....	61
Σχήμα 27: Το Υλοποιημένο Σύστημα.....	68
Σχήμα 28: Η διεπαφή χρήστη (UI) στην εισαγωγική σελίδα.....	76
Σχήμα 29: Εισερχόμενα και εξερχόμενα αίτηματα φιλίας.....	78
Σχήμα 30: Αναζήτηση και Αυτόματη Συμπλήρωση κάποιου χρήστη που περιέχει το γράμμα 's'.....	79
Σχήμα 31: Επιλογή μηνύματος και αναδυόμενο παράθυρο διαγραφής.....	82
Σχήμα 32: Εισερχόμενη κλήση απο φίλο του χρήστη.....	83
Σχήμα 33: Ένας χρήστης του συστήματος ξεκινά μία κλήση.....	85
Σχήμα 34: Δύο χρήστες του συστήματος βρίσκονται σε κλήση.....	85
Σχήμα 35: Τα βιοσήματα που έχει λάβει ένας χρήστης από έναν ασθενή του συστήματος.....	86
Σχήμα 36: Πρώτη εγγραφή του ασθενή για την σύνδεση στο σύστημα.....	87
Σχήμα 37: User Interface για τις υπενθυμίσεις.....	88
Σχήμα 38: Ενημέρωση του χρήστη για μια υπενθύμιση.....	89
Σχήμα 39: Τροποποίηση υπενθύμισης.....	89
Σχήμα 40: Η συλλογή υπενθυμίσεων του ασθενή.....	90

Σχήμα 41: Απεικόνιση και επεξεργασία βιοσημάτων.....	91
Σχήμα 42: Χειροκίνητη προσθήκη βιοσήματος στο σύστημα.....	92
Σχήμα 43: Αποθήκευση μέτρησης μέσω οξυμέτρου και ενημέρωση αποθήκευσης.....	93
Σχήμα 44: Αποστολή και απεικόνιση βιοσημάτων σε πραγματικό χρόνο.....	94
Σχήμα 45: Αίτημα λήψης βιοσημάτων από μη-εξουσιοδοτημένο χρήστη.....	94
Σχήμα 46: Ρυθμίσεις λογαριασμού του απλού χρήστη.....	95
Σχήμα 47: Διαγραφή χρήστη από την λίστα εξουσιοδοτημένων χρηστών.....	96

## Πίνακες

Πίνακας 1: Φυσιολογικές τιμές κορεσμού σε οξυγόνο στο αίμα .....	18
--	----

# 1 Εισαγωγή

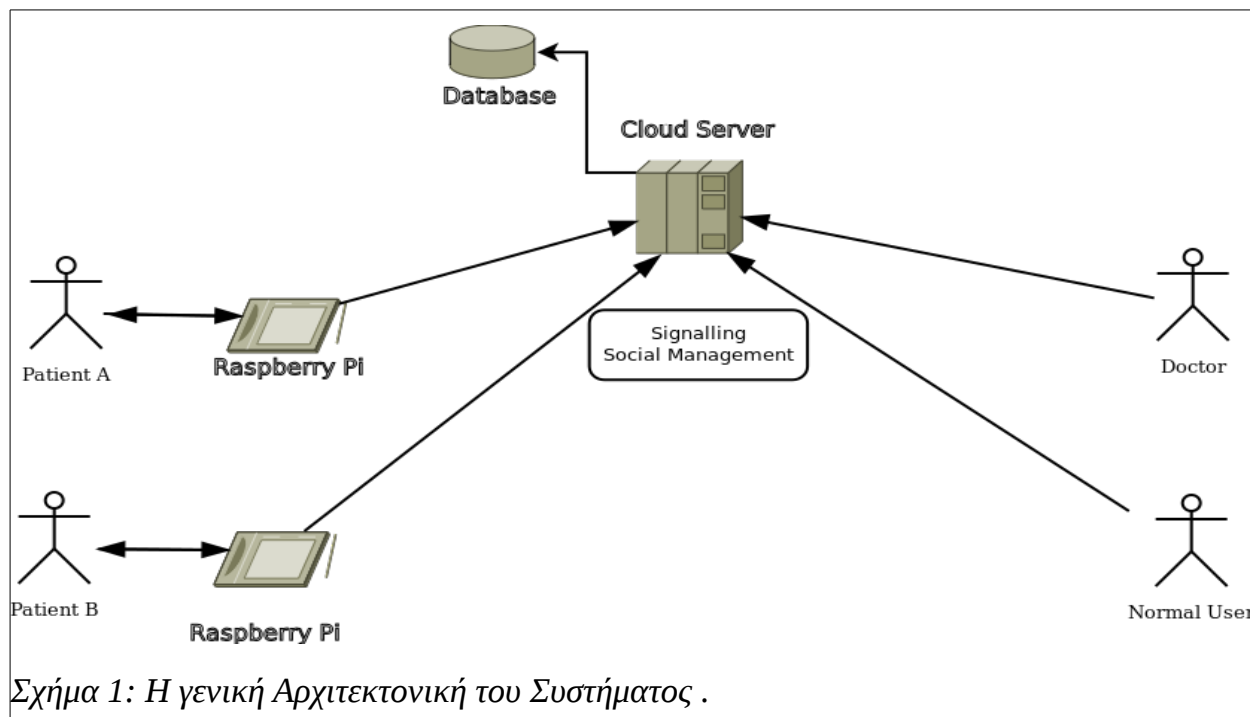
## 1.1 Σκοπός της Διπλωματικής

Η ραγδαία εξέλιξη της τεχνολογίας φαίνεται να έχει επηρεάσει τον τρόπο ζωής των ανθρώπων ριζικά και αμετάκλητα. Τις τελευταίες δεκαετίες η τεχνολογία έχει κληθεί να προσφέρει λύση σε μία από τις βασικότερες ανάγκες της ανθρωπίνης ύπαρξης, την υγεία. Το σύστημα υγείας του κάθε κράτους είναι υπεύθυνο για την παροχή υψηλής ποιότητας υγειονομικής περίθαλψης σε όλους τους πολίτες και ειδικότερα στις ευάλωτες ομάδες του πληθυσμού. Όμως οι σύγχρονες δημογραφικές, κοινωνικό-οικονομικές και γεωγραφικές ανάγκες καθιστούν δύσκολη την ισόνομη και εύκολη πρόσβαση όλων των ομάδων σε αυτή. Μια δυναμική και αξιόπιστη απάντηση προσφέρει η τεχνολογία μέσω της Τηλεϊατρικής. Είναι ένας τομέας που επιτρέπει την ισόνομη και εύκολη πρόσβαση ανθρώπων στις υγειονομικές υπηρεσίες, έχοντας καταφέρει να προσφέρει απομακρυσμένη και συστηματική παρακολούθηση ασθενών με σκοπό την αντιμετώπιση επειγόντων περιστατικών που μέχρι πρότινος ήταν αδύνατη για ορισμένες κοινωνικές ομάδες.

Ένα δεύτερο πρόβλημα που έχει να αντιμετωπίσει η Τηλεϊατρική είναι η εξοικείωση των ανθρώπων με αυτήν τη νέα πραγματικότητα. Οι ρυθμοί εξέλιξης όμως της τεχνολογίας είναι ταχείς και είναι δύσκολο κάποιος να συμβαδίσει μαζί τους. Μία τεχνική είναι η δημιουργία ολοκληρωμένων πλατφορμών, οι οποίες θα είναι φιλικές απέναντι στον χρήστη και θα προσφέρουν όσες υπηρεσίες θα χρειαστεί πιθανώς ο χρήστης, στο βαθμό φυσικά που αυτό είναι εφικτό.

Σκοπός της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός τέτοιου συστήματος, το οποίο θα προσφέρει μια ολοκληρωμένη εμπειρία στον χρήστη-ασθενή και θα βασίζεται στις τελευταίες τεχνολογίες του παγκόσμιου ιστού (HTML5, CCS3, JavaScript, Nodejs, MongoDB). Ως περιβάλλον εκτέλεσης του παραπάνω συστήματος επιλέχτηκε ο browser, στον οποίο θα τρέχει η εφαρμογή του χρήστη-ασθενή, καθώς και των απλών χρηστών που θα συνδέονται στο σύστημα (γιατροί, φίλοι, συγγενείς). Επιπλέον, στους ασθενείς θα παραχωρείται ένα Raspberry Pi, ένας μικρός δηλαδή υπολογιστής, μέσω του οποίου θα μπορούν να αποθηκεύουν χειροκίνητα ή αυτόματα μετρήσεις με τη βοήθεια ενός ασύρματου οξυμέτρου, να παρακολουθούν το ιστορικό των μετρήσεών τους σε κατάλληλα διαδραστικά διαγράμματα, να ρυθμίζουν υπενθυμίσεις καθημερινών δραστηριοτήτων και λήψης φαρμάκων, καθώς και να συνδέονται στο κοινωνικό δίκτυο του συστήματος. Το τελευταίο παρέχει εύκολη επικοινωνία, είτε με βίντεο-εικόνα είτε με γραπτά μηνύματα, καθώς και αποστολή των μετρήσεων σε άλλους χρήστες συνδεδεμένους στο σύστημα. Η ασφάλεια και η ιδιωτικότητα είναι επίσης σημαντικά ζητήματα, καθώς τα ιατρικά δεδομένα είναι εξ' ορισμού ευαίσθητης φύσεως και θα πρέπει να είναι διαθέσιμα μόνο σε εξουσιοδοτημένους χρήστες, πράγμα το οποίο σέβονται οι κανόνες και η πολιτική του παραπάνω συστήματος.

Το Σχήμα 1 αποτελεί μια προσπάθεια για μια γενική και αφαιρετική οπτικοποίηση του συστήματος που υλοποιήθηκε, στο οποίο φαίνονται τόσο τα μέρη από τα οποία αποτελείται καθώς και από τις κατηγορίες των χρηστών στους οποίους αναφέρεται.



## 1.2 Οργάνωση Κειμένου

Η διπλωματική εργασία είναι οργανωμένη σε πέντε κεφάλαια. Η δομή τους είναι η εξής:

- ❖ Στο **2ο Κεφάλαιο** παρουσιάζονται οι σημαντικότερες έννοιες, θεωρητικές αλλά και τεχνικές, οι οποίες είναι άρρηκτα συνδεδεμένες με το έργο και θα βοηθήσουν στην διαμόρφωση μιας σφαιρικής εικόνας. Επιπλέον παρουσιάζει διαφορετικές προσπάθειες που έχουν γίνει στο παρελθόν για συναφή αντικείμενα.
- ❖ Στο **3ο Κεφάλαιο** παρουσιάζεται η αρχιτεκτονική του συστήματος, οι απαιτήσεις του, τα μέρη που το αποτελούν καθώς και σημαντικότερες περιπτώσεις χρήσης του συστήματος (Use Cases)
- ❖ Στο **4ο Κεφάλαιο** παρουσιάζονται οι λεπτομέρειες της υλοποίησης καθώς και τα πιο σημαντικά ζητήματα, όπως η λειτουργία του εξυπηρετητή (server), του Raspberry-Pi, των μεθόδων επικοινωνίας, διαφόρων τεχνικών που χρησιμοποιήθηκαν, η διεπαφή (User Interface ) της εφαρμογής καθώς και μερικά μέρη του κώδικα.
- ❖ Στο **5ο Κεφάλαιο** παρουσιάζεται μια αποτίμηση της εργασίας ως σύνολο και πιθανές επεκτάσεις – βελτιώσεις της λειτουργικότητας του συστήματος.
- ❖ Στο **6ο Κεφάλαιο** παρατίθεται η βιβλιογραφία που χρησιμοποιήθηκε κατά τη συγγραφή της διπλωματικής εργασίας.

## 2 Θεωρητικό και Τεχνολογικό Υπόβαθρο – Βασικές έννοιες

Σκοπός του παρόντος κεφαλαίου είναι να εισαγάγει τον αναγνώστη στις βασικότερες έννοιες οι οποίες χρησιμοποιούνται σε αυτή τη διπλωματική εργασία. Καθώς η παρούσα διπλωματική εργασία συνδυάζει δύο διαφορετικούς επιστημονικούς τομείς (Ιατρική και Πληροφορική), το κεφάλαιο χωρίστηκε σε τρία κεφάλαια. Το πρώτο αναφέρεται σε θεωρητικές κυρίως γνώσεις σχετικά με ιατρικά θέματα, το δεύτερο αναφέρει άλλες ερευνητικές προσπάθειες που έχουν γίνει στον τομέα αυτό, καθώς και τα πορίσματά τους, ενώ το τρίτο αναλύει τις πιο βασικές τεχνολογίες που χρησιμοποιήθηκαν για την ολοκλήρωση της διπλωματικής εργασίας.

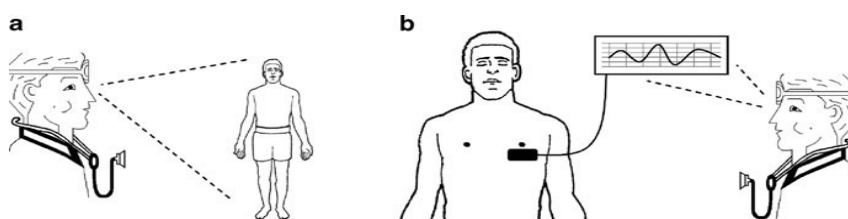
### 2.1 Θεωρητικό υπόβαθρο – Βασικές έννοιες

Τα δύο φυσικά μεγέθη, τα οποία είναι σε θέση να παρακολουθεί και να αποθηκεύει ο ασθενής στο Raspberry, είναι ο καρδιακός ρυθμός (heart rate) και ο περιφερειακός κορεσμός σε οξυγόνο του αίματος (Peripheral Oxygen Saturation – SpO<sub>2</sub>). Η παρακολούθηση θα γίνεται μέσω των αντίστοιχων βιοσημάτων των δύο αυτών μεγεθών. Έτσι κρίνεται απαραίτητο να οριστεί ο όρος βιοσήμα, ενώ παράλληλα να παρουσιαστούν οι βασικοί ορισμοί των δύο παραπάνω μεγεθών καθώς και οι τρόποι συλλογής πληροφοριών μέσω οργάνων.

#### 2.1.1 Βιοσήματα

##### 2.1.1.1 Ορισμός των Βιοσημάτων και η Χρησιμότητά τους

Στο πλαίσιο της Βιοϊατρικής, ένα βιοσήμα μπορεί να οριστεί ως μια περιγραφή ενός φυσιολογικού φαινομένου, ανεξάρτητα της φύσης της περιγραφής αυτής. Από τη στιγμή που υπάρχουν αμέτρητοι φυσικοί μηχανισμοί, το πλήθος των βιοσημάτων είναι και αυτό αντίστοιχα πολύ μεγάλο. Το εύρος των βιοσημάτων είναι τέτοιο, που εκτείνεται από την οπτική επιθεώρηση ενός ασθενή έως και την παρακολούθηση του ανθρώπινου σώματος μέσω αισθητήρων, για παράδειγμα το ηλεκτρομυογράφημα. Αυτή η μεγάλη ποικιλία έχει οδηγήσει σε μια πολύ μεγάλη κατηγοριοποίηση των βιοσημάτων, όπως κατηγοριοποίηση σε μόνιμα ή παροδικά, δυναμικά ή στατικά, ανάλογα με την προέλευσή τους κ.α.



Σχήμα 2: Λήψη Βιοσήματος (a) μέσω απλής παρατήρησης (b) μέσω τοποθέτησης κάποιου αισθητήρα



Το Σχήμα 2 δείχνει ένα παράδειγμα δημιουργίας ενός ακουστικού βιοσήματος, το οποίο χρησιμοποιείται για την εκτίμηση καρδιοπνευμονικών παθήσεων ή της μέτρησης των παλμών της καρδιάς μέσω ενός ακουστικού σήματος, το οποίο στην συνέχεια προβάλλεται σε διάγραμμα. Αυτή η περιγραφή επομένως μέσω της παρατήρησης μπορεί να μετατραπεί σε ένα μαθηματικό μοντέλο το οποίο είναι πιο εύκολο να αναλυθεί και να μελετηθεί. Αυτό σημαίνει ότι τα βιοσήματα μπορούν να χρησιμοποιηθούν τόσο σε διάγνωση όσο και σε θεραπεία των ασθενών, αφού μπορούν να προβλέψουν παθήσεις, ή να βρουν την πηγή παθήσεων, μελετώντας πλέον τις τιμές που λαμβάνει ο ασθενής με τα μοντέλα που έχουν δημιουργηθεί από προηγούμενες αναλύσεις.

### 2.1.1.2 Ιστορικά Στοιχεία

Η παρακολούθηση και ανάλυση των ανθρώπινων βιοσημάτων αναπτύχθηκε με την πάροδο πολλών αιώνων. Αρχικά ξεκίνησε ως απλή οπτική εξέταση χωρίς τη χρήση κάποιου οργάνου, στη συνέχεια έκανε χρήση τεχνικών εργαλείων για την καταγραφή των σημάτων. Σήμερα πλέον τα βιοσήματα παρακολουθούνται σχεδόν αποκλειστικά με συσκευές. Οι πρώτες διαγνώσεις στηρίζονταν στα "λεγόμενα" του ασθενή, ο οποίος έπρεπε να περιγράψει το πρόβλημα ή το πόνο που ένιωθε. Οι διαγνώσεις αυτές στηρίζονταν επίσης στην παρατήρηση του ασθενή μέσω της όρασης, στην ψηλάφηση του σώματος, π.χ. ασκώντας κάποια δύναμη, στην επίκρουση του σώματος σε διάφορα σημεία, καθώς και σε ακουστικές παρατηρήσεις.



Σχήμα 3: Ψηλάφηση ενός ασθενή

Σημαντική πρόοδος στη διάγνωση έφερε η εφεύρεση του στηθοσκοπίου, το οποίο μάλιστα θεωρείται και το πρώτο σύμβολο της Ιατρικής. Το στηθοσκόπιο μπορούσε να μεγεθύνει ήχους του ανθρώπινου σώματος και να τους μεταφέρει στο αυτί του γιατρού με σχετικά μικρές απώλειες. Το στηθοσκόπιο παρέμεινε από τότε το βασικό μέσο για τις διαγνώσεις. Παράλληλα άρχισαν να δημιουργούνται νέες ενστάσεις στις μέχρι τότε υπάρχουσες μεθόδους σχετικά με την αντικειμενικότητα των αποτελεσμάτων, οι οποίες ήταν μέχρι τότε στην κρίση του εκάστοτε γιατρού. Αυτό δημιουργούσε πρόβλημα καθώς τα αποτελέσματα μιας διάγνωσης δεν μπορούσαν να ελεγχθούν από τρίτους. Έτσι δημιουργήθηκε η ανάγκη να χαρακτηριστούν και να κατηγοριοποιηθούν τα βιοσήματα. Ο χαρακτηρισμός των βιοσημάτων με την πάροδο των χρόνων πέρασε από τα εξής στάδια: λεκτική περιγραφή, μουσική περιγραφή και τέλος μέσω τεχνικών εργαλείων. Αν

και η μεταστροφή στην μουσική ως μέσου περιγραφής δεν έκανε τις περιγραφές πολύ αντικειμενικές, ο ρυθμός και η ένταση μιας μουσικής νότας μπορούσε να χρησιμοποιηθεί ώστε να κωδικοποιήσει τα βιοσήματα. Φυσικά η πιο επιτυχημένη προσέγγιση των βιοσημάτων έγινε με την βοήθεια τεχνικών εργαλείων, τα οποία εξάλειψαν την υποκειμενικότητα των διαγνώσεων, αφού πλέον τα μεγέθη μπορούν να μετρηθούν και να ποσοτικοποιηθούν. Επανάσταση αποτέλεσε και το γεγονός ότι δημιουργήθηκαν φορητά εργαλεία και όργανα τα οποία θεμελίωσαν και διέδωσαν την μέθοδο αυτή [37].

### **2.1.2 Η καρδιά και ο Καρδιακός Ρυθμός**

Η καρδιά είναι κοίλος μυς, ο οποίος, δίνοντας στο αίμα πίεση, το κάνει να κυκλοφορεί στο εσωτερικό των αρτηριών, με τέτοιο τρόπο, ώστε να φτάνει σε όλα τα όργανα. Είναι φυσική αντλία που παίρνει το αίμα από τις φλέβες, στις οποίες βρίσκεται υπό χαμηλή πίεση, και το στέλνει στις αρτηρίες με υψηλή πίεση. Η καρδιά αποτελείται από έναν ειδικό τύπο σκελετικού μυ, ο οποίος βρίσκεται μόνο σε αυτή και αποκαλείται καρδιακός μυς ή μυοκάρδιο. Η καρδιά, εν γένει, λειτουργεί ακατάπαυστα καθ' όλη τη διάρκεια της ζωής του οργανισμού. Ένα μέτρο της κατάστασης της καρδιάς είναι ο καρδιακός σφυγμός. Ο καρδιακός παλμός είναι η ταχύτητα του παλμού της καρδιάς εκφρασμένη ως ο αριθμός των συστολών της καρδιάς ανα λεπτό (beats per minute - bpm). Ο καρδιακός παλμός διαφέρει από άτομο σε άτομο, λόγω των διαφορετικών φυσικών αναγκών του κάθε οργανισμού, συμπεριλαμβανομένης της ανάγκης για την απορρόφηση οξυγόνου και την έκκριση διοξειδίου του άνθρακα. Ο φυσικός καρδιακός παλμός ενός ενήλικου ανθρώπου ο οποίος βρίσκεται σε ηρεμία, κυμαίνεται μεταξύ 60 και 100 παλμών ανά λεπτό.

Βραδυκαρδία είναι ο καρδιακός παλμός ενός ενήλικα σε ηρεμία ο οποίος βρίσκεται κάτω από 60 παλμούς το λεπτό, ενώ αντίθετα ταχυκαρδία ορίζεται αν βρίσκεται πάνω από 100 παλμούς το λεπτό.

### **2.1.3 Το Οξυγόνο και η Οξυγόνωση του Αίματος**

Το οξυγόνο είναι το σημαντικότερο στοιχείο για τη διατήρηση της ζωής. Η πρόσληψή του από την ατμόσφαιρα και η μεταφορά του στους ιστούς είναι ο κύριος σκοπός της λειτουργίας τριών συστημάτων: αναπνευστικού, κυκλοφορικού και αίματος. Στα κύτταρα των ιστών το οξυγόνο χρησιμοποιείται κυρίως μέσα στα μιτοχόνδρια για την παραγωγή ενέργειας και την κάλυψη των μεταβολικών αναγκών του οργανισμού. Το οξυγόνο μεταφέρεται στους ιστούς με δύο τρόπους: διαλυμένο στο αίμα και ενωμένο με την αιμοσφαιρίνη. Η ποσότητα του οξυγόνου που μεταφέρεται διαλυμένη είναι πολύ μικρή. Κατά συνέπεια, ο κύριος τρόπος μεταφοράς του οξυγόνου είναι δια μέσου της αιμοσφαιρίνης

Ο κορεσμός σε οξυγόνο του αίματος (oxygen saturation –  $spO_2$  ή  $saO_2$ ) ορίζεται ως το ποσοστό της αρτηριακής αιμοσφαιρίνης ( $HbO_2$ ) προς τη συνολική συγκέντρωση της αιμοσφαιρίνης στο αίμα και εκφράζεται ως ποσοστό %. Τα σύμβολα  $SpO_2$  (Peripheral Oxygen Saturation) και  $SaO_2$  (arterial oxygen saturation) εκφράζουν και τα δύο τον κορεσμό σε οξυγόνο του αίματος, όμως, έχουν προκύψει με διαφορετικό τρόπο. Το  $SpO_2$  προκύπτει με έμμεση (indirect) μέτρηση, με τη χρήση της παλμικής οξυμετρίας, για παράδειγμα από το δάκτυλο του χεριού, γι' αυτό και εκφράζει την

περιφερική τριχοειδή (peripheral capillary) κυκλοφορία. Από την άλλη πλευρά, το SaO<sub>2</sub> προκύπτει με απευθείας (direct) μέτρηση, με ανάλυση από δείγμα αίματος, για παράδειγμα με την Ανάλυση Αερίων Αρτηριακού Αίματος (AAAA – AGB – Arterial Blood Gases) από δείγμα που έχει ληφθεί από αρτηρία [3,4].

Η φυσιολογική ένδειξη για τον κορεσμό του οξυγόνου σε ασθενείς χωρίς καμία πνευμονική παθολογία είναι άνω του 95%. Ποσοστό 92% ή λιγότερο υποδηλώνει υποξαιμία όπως φαίνεται στο Σχήμα 1. Τα φάσματα απορρόφησης της κορεσμένης σε οξυγόνο αιμοσφαιρίνης (HbO<sub>2</sub>) και της φτωχής σε οξυγόνο αιμοσφαιρίνης (Hb) για κόκκινα και υπέρυθρα μήκη κύματος αντιστοιχούν σε υποξία (hypoxia, hypoxiation), παθολογική κατάσταση κατά την οποία ολόκληρο το σώμα (γενικευμένη υποξία) ή ένα μέρος του στερείται επαρκούς οξυγόνωσης. Τιμές κάτω του 89% υποδηλώνουν αναπνευστική ανεπάρκεια, που σημαίνει ότι, εάν δεν υπάρχει κάποιος αναστρέψιμος παράγοντας, ο ασθενής χρήζει συμπληρωματικής χορήγησης οξυγόνου [5].

**Πίνακας 1.** Φυσιολογικές τιμές κορεσμού σε οξυγόνο του αίματος

<b>Κατάσταση</b>	<b>Εύρος Τιμών (% SpO<sub>2</sub>)</b>
Φυσιολογική	> 95
Υποξαιμία	<= 92
Αναπνευστική Ανεπάρκεια	< 89

#### **2.1.4 Παλμικό Οξύμετρο, και οι Φυσικές Παράμετροι που Μετρά, Κορεσμός σε Οξυγόνο**

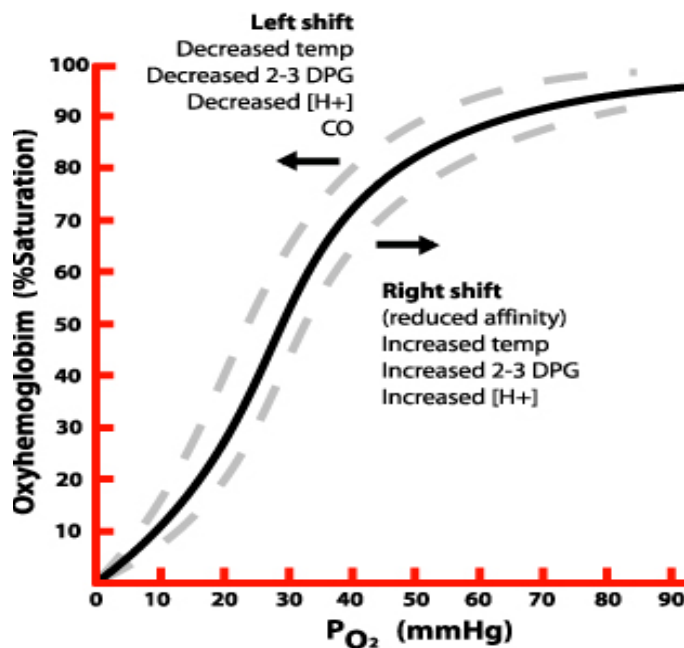
Στην παρούσα διπλωματική εργασία για τη πραγματοποίηση των μετρήσεων των απαραίτητων φυσιολογικών παραμέτρων χρησιμοποιήθηκε ως όργανο λήψης το παλμικό οξύμετρο. Στην ενότητα αυτή θα περιγραφεί το παλμικό οξύμετρο και οι τιμές που αυτό μετρά, καθώς χρησιμοποιείται ευρέως στην εργασία αυτή.

##### **Το Παλμικό Οξύμετρο**

Το παλμικό οξύμετρο (pulse oximeter) είναι το δεύτερο ιατρικό όργανο που χρησιμοποιήσαμε για τις μετρήσεις μας, με τη βοήθεια του οποίου μετρήθηκε ο κορεσμός σε οξυγόνο του περιφερικού αίματος (Peripheral Oxygen Saturation – SpO<sub>2</sub>), ο καρδιακός παλμός (heart rate – beats per minute – bpm), ο δείκτης διάχυσης (Perfusion Index – PI) σε πραγματικό χρόνο. Το παλμικό οξύμετρο είναι μία συσκευή που αποτελείται από έναν αισθητήρα, η λειτουργία του οποίου θα αναλυθεί παρακάτω, και η οποία επιτρέπει τη μη επεμβατική παρακολούθηση των επιπέδων του οξυγόνου στο αίμα, ή όπως λέγεται του κορεσμού σε οξυγόνο του αίματος (oxygen saturation – SO<sub>2</sub>). Το βασικό πλεονέκτημα των οπτικών αισθητήρων, όπως και στην περίπτωση του παλμικού οξυμέτρου, στις ιατρικές εφαρμογές είναι ότι έχουν εγγενή ασφάλεια καθώς δεν υπάρχει καμία ηλεκτρική επαφή μεταξύ του ασθενούς και του εξοπλισμού [7]. Η παλμική οξυμετρία παρέχει έναν μη επεμβατικό, ανώδυνο, εύκολα προσβάσιμο και φθηνό τρόπο της μέτρησης του οξυγόνου στο αίμα σε πραγματικό χρόνο.

Επιπλέον, τα περισσότερα οξύμετρα σήμερα επιτρέπουν την παρακολούθηση σε πραγματικό χρόνο (real-time) του καρδιακού παλμού όπως αυτά που χρησιμοποιήσαμε σε αυτή τη διπλωματική εργασία αυτή.

Στον πιο συνήθη τρόπο λειτουργίας του, εκείνον της μετάδοσης (transmissive), ο οποίος χρησιμοποιείται σε αυτή την εργασία, μία συσκευή αισθητήρα (sensor device) τοποθετείται σε κάποιο λεπτό μέρος του ανθρώπινου σώματος, συνήθως στο δάκτυλο ή το λοβό του αυτιού ή στην περίπτωση βρέφους στην πατούσα. Η συσκευή μεταδίδει φως δύο μηκών κύματος μέσω του σώματος σε έναν φωτοανιχνευτή (photodetector). Μετρώντας την αλλαγή στην απορρόφηση καθενός από τα δύο μήκη κύματος, προσδιορίζει πόσο απορροφήθηκαν λόγω του παλμικού αρτηριακού αίματος, αποκλείοντας το φλεβικό αίμα, το δέρμα, τα κόκαλα, τους μύες, το λίπος και στις περισσότερες περιπτώσεις το βερνίκι νυχιών.



Σχήμα 4: Καμπύλη της οξυγόνωσης του αίματος



*Σχήμα 5: Παλμικό Οξύμετρο  
Τοποθετημένο στον Καρπό`*

Εναλλακτικός τρόπος λειτουργίας του οξυμέτρου είναι αυτός της ανάκλασης (reflectance pulse oximetry), ο οποίος σε αντίθεση με τον μετάδοσης, δεν απαιτεί κάποιο λεπτό μέρος του σώματος και μπορεί να χρησιμοποιηθεί πιο καθολικά, στα πόδια, το μέτωπο, το στήθος, αλλά έχει κάποιους περιορισμούς. Σε περιπτώσεις μειωμένης φλεβικής επιστροφής της καρδιάς, όπως στη συγγενή κυανωτική καρδιοπάθεια, εκ γενετής ασθένεια που σχετίζεται με δυσχρωμία του δέρματος (μπλε χρώμα) λόγω της χαμηλής συγκέντρωσης οξυγόνου στο αίμα (congenital cyanotic heart disease) ή σε ασθενείς σε θέση Trendelenburg (δηλαδή σε κληνήρη στάση με κλίση προς τα πάνω), προκαλείται αγγειοδιαστολή και συγκέντρωση φλεβικού αίματος στο κεφάλι, η οποία μπορεί να προκαλέσει φλεβικούς και αρτηριακούς παλμούς στο μέτωπο και να οδηγήσει σε αμφίβολης αξιοπιστίας SpO<sub>2</sub> αποτελέσματα [6].

#### **- Ιστορικά Στοιχεία**

Τη δεκαετία του 1930 κατασκευάστηκαν οι πρώτες συσκευές για τη μέτρηση του κορεσμού του οξυγόνου στο ανθρώπινο αίμα οι οποίες δούλευαν με χρωματιστό φως. Αυτές οι συσκευές δεν μπορούσαν να διακρίνουν τη διαφορά μεταξύ αρτηριακού και φλεβικού (ή τριχοειδούς) αίματος. Στη συνέχεια, έγινε προσπάθεια για τη συμπερίληψη του φλεβικού και του τριχοειδούς αίματος χρησιμοποιώντας μία από τις δύο ακόλουθες μεθόδους:

- Η πρώτη αφορούσε στο μηδενισμό του οξυμέτρου με τη λήψη αναίμακτης μέτρησης από το λοβό του αυτιού όταν συμπιεζόταν από δύο δάχτυλα.
- Η δεύτερη μέθοδος, που αναπτύχθηκε κατά το Β' Παγκόσμιο Πόλεμο με σκοπό την κατανόηση της απώλειας συνείδησης που υφίσταντο οι πιλότοι της Βασιλικής Αεροπορίας κατά τη διάρκεια αερομαχιών, περιελάμβανε θέρμανση του αίματος στους 43°C.

Το πρόβλημα, όμως, και στις δύο μεθόδους ήταν ότι το φως που μεταδιδόταν στο αυτί ή το δάκτυλο, εξασθενούσε από το αρτηριακό, φλεβικό και τριχοειδές αίμα, όπως και

από το δέρμα του οποίου το χρώμα και άρα η απορρόφηση διαφέρει σε κάθε άνθρωπο, αλλά και από ιστούς όπως οι μύες, τα κόκκαλα και άλλα [6].

Το 1964 ένας χειρουργός, ο Robert Shaw, κατασκεύασε το πρώτο οξύμετρο αυτιού χρησιμοποιώντας οκτώ διαφορετικά μήκη κύματος. Η συσκευή βγήκε στο εμπόριο από την εταιρεία Hewlett-Packard. Δεν έτυχε ευρύτερης αποδοχής λόγω του μεγέθους και του κόστους της και η χρήση της περιορίστηκε σε πνευμονικές λειτουργίες και σε εργαστήρια μελέτης του ύπνου. Η παλμική οξυμετρία αναπτύχθηκε το 1972 από τους μηχανικούς Βιοιατρικής Takuo Aoyagi και Michio Kishi και από τον Nihon Kohden χρησιμοποιώντας την αναλογία απορρόφησης της κόκκινης προς την υπέρυθρη ακτινοβολία παλλόμενων εξαρτημάτων στην περιοχή μέτρησης. Η συσκευή εμπορευματοποιήθηκε το 1981 από την εταιρεία Biox και το 1983 από την Nellcor. Μέχρι το 1987 η χρήση της παλμικής οξυμετρίας είχε βρει τη θέση της στο πρότυπο της φροντίδας για τη χορήγηση της γενικής αναισθησίας στα αμερικανικά νοσοκομεία και η χρήση της σταδιακά εδραιώθηκε και στους υπόλοιπους τομείς των νοσοκομείων [7]. Οι συσκευές, όμως, που κατασκευάστηκαν τη δεκαετία του '80 στήριζαν τη λειτουργία τους στο Νόμο Beer-Lambert, ο οποίος αγνοεί τη διάχυση του φωτός λόγω των ερυθρών 41 αιμοσφαιρίων με αποτέλεσμα οι μετρήσεις να μην είναι ορθές. Στη συνέχεια, όπως γίνεται και σήμερα, χρησιμοποιείται ειδικός πίνακας βασισμένος σε εμπειρικά δεδομένα για το συσχετισμό μίας συγκεκριμένης αναλογίας (ratio) R, που υπολογίζεται εσωτερικά στη συσκευή από την απόσβεση του φωτός, και του κορεσμού σε οξυγόνο και οι μετρήσεις είναι ορθές. Να σημειωθεί εδώ ότι ο Νόμος Beer-Lambert συσχετίζει την απόσβεση του φωτός με τις ιδιότητες του υλικού διαμέσου του οποίου μεταδίδεται το φως. Η παλμική οξυμετρία σήμερα έχει ευρεία χρήση, τόσο σε ιατρεία και νοσοκομεία (έως και σε μονάδες εντατικής θεραπείας) αλλά και στο σπίτι.

## - Αρχή Λειτουργίας

Το ατμοσφαιρικό οξυγόνο εισέρχεται στις κυψελίδες του πνεύμονα σε κάθε εισπνοή (από τις κυψελίδες περνά στο αίμα). Εκεί, ανακαλύφθηκε το 1860 ότι συνδέεται κατά το μεγαλύτερο ποσοστό με την αιμοσφαιρίνη (haemoglobin), πρωτεΐνη που βρίσκεται μέσα στα ερυθρά αιμοσφαίρια, και μέσω αυτής μεταφέρεται στη συνέχεια στην κυκλοφορία ώστε να φτάσει σε όλους τους περιφερικούς ιστούς του σώματος. Έτσι, το παλμικό οξύμετρο μετρά το ποσοστό του αίματος που έχει συνδεθεί με οξυγόνο, και πιο συγκεκριμένα το ποσοστό της αιμοσφαιρίνης που έχει συνδεθεί με οξυγόνο. Η τεχνολογία της παλμικής οξυμετρίας χρησιμοποιεί τα χαρακτηριστικά της απορρόφησης του φωτός από την αιμοσφαιρίνη και τον παλμικό τρόπο ροής του αίματος στις αρτηρίες. Συγκεκριμένα, παρατηρήθηκε το 1860 ότι η απορρόφηση του ορατού φωτός διαμέσου διαλύματος αιμοσφαιρίνης διέφερε με την οξυγόνωση. Αυτό συμβαίνει γιατί οι δύο κοινές μορφές του μορίου, η αρτηριακή αιμοσφαιρίνη, που είναι κορεσμένη σε οξυγόνο (oxidised haemoglobin (HbO<sub>2</sub>)) και έχει έντονο κόκκινο χρώμα και η φλεβική, που είναι φτωχή σε οξυγόνο (reduced haemoglobin (Hb)) και έχει πιο σκούρο χρώμα, έχουν σημαντικά διαφορετικά οπτικά φάσματα σε μήκη κύματος που κυμαίνονται από 500 nm έως 1000 nm, όπως φαίνεται στο Σχήμα 10. Τη διαφορά αυτή του χρώματος, και άρα τη διαφορετική απορρόφηση σε φως ορισμένου μήκους κύματος, εκμεταλλεύεται η παλμική οξυμετρία. Επίσης, σε κάθε παλμό, υπάρχει μία ελαφρά αύξηση του όγκου του αίματος που ρέει μέσω των αρτηριών, η οποία συνεπάγεται και μία αύξηση στην πλούσια σε οξυγόνο αιμοσφαιρίνη [5]–[7].

Το παλμικό οξύμετρο είναι μικρή ιατρική συσκευή, η οποία τοποθετείται συνήθως σε ένα από τα δάχτυλα του χεριού και έχει έναν αισθητήρα (sensor). Ο αισθητήρας αυτός φιλοξενεί μία πηγή φωτός, έναν ανιχνευτή φωτός και έναν μικροεπεξεργαστή, ο οποίος συγκρίνει και υπολογίζει τις διαφορές μεταξύ της πλούσιας και της πτωχής σε οξυγόνο αιμοσφαιρίνης [5]. Η μία πλευρά του αισθητήρα έχει μια πηγή φωτός, η οποία αποτελείται από ένα ζεύγος μικρών διόδων εκπομπής φωτός (Light Emitting Diode – LEDs) μέσα από μία φωτοδίοδο (photodiode), και έτσι εκπέμπεται φως σε δύο μήκη κύματος, κόκκινο (red – 660 nm) και υπέρυθρο (InfraRed – IR – 940 nm), τα οποία μεταδίδονται στην πλευρά του ανιχνευτή φωτός (photodetector) διαμέσου μέρους του σώματος που είναι σχετικά "διαφανές" και με καλή παλμική ροή αρτηριακού αίματος, όπως το δάχτυλο ή ο λοβός του αυτιού. Η πλούσια σε οξυγόνο αιμοσφαιρίνη, καθώς έχει έντονο κόκκινο χρώμα απορροφά περισσότερο το υπέρυθρο φως και αφήνει να περάσει περισσότερο το κόκκινο, ενώ η φτωχή σε οξυγόνο αιμοσφαιρίνη που έχει πιο σκούρο κόκκινο χρώμα, απορροφά περισσότερο το κόκκινο φως και αφήνει να περάσει το υπέρυθρο. Η κόκκινη ακτινοβολία έχει μήκη κύματος εύρους 630 – 700 nm και η υπέρυθη ακτινοβολία έχει μήκη κύματος εύρους 750 – 106 nm. Το φως που δεν απορροφάται αλλά διαπερνά το μέρος του σώματος και μεταδίδεται στην άλλη πλευρά του οξύμετρου, ανιχνεύεται από τη φωτοδίοδο, ενισχύεται και μετατρέπεται σε τάση. Η παλμική οξυμετρία υποθέτει ότι η απόσβεση του φωτός μπορεί να χωριστεί σε τρεις ανεξάρτητες συνιστώσες, αυτήν που οφείλεται στο αρτηριακό αίμα, τη συνιστώσα που οφείλεται στο φλεβικό αίμα και τέλος στους ιστούς. Ο μικροεπεξεργαστής, αφαιρώντας τις δύο τελευταίες χρησιμοποιεί μόνο τη συνιστώσα που οφείλεται στην εισροή του αρτηριακού αίματος για τον υπολογισμό του κορεσμού σε οξυγόνο, όπως αναφέρθηκε νωρίτερα στον τρόπο λειτουργίας της μετάδοσης (transmissive) [6]. Κάνει στη συνέχεια κάποιους υπολογισμούς και με τη χρήση ειδικού πίνακα βασισμένου σε εμπειρικά δεδομένα εξάγει τον κορεσμό σε οξυγόνο του αίματος. Μετατρέπει, τέλος, το αποτέλεσμα σε ψηφιακή ανάγνωση, ορατό στην οθόνη της συσκευής. Αυτές οι μετρήσεις βοηθούν τον ιατρό να εκτιμήσει την ποσότητα του οξυγόνου που μεταφέρεται στο αίμα και να αξιολογήσει την ανάγκη για συμπληρωματικό οξυγόνο. Τα περισσότερα οξύμετρα εμφανίζουν επίσης τον αριθμό των καρδιακών παλμών ενώ κάποια, όπως αυτά που χρησιμοποιήθηκαν στις μετρήσεις της παρούσας εργασίας εμφανίζουν και το φωτοπληθυσμογράφημα, έννοια που θα αναλυθεί στη συνέχεια [5,6,7] Περισσότερες πληροφορίες για τον τρόπο λειτουργίας του οξύμετρου στο σχετικό έγγραφο του Πανεπιστημίου της Οξφόρδης από το Τμήμα Ηλεκτρολόγων Μηχανικών (βλ. βιβλιογραφία [6]).

#### - Ενδείξεις: Πλεονεκτήματα και Περιορισμοί

Τα οξύμετρα είναι χρήσιμα σε πληθώρα περιπτώσεων όπου υπάρχει ανάγκη εκτίμησης των επιπέδων οξυγόνου στο αίμα με μη επεμβατικό τρόπο. Εναλλακτικός τρόπος εκτίμησης των επιπέδων οξυγόνου στο αίμα είναι στο εργαστήριο σε δείγμα αίματος που έχει ληφθεί επεμβατικά. Λόγω της ταχύτητας της εξέτασης η παλμική οξυμετρία είναι πολύ χρήσιμη σε τομείς όπως, στην ιατρική των επειγόντων, για παράδειγμα στην εκτίμηση της οξείας δύσπνοιας, ακόμη στη χειρουργική, σε διαδικασίες όπου χορηγείται καταστολή, όπως κατά τη βρογχοσκόπηση (διαγνωστική μέθοδο επισκόπησης των αναπνευστικών οδών των πνευμόνων ή αλλιώς των αεραγωγών) ή για την παρακολούθηση κατά τη μεταφορά βαρέως πασχόντων ασθενών.

Έπειτα, σε ασθενείς με αναπνευστικά ή καρδιολογικά προβλήματα όπου ενδέχεται να απαιτείται συμπληρωματική χορήγηση οξυγόνου, για να εκτιμηθεί η αποτελεσματικότητα της θεραπείας ή σε ασθενείς σε κίνδυνο αναπνευστικής ανεπάρκειας (respiratory failure) πρέπει να ελέγχεται η αποτελεσματικότητα της ανταλλαγής αερίων στους πνεύμονες, όπως για παράδειγμα πόσο καλά οξυγονώνεται το αρτηριακό αίμα στους πνεύμονες, σε συνεχή βάση [6]. Επίσης η χρησιμότητά τους είναι ανεκτίμητη στην εντατική θεραπεία (intensive care) για τη συνεχή παρακολούθηση των ασθενών υπό μηχανικό αερισμό. Χρησιμοποιείται και σε διαταραχές του ύπνου, συγκεκριμένα στην άπνοια ύπνου και την υπόπνοια όπου γίνεται ολονύκτια καταγραφή με ειδικά καταγραφικά οξύμετρα (νυκτερινή οξυμετρία) για διαγνωστικούς λόγους. Τέλος, στα μεγάλα υψόμετρα η πυκνότητα του οξυγόνου στην ατμόσφαιρα είναι μικρότερη και έτσι τα φορητά οξύμετρα μπορεί να είναι χρήσιμα στους ορειβάτες, όπως επίσης και στους πιλότους που πετάνε πάνω από 10.000 πόδια, όπου μπορεί να απαιτείται συμπληρωματικό οξυγόνο [5].

Η παλμική οξυμετρία εκτιμά μόνο την οξυγόνωση, όχι το μεταβολισμό του οξυγόνου ή τις απόλυτες τιμές του οξυγόνου στο αίμα, όπως για παράδειγμα στη σοβαρή αναιμία όπου το αίμα μεταφέρει λιγότερο οξυγόνο, η αιμοσφαιρίνη όμως είναι 100% κορεσμένη. Γι' αυτό το σκοπό, πρέπει να προσδιορισθούν και τα επίπεδα του διοξειδίου του άνθρακα στο αίμα (carbon dioxide – CO<sub>2</sub>). Έτσι, δεν υποκαθιστά τα αέρια αίματος που ελέγχονται στο εργαστήριο, διότι δεν παρέχει καμία ένδειξη για άλλες πολύ χρήσιμες παραμέτρους της αναπνευστικής λειτουργίας (διοξείδιο του άνθρακα, pH, διττανθρακικό HCO<sub>3</sub>- κ.ά.) [5,6].

#### - Παράγοντες που μπορεί να επηρεάσουν την εξέταση

Παράγοντες που μπορεί να επηρεάσουν τα αποτελέσματα της εξέτασης είναι:

- Μειωμένη ροή του αίματος προς τα περιφερικά αγγεία (αγγειοπάθειες)
- Ψυχρά άκρα ή εφίδρωση της περιοχής όπου συνδέεται ο αισθητήρας
- Κίνηση της περιοχής όπου συνδέεται ο αισθητήρας, π.χ. ρίγη
- Πολλαπλοί κάλοι στο δέρμα
- Πρόσφατη χορήγηση σκιαγραφικής ουσίας
- Κάπνισμα
- Εσφαλμένη εφαρμογή του αισθητήρα
- Σοβαρή αναιμία
- Δηλητηρίαση από μονοξείδιο του άνθρακα ή κυανούχα
- Μεθαιμοσφαιριναιμία (υψηλότερη από τη φυσιολογική παρουσία μεθαιμοσφαιρίνης – metHb)

Για να εξασφαλιστεί η ακρίβεια, θα πρέπει ο αισθητήρας να επιστρέφει σταθερές ενδείξεις, σταθερό παλμό και κορεσμό σε οξυγόνο. Απότομες αποκλίσεις μπορεί να υπονοούν λανθασμένες μετρήσεις για κάποιον από τους παραπάνω λόγους [5,6]



## - Δείκτης Διάχυσης (Perfusion Index – PI)

Ο δείκτης διάχυσης που εμφανίζεται στην οθόνη του παλμικού οξύμετρου αποτελεί μετρική αξιολόγησης της δύναμης του παλμού στην εκάστοτε τοποθεσία μέτρησης. Είναι ο λόγος του παλμικού αίματος προς το μη-παλμικό αίμα στους περιφερικούς ιστούς ενός ατόμου. Κυμαίνεται από 0.02%, για πολύ αδύναμο παλμό, έως 20%, για πολύ δυνατό παλμό. Αποτελεί μία αριθμητική μέτρηση που υποδεικνύει το ποσοστό της υπέρυθρης (IR) ακτινοβολίας που επιστρέφει στο φωτοανιχνευτή.

Μεταβάλλεται ανάλογα με τον ασθενή, την τοποθεσία μέτρησης και τις εκάστοτε φυσιολογικές συνθήκες. Αποτελεί μέτρο της αξιοπιστίας της εκάστοτε μέτρησης, καθώς αυξάνει ο PI αυξάνει και η αξιοπιστία των μετρήσεων. Έτσι, κάθε ασθενής πρέπει να προσδιορίσει το δικό του δείκτη διάχυσης για τον οποίο η μέτρηση είναι αξιόπιστη. Γενικά, κάτω από 0.4% οι τιμές του οξύμετρου είναι αναξιόπιστες. Το PI χρησιμοποιείται, μαζί με άλλες παραμέτρους, για την παρακολούθηση βαρέως πασχόντων ασθενών. Έχει πολλές εφαρμογές, όπως για παράδειγμα ως πρώτη προειδοποίηση σε περίπτωση αποτυχίας αναισθησίας, όπου έρευνες έχουν δείξει ότι η αύξηση του PI είναι ένας πρώιμος δείκτης ότι η γενική ή επισκληριδίου αναισθησία εκκίνησε τη διαστολή των περιφερικών αιμοφόρων αγγείων, το οποίο τυπικά συμβαίνει πριν την εκκίνηση της αναισθησίας, οπότε η έλλειψη αυτής της αιχμής (spike) υποδεικνύει την αποτυχία του επιθυμητού αποτελέσματος της αναισθησίας [8].

## 2.2 Ερευνητικό Υπόβαθρο

Όπως αναφέραμε, τα βιοσήματα ως μέσο για την παρακολούθηση της υγείας ενός ανθρώπου είναι μια τεχνική που υπάρχει πάρα πολλά χρόνια. Όμως πια ζούμε σε έναν κόσμο όπου ο δυτικός τρόπος ζωής, η καθιστική ζωή, η μεγάλη κατανάλωση λίπους και ζάχαρης, δημιουργεί χρόνιες παθήσεις. Επιπλέον η πρόοδος της Ιατρικής έχει αυξήσει τον προσδοκώμενο μέσο όρο ζωής, δημιουργώντας έναν πληθυσμό μεγαλύτερου όρου ηλικίας αλλά και ενός πληθυσμού ο οποίος αυξάνεται σταθερά. Όλα αυτά έχουν καταστήσει απαραίτητη την χρόνια και συστηματική παρακολούθηση των ασθενών. Η Τηλεϊατρική είναι ένα από τα πιο υποσχόμενα μέσα για τα παραπάνω καθώς μπορεί να συνδυάσει την πρόοδο της τεχνολογίας και την Ιατρική. Πολλές προσπάθειες έχουν ήδη γίνει ώστε να μπορέσουν αυτοί οι δύο αχανείς και αρχικά ξένοι τομείς να συνδυαστούν προς όφελος της υγείας των ανθρώπων. Μία από τις κατευθύνσεις που φαίνεται να έχει υπερισχύσει στην Τηλεϊατρική είναι η φορητότητα των συσκευών οι οποίες είναι υπεύθυνες για την λήψη, επεξεργασία και αποστολή των βιοσημάτων των ασθενών. Πολλά ερευνητικά έργα (research projects) έχουν στηριχτεί σε φορητούς υπολογιστές και συσκευές μικρού μεγέθους που όμως πλέον διαθέτουν ικανοποιητικές υπολογιστικές δυνατότητες επεξεργασίας. Όπως αναφέρεται στο άρθρο [37] έγινε προσπάθεια να δημιουργηθεί ένα σύστημα παρακολούθησης των βιοσημάτων του ασθενή μέσω της φορητής συσκευής ηλεκτρονικών παιχνιδιών nintendo DS η οποία λάμβανε βιοσήματα μέσω bluetooth και μέσω του πρωτοκόλλου TCP/IP και τα μετέφερε σε κάποια ζητούμενη απομακρυσμένη ηλεκτρονική διεύθυνση. Απαραίτητη προϋπόθεση ήταν η σύνδεση στο διαδίκτυο με χρήση Wi-Fi. Το παραπάνω αποτελεί μια πολύ εύχρηστη πρόταση για χρήση σε ασθενοφόρα και για κατ' οίκον παρακολούθηση, καθώς έχει τόσο την απαραίτητη υπολογιστική ισχύ όσο και την φορητότητα [35]. Επίσης στη δημοσίευση [38] δημιουργήθηκε ένα σύστημα χαμηλής

κατανάλωσης ενέργειας για την παρακολούθηση διάφορων βιοσημάτων ασθενών, με τη μορφή μιας εφαρμογής σε καθημερινά σύγχρονα κινητά τηλέφωνα (smartphones). Μέσω του συστήματος αυτού, τα βιοσήματα που λάμβανε ο ασθενής με χρήση του ήδη υπάρχοντος συστήματος bluetooth, που διαθέτουν πλέον όλα τα σύγχρονα τηλέφωνα, αποστέλλονταν μέσω Wi-Fi ή 3G σε έναν εξωτερικό εξυπηρετητή απ' όπου μπορούσε να γίνει η απομακρυσμένη παρακολούθηση καθώς και η ανταλλαγή πληροφοριών και εντολών [36]. Πέρα από χρήση ήδη υπάρχοντων συσκευών έχουν δημιουργηθεί και χρησιμοποιηθεί και αυτοσχέδιες συσκευές για παρακολούθηση ασθενών στην καθημερινή τους ζωή. Στη δημοσίευση [39] μια συσκευή παρακολούθησης βιοσημάτων σχεδιάστηκε και πειραματικά τοποθετήθηκε στη μέση των ασθενών. Η ζώνη αυτή, η οποία είχε επεξεργαστική μονάδα, επιταχυνσιόμετρο και σύστημα ασύρματης επικοινωνίας, βοηθούσε την παρακολούθηση του ασθενή κατά τη διάρκεια σωματικής άσκησης [37]. Τέλος προσπάθειες έγιναν ώστε η παρακολούθηση να μπορεί να χρησιμοποιεί τις πλήρεις δυνατότητες που παρέχει ένας σύγχρονος Υπολογιστής. Στη δημοσίευση [40] προτάθηκε η χρήση του Raspberry-Pi ως μια συσκευή για την παρακολούθηση και επίβλεψη των βιοσημάτων του ασθενή, το οποίο καθώς πρόκειται για έναν πλήρη φορητό ενσωματωμένο υπολογιστή με λειτουργικό σύστημα Linux (Rasbian, λειτουργικό βασισμένο σε Debian) παρέχει πολύ μεγαλύτερη ποικιλία δυνατοτήτων και ευελιξία καθιστώντας το πολύ πιο κατάλληλο από άλλους μικροελεγκτές (microcontrollers), οι οποίοι χρησιμοποιούνται σε αρκετά άλλα projects [38].

Λαμβάνοντας υπόψιν τα παραπάνω ευρήματα πάνω στον τομέα των βιοσημάτων, στην παρούσα διπλωματική εργασία θα δημιουργηθεί ένα σύστημα παρακολούθησης και αποστολής των βιοσημάτων των ασθενών στηριγμένο σε Raspberry-Pi. Οι λόγοι που επιλέχθηκε η συγκεκριμένη συσκευή από τους ανταγωνιστές της ήταν η χαμηλή τιμή αγοράς, η open-source κοινότητα, η οποία παρέχει πρόσβαση σε μια πολύ μεγάλη βάση από σχετικά projects, καθώς και το πλήθος των ενσωματωμένων δυνατοτήτων που παρέχει (USB-ports, HDMI έξοδο, Bluetooth και Wi-Fi σύνδεση, λειτουργικό σύστημα, επεξεργαστική μονάδα) [27]. Όπως και σε όλες τις προηγούμενες δημοσιεύσεις υπήρξε η ανάγκη για την προβολή και αποστολή των βιοσημάτων έτσι και στην παρούσα διπλωματική εργασία ως μέσο προβολής των αποτελεσμάτων χρησιμοποιήθηκαν οι φυλλομετρητές (browsers) ενός σύγχρονου υπολογιστή ενώ ως μέσο αποστολής το διαδίκτυο και το πρωτόκολλο TCP/IP.

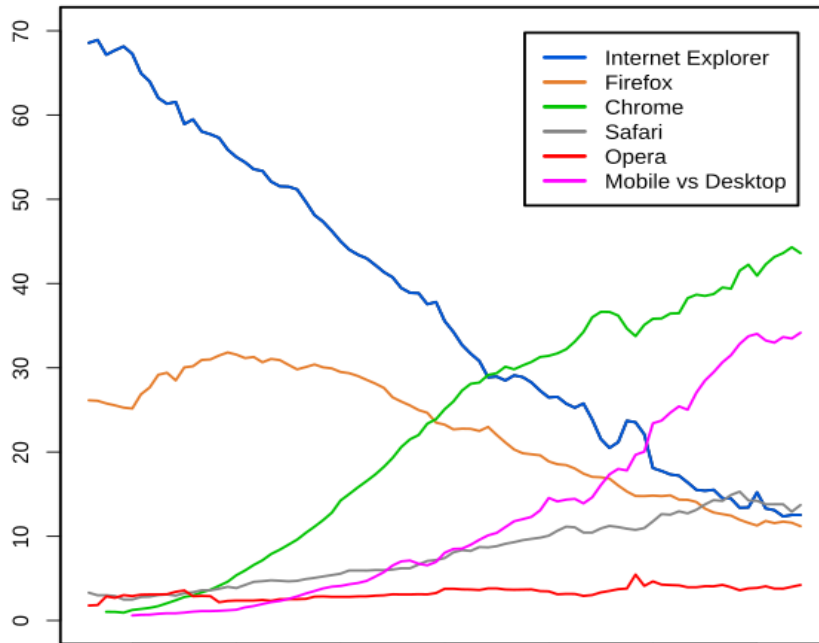
## **2.2 Τεχνολογικό Υπόβαθρο – Βασικές Έννοιες**

### **2.2.1 Φυλλομετρητές Ιστού (Web Browsers)**

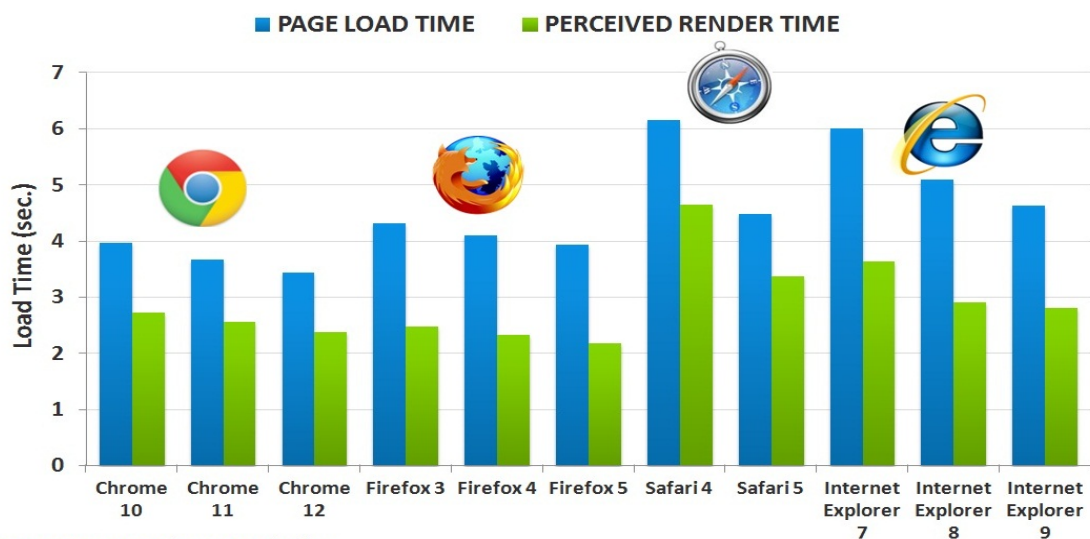
Ο web browser ή φυλλομετρητής ιστοσελίδων είναι μια μορφή λογισμικού που επιτρέπει στον χρήστη του να προβάλλει και να αλληλεπιδρά με κείμενα, εικόνες, βίντεο, μουσική, και άλλες πληροφορίες, συνήθως αναρτημένες σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Οι φυλλομετρητές χρησιμοποιούν τη γλώσσα μορφοποίησης HTML για την προβολή των ιστοσελίδων, πράγμα που οδηγεί σε διαφορετική εμφάνιση της ίδιας σελίδας σε διαφορετικό browser. Οι πλοηγοί Web ουσιαστικά αποτελούν λογισμικό πελάτη του δικτυακού πρωτοκόλλου επιπέδου εφαρμογών HTTP [9]. Όπως φαίνεται από το διάγραμμα των σχημάτων 3a

και 3b, τα τελευταία χρόνια ο πλέον δημοφιλής web browser είναι αυτός που εισήγαγε η Google το 2008, ο οποίος δεν είναι άλλος από τον Chrome. Λαμβάνοντας υπόψη τη δημοτικότητα του Chrome καθώς και την προσωπική προτίμηση ως προς αυτόν, επιλέχθηκε ως περιβάλλον εξέλιξης της εφαρμογής – συστήματος. Ένας δεύτερος λόγος επιλογής του Google Chrome ως περιβάλλοντος εξέλιξης της εφαρμογής είναι η απόδοση και η σταθερότητα που παρουσιάζει σε σχέση με τους άλλους browsers, το οποίο είναι ιδιαίτερα σημαντικό για εφαρμογές Τηλεϊατρικής.

Usage share of web browsers



Σχήμα 6: Ποσοστά χρήσης των web browsers παγκοσμίως.



Source: Gomez Real-User Monitoring

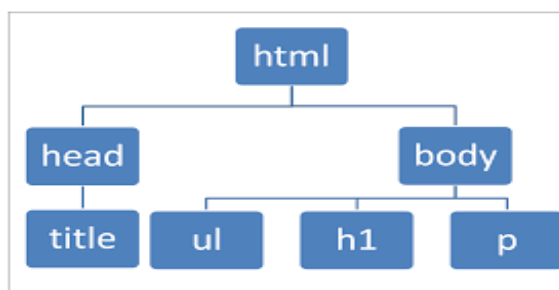
Σχήμα 7: Απόδοση Διάφορων Web Browsers.

## 2.2.2 Web Applications ως Ολοκληρωμένες Εφαρμογές

Μια Εφαρμογή Ιστού (Web Application) έχει ως σκοπό την αλληλεπίδραση του χρήστη όχι μόνο με απλή προβολή κειμένου και εικόνας αλλά με πιο πλούσια και πιο περίπλοκη λειτουργικότητα. Η γλώσσα HTML γράφεται με τη μορφή στοιχείων (HTML elements) αποτελούμενων από tags, περικλειόμενα από τους χαρακτήρες '<', '>', όπως φαίνεται και στην Σχήμα 4. Ο κώδικας μετατρέπεται από τον browser σε μια δενδρική δομή αποτελούμενη από JavaScript αντικείμενα-κόμβους, ή όπως αλλιώς ονομάζεται DOM (Document Object Model). Ο σκοπός της μετατροπής αυτής είναι η παροχή μιας προγραμματιστικής διεπαφής για την εκτέλεση σεναρίων (scripting), όπως αφαίρεση, πρόσθεση, αντικατάσταση και τροποποίηση του «ενεργού» HTML εγγράφου χρησιμοποιώντας JavaScript. Στα HTML στοιχεία μπορούν να αποδοθούν γνωρίσματα με τη μορφή ζευγών ονόματος-τιμής (key-value pairs), ώστε να καταστεί δυνατή η παραμετροποίησή τους. Στοιχεία που αντιστοιχούν σε κόμβους στο DOM μπορούν επίσης να δημιουργηθούν δυναμικά χρησιμοποιώντας μεθόδους που παρέχει η JavaScript, όπως η createElement(). Το νόημα όλων αυτών είναι πως JavaScript και HTML είναι στενά συνδεδεμένες, υπό την έννοια ότι ο προγραμματισμός σε HTML μπορεί να επιτευχθεί με χρήση JavaScript.



Σχήμα 8: Γραφική μορφή ενός απλού HTML κειμένου.



Σχήμα 9: Document Object Model μιας web σελίδας.

Η δημιουργία ιστοσελίδων μόνο με HTML ωστόσο κρίνεται ανεπαρκής για τα σημερινά δεδομένα του δικτυακού προγραμματισμού. Οι σύγχρονοι ιστότοποι μπορούν να εννοηθούν ως ένας συνδυασμός δομής (structure), ύφους (style) και διαδραστικότητας (interactivity) [10]. Υπεύθυνες για τα τρία αυτά διαφορετικά στοιχεία είναι οι τεχνολογίες HTML, CSS και JavaScript, αντίστοιχα, οι οποίες από κοινού συνεισφέρουν στη διανομή πλούσιου περιεχομένου σελίδων ιστού. Η HTML μέσω κατάλληλων tag μπορεί να συμπεριλάβει πλέον CSS και JavaScript αρχεία ( tag <style> για CSS και tag <script> για JavaScript).



*Σχήμα 10: Οι τεχνολογίες που σε συνδυασμό δημιουργούν εφαρμογές δικτύου.*

Η γλώσσα CSS (Cascading Style Sheet) δημιουργήθηκε και προτάθηκε από την κοινοπραξία W3C (World Wide Web Consortium), τον κύριο διεθνή οργανισμό προτύπων για το Web, τον Αύγουστο του 1996 [11]. Γενικά χρησιμοποιείται για να παραμετροποιήσει την εμφάνιση του HTML περιεχομένου, προσθέτοντας γνωρίσματα όπως στυλ γραμματοσειράς, χρώμα, διαστάσεις κτλ. Παλιότερα, κάτι τέτοιο ήταν εφικτό και με χρήση μόνο HTML, αλλά ήταν μια επίπονη προγραμματιστική διαδικασία, καθώς για κάθε στοιχείο έπρεπε να γραφτούν όλα τα γνωρίσματα «με το χέρι». Με την εισαγωγή της CSS, σε ένα ή περισσότερα στοιχεία μπορεί να ανατεθεί μια κλάση (class attribute) και ο κώδικας που είναι υπεύθυνος για τους κανόνες εμφάνισης της κλάσης να συνταχτεί μία φορά και η αντιστοίχιση με τα στοιχεία να γίνεται αυτόματα από το browser.

Το πρόβλημα όμως της διαδραστικότητας των ιστοσελίδων παρέμεινε. Η λύση δόθηκε το Δεκέμβριο του 1995. Η JavaScript δημιουργήθηκε από την Netscape Communications Corporation [12]. Με την εισαγωγή της JavaScript οι ιστοσελίδες είναι πιο δυναμικές καθώς υπάρχει η δυνατότητα για την αλληλεπίδραση με το D.O.M. και για προσθήκη λειτουργικότητας: φόρμες κειμένου, κουμπιά, εκτέλεση υπολογισμών και αλγορίθμων, προσθήκη event handlers (ειδικές συναρτήσεις που καλούνται όταν συμβεί κάποιο γεγονός), επικοινωνία με τον εξυπηρετητή και γενικότερα όλες τις δυνατότητες που προσφέρει μια πλήρης (κατα Turing) γλώσσα προγραμματισμού. Ένα επιπλέον πλεονέκτημα αυτής της προσέγγισης είναι ότι ο κώδικας JavaScript μεταφράζεται και εκτελείται από τον browser του χρήστη μέσω των JavaScript engines που υπάρχουν στον πυρήνα των browsers. Έτσι η εκτέλεση δεν βασίζεται στην ταχύτητα του δικτύου αλλά αποκλειστικά στο τοπικό λογισμικό και υλικό.

Η πλέον διαδεδομένη αρχιτεκτονική οργάνωσης μιας web page είναι η HTML σελίδα να περιέχει τα κατάλληλες αναφορές σε εξωτερικά αρχεία (<script > και <style> αντίστοιχα) και ο αντίστοιχος πραγματικός κώδικας να βρίσκεται σε διαφορετικά αρχεία, πράγμα που εξυπηρετεί την αυτονομία των επιμέρους στοιχείων, την αποσφαλμάτωση,

τη συντήρηση και την εύκολη αναγνωσιμότητα. Η τάση γι' αυτή την οργάνωση είναι πλέον αποκλειστική, γεγονός που έχει οδηγήσει στην ανάπτυξη ειδικών εργαλείων τα οποία βοηθούν το έργο του προγραμματιστή. Φυσικά κάποια από αυτά χρησιμοποιήθηκαν για την παρούσα πλατφόρμα.

### 2.2.3. Web Browser Συμβατότητα

Ως web browser συμβατότητα ορίζεται η δυνατότητα μιας ιστοσελίδας, διαδικτυακής εφαρμογής, ενός σεναρίου, ή μιας μορφοποίησης HTML να λειτουργεί με τον ίδιο τρόπο σε όλους τους browsers που υπάρχουν στο εμπόριο.

Το κέρδος αυτής της προσέγγισης είναι ότι η ιστοσελίδα είναι διαθέσιμη σε όλο το κοινό ανεξάρτητα από τον φυλλομετρητή, ενώ δεν υπάρχει κάποια μείωση απόδοσης μεταξύ διαφορετικών browsers [13]. Παρ' όλες τις προσπάθειες που έχουν γίνει για μια ομοιογένεια και προτυποποίηση των browsers ακόμα υπάρχουν διαφοροποιήσεις μεταξύ τους οι οποίες μπορούν να οδηγήσουν ακόμα και σε ανεξήγητες συμπεριφορές. Γι' αυτό δίνεται η δυνατότητα στον προγραμματιστή να ρωτήσει τον Web Browser αν υποστηρίζει κάποια χαρακτηριστικά και να αποφευχθούν έτσι λάθη. Οι τεχνολογίες που χρησιμοποιήθηκαν στην παρούσα εφαρμογή (WebRTC, Websockets, Ajax, HTTP requests) είναι πλέον εγκατεστημένες στους περισσότερους browsers και έτσι δεν αντιμετωπίστηκαν περιπτώσεις μη συμβατότητας των τεχνολογιών.

### 2.2.4 CORS

Το CORS (Cross-origin resource sharing) είναι ένας μηχανισμός που επιτρέπει σε κάποιους πόρους μια ιστοσελίδας να χρησιμοποιηθούν από διαφορετικό domain εκτός αυτού στο οποίο βρίσκονται. Μια ιστοσελίδα γενικά μπορεί να συμπεριλάβει ελεύθερα, εικόνες, stylesheets, scripts, videos και plugins που προέρχονται από οποιοδήποτε άλλο domain. Δεν ισχύει το ίδιο όμως στην περίπτωση των AJAX Requests (XMLHttpRequest) όταν προέρχονται από διαφορετικό domain από αυτό στο οποίο βρίσκονται τα δεδομένα που ζητούνται (cross-domain). Τα cross-domain AJAX request είναι απαγορευμένα από προεπιλογή για λόγους ασφαλείας. Το CORS δίνει τη δυνατότητα να καθοριστεί με ασφάλεια αν θα πρέπει να επιτραπεί η όχι κάποιο cross-domain request μεταξύ του browser και ενός server.

### 2.2.5 JavaScript

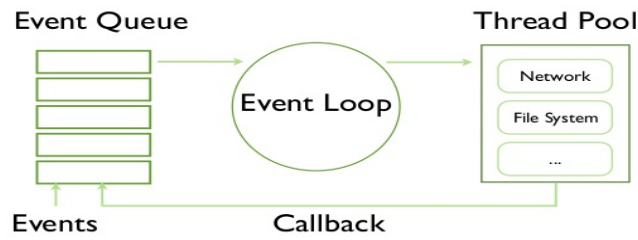
Για ολόκληρη την εφαρμογή χρησιμοποιήθηκε η JavaScript (ο server υλοποιήθηκε με nodejs που είναι σε Javascript, βλέπε Node.js παρ. 2.2.9), οπότε κρίνεται απαραίτητο να γίνει μια πιο αναλυτική αναφορά σε αυτήν. Όπως είδαμε στα παραπάνω κεφάλαια τα προγράμματα clients έχουν μεγάλο φόρτο εργασίας, πράγμα που σημαίνει ότι ο κώδικας JavaScript είναι εκτενέστερος και υπολογιστικά απαιτητικότερος. Αυτό σημαίνει ότι οι μηχανές JavaScript πρέπει να είναι γρήγορες και αποτελεσματικές.

### 2.2.5.1 Δυναμικό Σύστημα Τύπων και Ταυτοχρονισμός

Η γλώσσα JavaScript είναι μονοθηματική (single-threaded), έτσι πολλαπλά scripts δεν είναι σε θέση να τρέξουν παράλληλα. Ένα δεύτερο χαρακτηριστικό της JavaScript, που δυνητικά μπορεί να καθυστερήσει την εκτέλεση ενός προγράμματος, είναι το δυναμικό σύστημα τύπων και ο δυναμικός έλεγχος. Αυτό σημαίνει ότι ο συμπερασμός τύπου των μεταβλητών (int, float, string) γίνεται κατά τη διάρκεια της εκτέλεσης, οπότε και πιθανά σφάλματα ανιχνεύονται σε αυτό το στάδιο και όχι νωρίτερα. Τα δύο παραπάνω χαρακτηριστικά της γλώσσας γίνονται άμεσα αντιληπτά στην περίπτωση που σε μία ιστοσελίδα-εφαρμογή θα χρειαστεί να τρέξουν πολλά διαφορετικά scripts, κάθε ένα από τα οποία θα έχει να επεξεργαστεί έναν μεγάλο όγκο δεδομένων. Αυτό θα έκανε την εφαρμογή αργή ή ακόμα και μη αποκρίσιμη. Όμως οι σύγχρονοι browsers είναι σε θέση να αποκρύπτουν την πολυπλοκότητα αυτή από τον χρήστη και να παραμένουν αποκρίσιμοι.

### 2.2.5.2 Ο Βρόχος Συμβάντων (event loop)

Στη JavaScript, σχεδόν όλες οι λειτουργίες εισόδου/εξόδου (I/O) εκτελούνται ασύγχρονα, χωρίς μπλοκάρισμα (non-blocking execution). Σε αυτές περιλαμβάνονται HTTP αιτήσεις, ενέργειες πάνω σε βάσεις δεδομένων, εγγραφές και αναγνώσεις από το σκληρό δίσκο. Το μοναδικό νήμα ζητά από το περιβάλλον εκτέλεσης να πραγματοποιήσει μια ενέργεια, παρέχοντάς του μια συνάρτηση επανάκληση (callback function), συνεχίζοντας με την εκτέλεση άλλων εργασιών. Όταν η ενέργεια που ζητήθηκε προηγουμένως ολοκληρωθεί, ένα μήνυμα εισάγεται σε μια ουρά μαζί με ένα παρεχόμενο callback. Κάποια στιγμή στο μέλλον, το μήνυμα αυτό θα αφαιρεθεί από την ουρά και η συνάρτηση επανάκλησης θα εκτελεστεί. Αυτό το διαδραστικό, πλήρως ασύγχρονο, μοντέλο μπορεί να είναι γνώριμο στους προγραμματιστές που αναπτύσσουν λογισμικό διεπαφής χρήστη – εκεί που συμβάντα όπως το πάτημα ενός κουμπιού ή η κύλιση του παραθύρου μπορούν να προκύψουν οποιαδήποτε στιγμή και πρέπει να «εξυπηρετηθούν», αλλά διαφέρει σημαντικά από το σύγχρονο μοντέλο αίτησης-απόκρισης που συναντάται σε τυπικές υλοποιήσεις εφαρμογών εξυπηρετητή. Αυτή η απεμπλοκή του καλούντος από την απάντηση που αυτός αναμένει, επιτρέπει στο περιβάλλον εκτέλεσης της JavaScript να ασχοληθεί με άλλες διεργασίες ενώ «περιμένει» την ασύγχρονη ενέργεια να διεκπεραιωθεί και να έρθει η στιγμή να εκκινήσει το callback. Αυτή η ουρά – αόρατη στον προγραμματιστή – στην οποία τα μηνύματα αποθηκεύονται προσωρινά μαζί με τα αντίστοιχα εγγεγραμμένα callbacks, ονομάζεται βρόχος συμβάντων.



Σχήμα 11: Event loop της JavaScript.

### 2.2.5.3 Run-to-Completion Logic

Η πρακτική που ακολουθεί το περιβάλλον της JavaScript είναι η πλήρης επεξεργασία κάθε μηνύματος προτού συνεχίσει με το επόμενο. Αυτό προσφέρει κάποιες ελκυστικές ιδιότητες κατά το σχεδιασμό της λογικής των προγραμμάτων, συμπεριλαμβανομένης της εγγύησης της εγκυρότητας των δεδομένων κατά τη διάρκεια της εκτέλεσης συναρτήσεων. Παρατηρείται δηλαδή διαφορά με το μοντέλο της C, κατά το οποίο, όταν μία συνάρτηση ή κομμάτι κώδικα εκτελείται μέσα σε κάποιο νήμα, το σύστημα έχει τη δυνατότητα να διακόψει την εκτέλεσή τους και να μεταφέρει τον έλεγχο σε κάποιο άλλο νήμα. Βέβαια, υπάρχουν και μειονεκτήματα στην προσέγγιση αυτή, το σπουδαιότερο εκ των οποίων έχει να κάνει με το εάν ένα μήνυμα χρειαστεί σημαντικό χρονικό διάστημα για την επεξεργασία του, όλη η εφαρμογή καθίσταται ανίκανη να διαχειριστεί οποιαδήποτε αλληλεπίδραση με το χρήστη. Δεν είναι σπάνια, ακόμα και σήμερα, η εμφάνιση ειδοποίησης με τη μορφή ξεχωριστού παραθύρου από το browser, σύμφωνα με την οποία «η εκτέλεση κάποιου script καθιστά την εφαρμογή μη αποκρίσιμη». Οι προγραμματιστές για να υπερκεράσουν τους περιορισμούς αυτούς προσπαθούσαν να μειώσουν όσο γίνεται το χρόνο επεξεργασίας που απαιτείτο ή, αν αυτό δεν ήταν δυνατό, να «μοιράσουν» το φόρτο σε μικρότερα μηνύματα, τα οποία τοποθετούσαν εκ νέου στο βρόχο συμβάντων χρησιμοποιώντας ειδικά API calls, όπως `setTimeout()` και `setInterval()`.<sup>1</sup>

<sup>1</sup> Πλέον έχουν προστεθεί Web Workers, οι οποίοι είναι ένα API των browsers που αναλαμβάνει την παρασκηνακή εκτέλεση scripts από τις εφαρμογές ιστού ανεξάρτητη από το δικό τους νήμα εκτέλεσης.



## 2.2.6 Single Page Applications and Frameworks

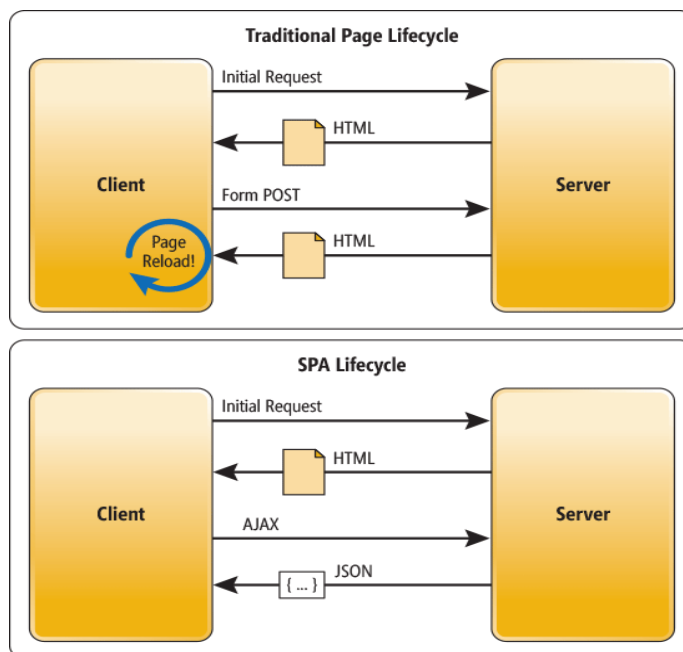
### 2.2.6.1 Τα Προγράμματα Πελάτες Γίνονται πιο Σύνθετα

Όπως έχει ήδη αναφερθεί από την αρχική εισαγωγή των browsers ως περιβάλλον εκτέλεσης του δικτυακού προγραμματισμού, η πρόοδος είναι αλματώδης. Ενώ η αρχική προσέγγιση ήταν οι HTML σελίδες να χρησιμοποιούνται ως απλό μέσο για την παρουσίαση στατικών διαδικτυακών σελίδων, με την εμφάνιση του CGI (Common Gateway Interface) ξεκίνησε η περίοδος των δυναμικών ιστοσελίδων, δηλαδή σελίδων που δημιουργούνταν ανάλογα με το συγκεκριμένο χρήστη σε κάποιο ευρύτερο πλαίσιο, με την βοήθεια κάποιου backend συστήματος, δηλαδή servers υλοποιημένων σε PHP/ASP/JSP, κλπ όπου οι πελάτες (thin clients) απλά απεικόνιζαν το περιεχόμενο αυτό.

### 2.2.6.2 AJAX (Asynchronous JavaScript and XML)

Η μεγάλη αλλαγή όμως στο διαδικτυακό προγραμματισμό επιτεύχθηκε με την είσοδο των AJAX (Asynchronous JavaScript and XML) τεχνικών οι οποίες πλέον μετατόπισαν τον έλεγχο της ροής και των πληροφοριών από τους εξυπηρετητές στους πελάτες. Ο client side προγραμματισμός είχε αναβαθμιστεί και πλέον είχε έναν πολύ σημαντικό ρόλο σε μια εφαρμογή. Ταυτόχρονα πλαίσια εργασίας στον εξυπηρετητή (server-side frameworks) επέτρεψαν τον διαμοιρασμό της λογικής μεταξύ πελάτη – εξυπηρετητή. Πλέον ο πελάτης δεν αιτείται μόνο HTML περιεχόμενο αλλά μπορεί να ζητήσει και δεδομένα από το backend, και όταν αυτά επιστραφούν να τα απεικονίσει / επεξεργαστεί όπως απαιτείται στην κάθε περίπτωση [14]. Με την έλευση της HTML5, οι browsers ισχυροποιήθηκαν και προστέθηκαν πολλές δυνατότητες. Σταδιακά η προσοχή του δικτυακού προγραμματισμού στράφηκε προς τις RIAs (Rich Internet Application), εφαρμογές, οι οποίες έχουν πολλά από τα χαρακτηριστικά των native, αλλά συνήθως η εκτέλεσή τους απαιτεί την εγκατάσταση κάποιας επέκτασης στο browser. Η επιστημονική κοινότητα αντιλήφθηκε αυτή τη νέα δυναμική που αναπτυσσόταν και πλέον από το 2012, σύμφωνα με τις τάσεις που καταγράφει η Google [15] άρχισε σταδιακά να εγκαταλείπει τα frameworks που στηρίζονταν σε plug-ins και third-party εργαλεία. Τη μεταστροφή αυτή βοήθησε και η αντικατάσταση των τελευταίων με πληθώρα νέων που αξιοποιούσαν τα βελτιωμένα APIs που προσέφεραν πλέον οι browsers και την επεξεργαστική ισχύ των προσωπικών υπολογιστών και διευκόλυναν την ανάπτυξη και συντήρηση μεγάλης έκτασης εφαρμογών.

Αυτό σηματοδοτεί την εποχή όπου τα προγράμματα πελάτες (Client Programs) σταματούν να είναι thin clients (μικρός φόρτος εργασίας, ελάχιστες αρμοδιότητες, απλές δυνατότητες αλληλεπίδρασης χρήστη – εφαρμογής, εμφάνιση περιεχομένου) και πλέον γίνονται thick clients. Έτσι τώρα τα προγράμματα – πελάτες (ή αλλιώς frontend) αναλαμβάνουν υπολογιστικά περίπλοκες εργασίες, αλληλεπιδρούν άμεσα με το backend και η λογική παρουσίασης μετατοπίζεται στον frontend/πελάτη.



Σχήμα 12: Διαφορές μεταξύ της υπάρχουσας λογικής του διαδικτύου και της SPA.

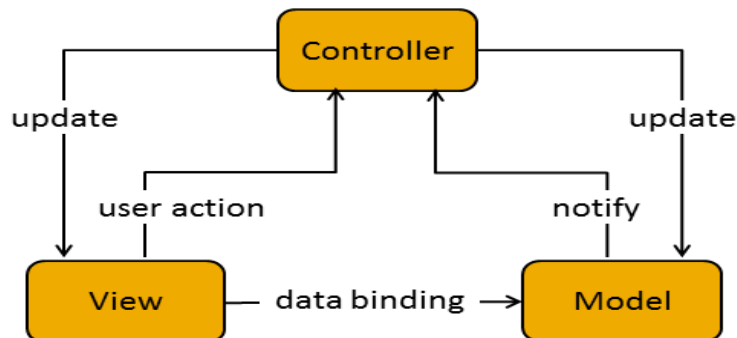
### 2.2.6.3 SPA και το Μοντέλο MVC

Η επόμενη μεγάλη ιδέα ήρθε το 2002 με την περουσίαση της ιδέας των εφαρμογών μονής σελίδας (Single Page Applications). Η λογική εδώ είναι ότι μια web εφαρμογή λαμβάνει χώρα σε μία μόνο ιστοσελίδα με στόχο να παρέχει μια πιο άμεση και ομαλή εμπειρία συγκρίσιμη με εφαρμογές επιφάνειας εργασίας (desktop applications). Η σελίδα φορτώνεται μία μόνο φορά και στη συνέχεια τα δεδομένα φορτώνονται δυναμικά μέσω αλληλεπίδρασης server – client, ενώ ταυτόχρονα εξακολουθεί να δίνει στον χρήστη την εντύπωση ότι πλοηγείται σε διαφορετικές σελίδες, μέρη της εφαρμογής.

Η πολυπλοκότητα και η ποσότητα της λογικής που δόθηκε στον πελάτη δημιούργησε νέα προβλήματα: το μέγεθος του κώδικα αυξήθηκε δραματικά, και οι τεχνοτροπίες που δημιουργήθηκαν ποίκιλλαν πολύ. Η λύση δόθηκε ομαδοποιώντας πρακτικές και σχεδιαστικές επιλογές σε ολοκληρωμένες τεχνοτροπίες (frameworks) που μέχρι στιγμής υπήρχαν μόνο στους εξυπηρετητές. Η πιο γνωστή τεχνοτροπία όλων αυτών των JavaScript Frameworks είναι η Model – View – Controller (MVC pattern) η οποία αποτελείται από τα εξής τρία μέρη:

- **Μοντέλα (Models)** που αναπαριστούν τις σχετικές με την εφαρμογή πληροφορίες και δεδομένα, όπως συγκεκριμένες κλάσεις-δοχεία (container classes) δεδομένων. Τα μοντέλα μπορούν να ενημερώνουν τυχόν παρατηρητές όταν η κατάστασή τους αλλάζει (π.χ. όταν κάποια πληροφορία που κρατούν ενημερώνεται ή διαγράφεται).
- **Όψεις (Views)** που τυπικά θεωρούνται ως η διεπαφή του χρήστη με την εφαρμογή (π.χ. ο κώδικας HTML και CSS). Πρέπει να γνωρίζουν για την ύπαρξη των μοντέλων, έτσι ώστε να τα παρατηρούν, αλλά δεν επικοινωνούν κατευθείαν μαζί τους.

- **Ελεγκτές (Controllers)** που υλοποιούν τη λογική παρουσίασης (presentation logic) της εφαρμογής. Αυτοί είναι που παίρνουν τις αποφάσεις και ο συνδετικός κρίκος μεταξύ Μοντέλων και Όψεων.



Σχήμα 13: Διαφορές μεταξύ της υπάρχουσας λογικής του διαδικτύου και της SPA.

Η πληθώρα των MVC frameworks που υπάρχουν διαθέσιμα σήμερα, ανάλογα με τους στόχους τους, δεν υλοποιούν με τον ίδιο τρόπο το παραπάνω προγραμματιστικό μοτίβο αλλά συνήθως κάποια παραλλαγή του, με σκοπό πάντα όμως την επιβολή της σχεδιαστικής αρχής γνωστής ως SoC (Separation of Concerns). Παρατηρείται, για παράδειγμα, η συγχώνευση του ρόλου του ελεγκτή και της όψης σε μία οντότητα με αυξημένες αρμοδιότητες και λειτουργικότητα. Γι' αυτό το λόγο, οι υλοποιήσεις που στηρίζονται σε τέτοιου είδους πιο «χαλαρές» προσεγγίσεις συναντώνται στη βιβλιογραφία και ως MV\*.

## 2.2.7 Websockets

Η λογική των SPA έφερε μια επανάσταση στον δικτυακό προγραμματισμό. Η ανάπτυξη των frameworks, καθώς και οι τεχνικές AJAX ώθησαν σε νέα επίπεδα τον κόσμο του Internet. Όμως ένα πρόβλημα παρέμενε: το πρόβλημα της επικοινωνίας με τον client σε πραγματικό χρόνο. Μέσω των AJAX ο client μπορούσε, όποτε επιθυμούσε, να επικοινωνήσει με τον server και να αλληλεπιδράσει μαζί του. Το αντίθετο όμως δεν ήταν εφικτό, δηλαδή αν ο Server ήθελε να επικοινωνήσει με τον Client, δεν μπορούσε παρά μόνο αν ο Client έστελνε κάποιο αίτημα. Αρχικά αυτό αντιμετωπίστηκε με μία τεχνική γνωστή ως *polling*: ο client ρωτούσε ανά τακτά χρονικά διαστήματα τον Server αν συνέβη κάποια αλλαγή (π.χ κάθε 2 λεπτά). Αυτό καταναλώνει μεγάλο μέρος του διαθέσιμου εύρους ζώνης και επιπλέον δημιουργούσε καθυστερήσεις και δεν έδινε τη δυνατότητα επικοινωνίας πραγματικού χρόνου.

Λύση στα παραπάνω αποτέλεσαν τα Websockets, ένα πρωτόκολλο επικοινωνίας που επιτρέπει αμφίδρομη επικοινωνία πάνω από μια σταθερή TCP σύνδεση, το οποίο προτυποποιήθηκε από το IETF ως RFC 6455 το 2011. Πλέον η επικοινωνία μεταξύ πελάτη – εξυπηρετητή ήταν δυνατή με ελάχιστες καθυστερήσεις (μόνο μια αρχική χειραψία απαιτείται μεταξύ server – client) και το σημαντικότερο, επέτρεπε την ασύγχρονη και πραγματικού χρόνου μεταφορά μηνυμάτων και δεδομένων από τον εξυπηρετητή στον πελάτη και αντίστροφα. Κλειδί στα παραπάνω αποτέλεσε η

ανάπτυξη των εφαρμογών μονής σελίδας, καθώς όπως είπαμε η σύνδεση γίνεται πάνω από μια σταθερή TCP σύνδεση, και έτσι αν ο πελάτης συνεχώς άλλαζε σελίδα, τα websockets έχανα κάπως το νόημά τους.

## 2.2.8 AngularJS ως Framework

Το JavaScript Framework που επιλέχτηκε για την παρούσα διπλωματική εργασία είναι το *AngularJS*. Οι λόγοι που επιλέχτηκε ήταν διάφοροι. Βασικός παράγοντας ήταν η προσωπική προτίμηση, αλλά επίσης εξετάστηκαν οι δυνατότητες και η υποστήριξη που το συγκεκριμένο framework προσφέρει. Το συγκεκριμένο framework είναι επίσης ένα από τα δημοφιλέστερα και πολύ καλά παρουσιασμένο και τεκμηριωμένο με αποτέλεσμα την εύκολη εκμάθηση και συνεχή ανανέωσή του. Το AngularJS Framework δημιουργήθηκε από την Google το 2014 και από τότε εξακολουθεί να είναι υπό την επίβλεψή της. Παρέχει ένα πλήρες MVC μοντέλο, ενώ προσθέτει πολλά νέα tag attribute στο ήδη υπάρχον λεξιλόγιο της HTML (md-button, md-autocomplete κλπ), δηλαδή επεκτείνει την παραδοσιακή HTML. Μέσω των εύκολων APIs που παρέχει κατακερματίζει τη λογική της ιστοσελίδας και διευκολύνει το έργο του προγραμματιστή. Η διευκόλυνση όμως αυτή την οποία παρέχει δημιουργεί και ένα από τα μειονεκτήματα του συγκεκριμένου framework, δεν επιτρέπει δηλαδή ελευθερία στον προγραμματιστή σχετικά με το πως πρέπει να 'γίνουν' τα πράγματα αλλά απαιτεί συμμόρφωση με τη λογική που προσφέρει. Αυτό μπορεί να καθυστερήσει πολύ την εκμάθηση και κατ'επέκταση την πραγματική υλοποίηση εφαρμογών. Όσον αφορά τις δυνατότητες που παρέχει, ενδεικτικά θα αναφερθούν οι *controllers*, οι οποίοι παρακολουθούν τυχόν αλλαγές στις μεταβλητές στο περιβάλλον στο οποίο είναι υπεύθυνοι, *services* και *factories* για επικοινωνία και μεταφορά δεδομένων ανάμεσα σε διαφορετικά μέρη της εφαρμογής καθώς και *directives* για δημιουργία νέων HTML attributes [18].

## 2.2.9 WebRTC

### 2.2.9.1 Σύντομη Ιστορική Προσέγγιση

Μία από τις τελευταίες μεγάλες προκλήσεις για το διαδίκτυο είναι η παροχή της δυνατότητας για ανθρώπινη επικοινωνία μέσω ήχου και βίντεο, κάτι που επικράτησε να λέγεται επικοινωνία πραγματικού χρόνου (Real-Time Communication - RTC). Οι οραματιστές της κοινότητας θεωρούσαν ανέκαθεν πως οι RTC δυνατότητες θα πρέπει να είναι εξίσου φυσικές για τις εφαρμογές ιστού, όπως είναι και η εισαγωγή κειμένου σε φόρμες. Χωρίς αυτές αισθάνονταν περιορισμό ως προς την ικανότητά τους να καινοτομούν και να εξελίσσουν νέους τρόπους για την αλληλεπίδραση των χρηστών.

Εντούτοις, ιστορικά, η επικοινωνία πραγματικού χρόνου επιτυγχανόταν μέσα σε επιχειρησιακά περιβάλλοντα, καθώς, λόγω αυξημένης πολυπλοκότητας, απαιτούσε την απόκτηση ακριβών και αδειοδοτημένων τεχνολογιών βίντεο και ήχου ή την εξέλιξή τους από την ίδια την επιχείρηση. Επιπρόσθετα, η ενοποίηση των RTC τεχνολογιών με το ήδη υπάρχον περιεχόμενο, δεδομένα και υπηρεσίες, υπήρξε δύσκολο και χρονοβόρο έργο, ειδικά στον ιστό.

Το Gmail video chat έγινε δημοφιλές το 2008 και το 2011 η Google εισήγαγε την εφαρμογή Hangouts, η οποία χρησιμοποιούσε την ίδια υπηρεσία με το Gmail, γνωστή ως Google Talk. Στη συνέχεια, η ίδια εταιρία αγόρασε τη GIPS. Η τελευταία είχε εξελίξει μια σειρά από συστατικά (components) απαραίτητα για RTC, όπως κωδικοποιητές και τεχνικές ακύρωσης ηχούς (echo cancellation techniques), τα οποία κατόπιν κατέστησε «ανοικτά» στο κοινό (open source). Συνεργάστηκε επίσης με τις αρμόδιες επιτροπές IETF [47] και W3C [32] για να επιτύχει την αποδοχή της βιομηχανίας. Το Μάιο του 2011, η Ericsson κατασκεύασε την πρώτη υλοποίηση του WebRTC (Web Real-Time Communication). Το WebRTC περιέχει σήμερα ανοικτά πρότυπα για επικοινωνία πραγματικού χρόνου μέσω βίντεο, ήχου και δεδομένων στο browser, χωρίς την ανάγκη επεκτάσεων.

### 2.2.9.2 Η Τεχνολογία WebRTC

Με τον όρο WebRTC καλείται ένα σύνολο από προδιαγραφές, πρωτόκολλα και JavaScript APIs, ο συνδυασμός των οποίων επιτρέπει τον peer-to-peer διαμοιρασμό ήχου, βίντεο και δεδομένων μεταξύ browsers (peers). Αντί να βασίζεται σε εξωτερικές επεκτάσεις (third-party plug-ins) ή κλειστό (proprietary) λογισμικό, το WebRTC μετατρέπει την επικοινωνία πραγματικού χρόνου σε ένα καθιερωμένο χαρακτηριστικό στο οποίο μπορεί να έχει πρόσβαση οποιαδήποτε εφαρμογή ιστού με τη μορφή τριών α-κόμα απλών και διαισθητικών JavaScript APIs.

Το πρώτο από αυτά ονομάζεται MediaStream (ευρύτερα γνωστό και ως getUserMedia) και αναπαριστά συγχρονισμένες ροές πολυμέσων που μπορούν να μεταδοθούν μεταξύ των peers. Μια τέτοια ροή (stream) μπορεί να αποτελείται από ένα ή περισσότερα κομμάτια, αν και συνήθως αποτελείται από ένα για το βίντεο και ένα για τον ήχο. Μια ροή μπορεί να μεταδώσει είτε «ζωντανά» πολυμέσα (για ηχητικές κλήσεις ή βιντεοδιασκέψεις) ή αποθηκευμένα (π.χ. ταινίες ή μουσική που βρίσκονται στον τοπικό σκληρό δίσκο).

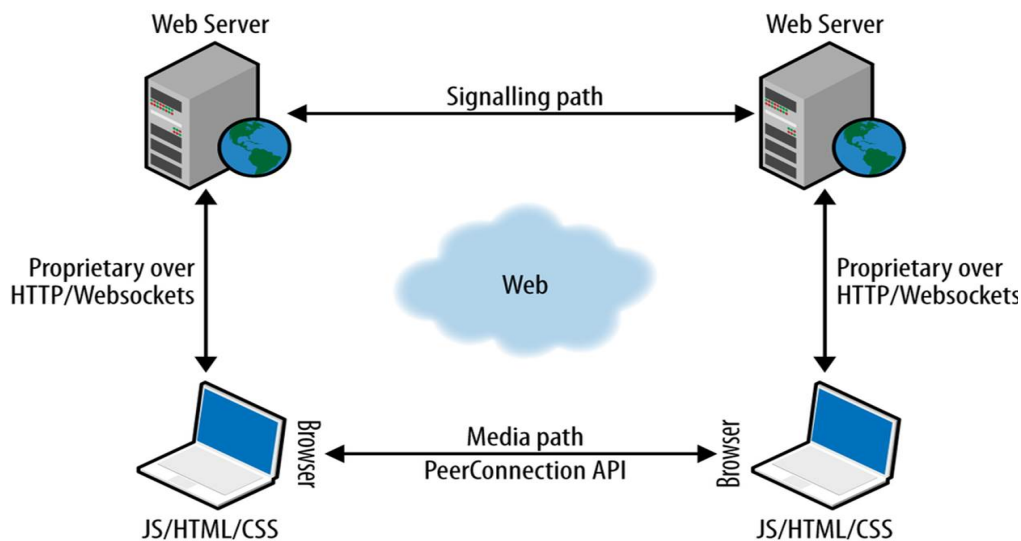
Το δεύτερο, και πιο σημαντικό ίσως, είναι το RTCPeerConnection, υπεύθυνο για τη σταθερή και αποδοτική μετάδοση των ροών δεδομένων μεταξύ των peers. Χάρη σε αυτό, οι προγραμματιστές κατά την ανάπτυξη των εφαρμογών δε χρειάζεται να ασχοληθούν με χαμηλού επιπέδου ζητήματα, όπως διαχείριση χαμένων.

Τέλος, εκτός από ήχο και βίντεο, το WebRTC, όπως προαναφέρθηκε, υποστηρίζει επίσης μετάδοση και άλλων τύπων δεδομένων σε πραγματικό χρόνο. Το σκοπό αυτό εξυπηρετεί το RTCDataChannel API. Επιτρέπει την αποστολή τυχαίων δεδομένων (arbitrary data) με χαμηλή καθυστέρηση (low latency) και υψηλή διεκπεραιωτικότητα. Μπορεί να χρησιμοποιηθεί μεταξύ άλλων για ανάπτυξη παιχνιδιών, εφαρμογών απομακρυσμένης επιφάνειας εργασίας, δωμάτια συνομιλιών, μεταφορά αρχείων μέχρι και επικοινωνία αποκεντρωποιημένων δικτύων.

### 2.2.9.3 (Σηματοδοσία) Signaling

Για να μπορέσει να υπάρξει επικοινωνία πραγματικού χρόνου ανάμεσα σε δύο υπολογιστές θα πρέπει να δημιουργηθούν τα δύο RTCPeerConnections και στη συνέχεια να υπάρξει η ανταλλαγή των απαραίτητων δεδομένων μεταξύ των δύο ισότιμων κόμβων (peers). Οι δύο αυτοί όμως peers δεν γνωρίζουν ο ένας για την

ύπαρξη του άλλου, ούτε υπάρχει κάποιος τρόπος ώστε να έρθουν σε επικοινωνία. Ο μηχανισμός που είναι υπεύθυνος για το συντονισμό και την εγκαθίδρυση των απομακρυσμένων συνδέσεων ονομάζεται Σηματοδοσία (Signaling). Ο παραπάνω μηχανισμός δεν παρέχεται από το WebRTC αλλά είναι στο χέρι του προγραμματιστή να εξετάσει τις επιλογές που έχει (Websockets, SIP, XMPP). Μόνο αφού ολοκληρωθεί η εγκαθίδρυση της επικοινωνίας μπορεί να ξεκινήσει η οποιαδήποτε αποστολή δεδομένων μεταξύ των δύο peers. Συνήθως το παραπάνω έργο αναλαμβάνει κάποιος server (STUN και TURN server) ο οποίος γνωρίζει τη διεύθυνση των δύο αυτών οντοτήτων (ή έμμεσων προορισμών) και μεταφέρει τα κατάλληλα μηνύματα. Αφού ολοκληρωθεί η διαδικασία, η επικοινωνία γίνεται μεταξύ των δυο peers και ο server δεν αναλαμβάνει κάποιον άλλον ρόλο.

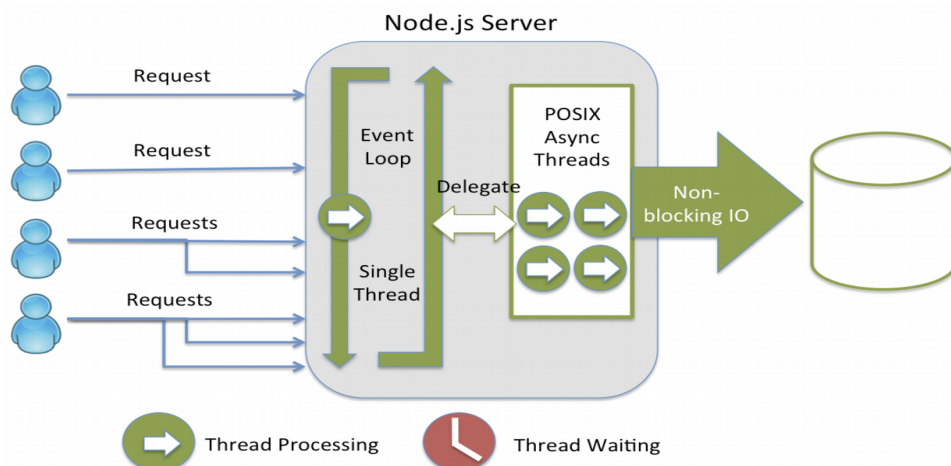


Σχήμα 14: Η διαδικασία Σηματοδότησης (Signaling) για την εγκαθίδρυση επικοινωνίας πραγματικού χρόνου

## 2.2.10 Node.js

Το Node.js είναι μια πλατφόρμα ανάπτυξης λογισμικού κυρίως για δημιουργία εξυπηρετητών (Web Servers) κτισμένη σε περιβάλλον Javascript το οποίο είναι βασισμένο πάνω στη μηχανή JavaScript V8 του Chrome (Chrome's V8 JavaScript engine) και αναπτύχθηκε το 2009 από την εταιρεία Joyent. Το κύριο μέλημα του δημιουργού του, Ryan Dalh, ήταν να βρεθεί ένας τρόπος, ο χρήστης να ενημερώνεται σε πραγματικό χρόνο για την κατάσταση κάποιου αρχείου που ανέβασε στο διαδίκτυο. Σε αντίθεση με τους παραδοσιακούς server που στηρίζονται στην πολυνηματικότητα, το Node.js ακολουθεί μια διαφορετική προσέγγιση: ασύγχρονη επικοινωνία εισόδου/εξόδου. Όπως και ο κώδικας που τρέχει σε έναν browser βασίζεται στα callbacks και στην ασύγχρονη εκτέλεσή τους. Έτσι, όταν ο επεξεργαστής περιμένει ένα γεγονός, ορίζεται μια συνάρτηση επιστροφής και μέχρι να ολοκληρωθεί μια διαδικασία ο επεξεργαστής μένει ανενεργός και ελεύθερος. Όταν ολοκληρωθεί το γεγονός, καλείται η συνάρτηση αυτή και κρατά τον επεξεργαστή κατειλημμένο μέχρι να ολοκληρωθεί ολόκληρη η διαδικασία. Για παράδειγμα, όταν ο Server λάβει ένα HTTP και πρέπει να

διαβάσει ένα αρχείο τότε αναθέτει στο λειτουργικό το άνοιγμα του αρχείου και μέχρι να ολοκληρωθεί η διαδικασία, ξεκινά να εξυπηρετεί το επόμενο request. Όταν ολοκληρωθεί επιτυχώς το άνοιγμα και η ανάγνωση του αρχείου, επιστρέφει το επιθυμητό αποτέλεσμα. Έτσι αυτό που πετυχαίνει είναι η ελαχιστοποίηση της αναμονής και η μικρή απαίτηση σε μνήμη. [21,22].



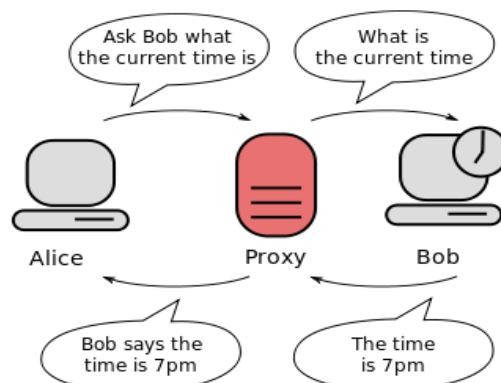
Σχήμα 15: Η λογική εξυπηρέτησης πολλών ταυτόχρονων αιτημάτων χωρίς παραλληλισμό.

Εκτός από τις πολύ καλές επιδόσεις που παρουσιάζει το Node.js, ένα επιπρόσθετο πλεονέκτημά του είναι η γλώσσα στην οποία είναι υλοποιημένο. Καθώς είναι γραμμένος σε Javascript, οι προγραμματιστές διαδικτυακών εφαρμογών (Web Developer) μπορούν να δημιουργήσουν ολόκληρα τα συστήματά τους σε μία μόνο γλώσσα προγραμματισμού, πράγμα που του επιτρέπει να στρέψει την προσοχή του σε θέματα διαφορετικά από την εκμάθηση διαφορετικών γλωσσών προγραμματισμού. Ένα επιπλέον πλεονέκτημα είναι η ύπαρξη μιας πολύ βολικής και εύχρηστης βιβλιοθήκης, η οποία αποτελείται από μια πληθώρα μικρών αρχείων (modules) που είναι συμβατά με το Node.js και με τη βοήθεια του διαχειριστή αρχείου του Node.js (NPM Node Package Manager) η διαχείριση και εγκατάσταση είναι πολύ εύκολη διαδικασία .

Παρ' όλη την αυξανόμενη δημοτικότητα και επίδοση του, το Node.js έχει κάποια μειονεκτήματα. Δεν επιτρέπει για παράδειγμα επεκτασιμότητα (scalability) καθώς δεν μπορεί να εκμεταλλευτεί τα πολυπύρηνά μηχανήματα της εποχής, ενώ εύκολα δημιουργούνται πολλαπλοί ορισμοί συναρτήσεων επιστροφής, γεγονός το οποίο μπορεί να δημιουργήσει προβλήματα. Επιπλέον ο Node.js δεν είναι ικανός για πολύ βαριές υπολογιστικά διαδικασίες αλλά ειδικεύεται σε λειτουργίες Εισόδου/Εξόδου (I/O).

## 2.2.11 Διακομιστής Μεσολάβησης και Πρωτόκολλο HTTPS

Ο διακομιστής μεσολάβησης είναι ένας εξυπηρετητής ο οποίος έχει στόχο να βελτιώσει την ταχύτητα πλοήγησης στο διαδίκτυο και παράλληλα να μειώσει την κίνηση του δικτύου προς αυτό. Η πιο συνηθισμένη χρήση του είναι να τοποθετείται μεταξύ ανάμεσα από τους χρήστες ενός δικτύου και του διαδικτύου. Έτσι η διαδικασία που πραγματοποιείται είναι η ακόλουθη: λαμβάνει αιτήματα ιστοσελίδων από έναν χρήστη, προσκομίζει τη σελίδα από το διαδίκτυο και έπειτα την επιστρέφει στον υπολογιστή που του τη ζήτησε. Μια εξίσου σημαντική χρήση του είναι ως μέρους ενός firewall που αποτρέπει hackers να χρησιμοποιήσουν το διαδίκτυο ώστε να αποκτήσουν πρόσβαση σε υπολογιστές ενός ιδιωτικού δικτύου. Η αποστολή ευαίσθητων δεδομένων καθώς και η ανάγκη ύπαρξης πρωτοκόλλου https για την πραγματοποίηση των βιντεοκλήσεων οδήγησε στην χρήση ενός proxy server [33].



Σχήμα 16: Διακομιστής Μεσολάβησης

### 2.2.11.1 Πρωτόκολλο HTTPS

Το HTTPS (*Hypertext Transfer Protocol Secure*) χρησιμοποιείται στην Πληροφορική για να δηλώσει μία ασφαλή δικτυακή σύνδεση http. Ένας σύνδεσμος (URL) που αρχίζει με το πρόθεμα https υποδηλώνει ότι θα χρησιμοποιηθεί κανονικά το πρωτόκολλο HTTP, αλλά η σύνδεση θα γίνει σε διαφορετική θύρα (443 αντί 80) και τα δεδομένα θα ανταλλάσσονται κρυπτογραφημένα. Το σύστημα αυτό σχεδιάστηκε αρχικά από την εταιρία Netscape Communications Corporation για να χρησιμοποιηθεί σε ιστοσελίδες (sites) όπου απαιτείται αυθεντικοποίηση χρηστών και κρυπτογραφημένη επικοινωνία. Σήμερα χρησιμοποιείται ευρέως στο διαδίκτυο όπου χρειάζεται αυξημένη ασφάλεια διότι διακινούνται ευαίσθητες πληροφορίες (πχ αριθμοί πιστωτικών καρτών, passwords κοκ)

Το HTTPS δεν είναι ξεχωριστό πρωτόκολλο, όπως νομίζουν μερικοί, αλλά αναφέρεται στον συνδυασμό του απλού HTTP πρωτοκόλλου και των δυνατοτήτων κρυπτογράφησης που παρέχει το πρωτόκολλο Secure Sockets Layer (SSL). Η κρυπτογράφηση που χρησιμοποιείται διασφαλίζει ότι τα κρυπτογραφημένα δεδομένα

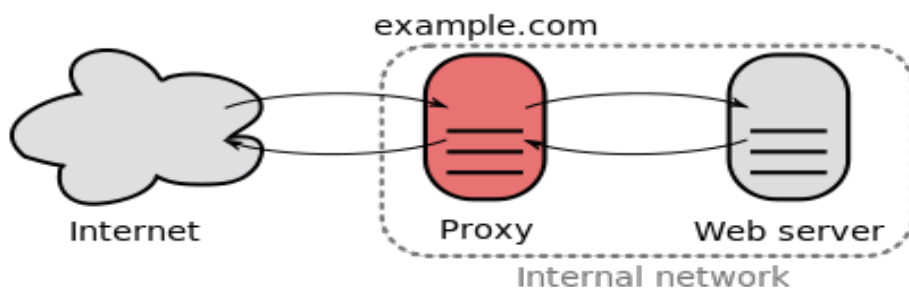


δεν θα μπορούν να υποκλαπούν από άλλους κακόβουλους χρήστες ή από επιθέσεις man-in-the-middle.

Για να χρησιμοποιηθεί το HTTPS σε έναν server, θα πρέπει ο διαχειριστής του να εκδώσει ένα πιστοποιητικό δημοσίου κλειδιού. Σε servers που χρησιμοποιούν το λειτουργικό σύστημα UNIX αυτό μπορεί να γίνει μέσω του προγράμματος OpenSSL. Στην συνέχεια το πιστοποιητικό αυτό θα πρέπει να υπογραφεί από μία αρχή πιστοποίησης (certificate authority), η οποία πιστοποιεί ότι ο εκδότης του πιστοποιητικού είναι νομότυπος και ότι το πιστοποιητικό είναι έγκυρο. Με τον τρόπο αυτό οι χρήστες μπορούν να δουν την υπογραφή της αρχής πιστοποίησης και να βεβαιωθούν ότι το πιστοποιητικό είναι έγκυρο και ότι κανένας κακόβουλος χρήστης δεν το έχει πλαστογραφήσει.[34]

### 2.2.11.2 Αντίστροφος Διακομιστής Μεσολάβησης (Reverse Proxy)

Ως αντίστροφος διακομιστής μεσολάβησης (*reverse proxy*) νοείται ο διακομιστής μεσολάβησης που υπάρχει πριν από κάποιον εξυπηρετητή ιστοσελίδων. Ο διακομιστής αυτός παραλαμβάνει όλα τα αιτήματα για τον διακομιστή των ιστοσελίδων και απαντά χρησιμοποιώντας προηγούμενα όμοια αιτήματα χωρίς να ξαναζητήσει τις σελίδες από τον εξυπηρετητή ιστοσελίδων. Επίσης μπορεί να χρησιμοποιηθεί για την κρυπτογράφηση των δεδομένων ώστε το πρόγραμμα κρυπτογράφησης να μην επιβαρύνει τον κύριο εξυπηρετητή. Ο αντίστροφος διακομιστής μεσολάβησης δηλαδή επικοινωνεί με τον κύριο εξυπηρετητή με το απλό πρωτόκολλο http και με τους πελάτες επικοινωνεί με το πιο περίπλοκο https [33].



Σχήμα 17: Αντίστροφος Διακομιστής Μεσολάβησης

### 2.2.11.3 Εξυπηρετητής Apache ως Reverse Proxy Server

Ο Apache HTTP, γνωστός και απλά ως Apache, είναι ένας εξυπηρετητής του παγκοσμίου ιστού(web). Όποτε ένας χρήστης επισκέπτεται έναν ιστότοπο, το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες, όπως τα Windows, το Linux, το UNIX και το MacOS. Κυκλοφόρησε υπό την άδεια λογισμικού Apache και είναι λογισμικού ανοικτού κώδικα Συντηρείται από μια κοινότητα ανοικτού κώδικα με

επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation). Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων π.χ. Oracle, MySQL. Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές, βασισμένους στο Unix, σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς, όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET. Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους. Τον Μάρτιο του 2017, το 49,48% του συνόλου των καταχωρισμένων Ελληνικών τομέων χρησιμοποιούσε το Apache [36].

Ο Server Apache θα χρησιμοποιηθεί στην πλατφόρμα ως ένας Reverse Proxy Server καθώς η ευρεία χρήση του έχει δημιουργήσει πολλές και καθοδηγητικές οδηγίες για την εγκατάσταση και χρήση του, ενώ ταυτόχρονα προσφέρει μεγάλη ευκολία και δεν απαιτεί σχεδόν καμία αλλαγή στον ήδη υπάρχοντα http server. Το παραπάνω θα δημιουργηθεί μέσω του παρεχόμενου module mod\_proxy. Για τη δημιουργία πιστοποιητικού θα χρησιμοποιηθεί η υπηρεσία certbot, η οποία δρα ως ένας πελάτης στο επίσημο 'Let's Encrypt' και επιτρέπει σε μια ιστοσελίδα να λειτουργεί ως https άμεσα. Επιπλέον η χρήση έχει βελτιστοποιηθεί σε συνδυασμό με εξυπηρετητές τύπου Apache [35].

## 2.2.12 Υπηρεσίες Rest (Representational State Transfer)

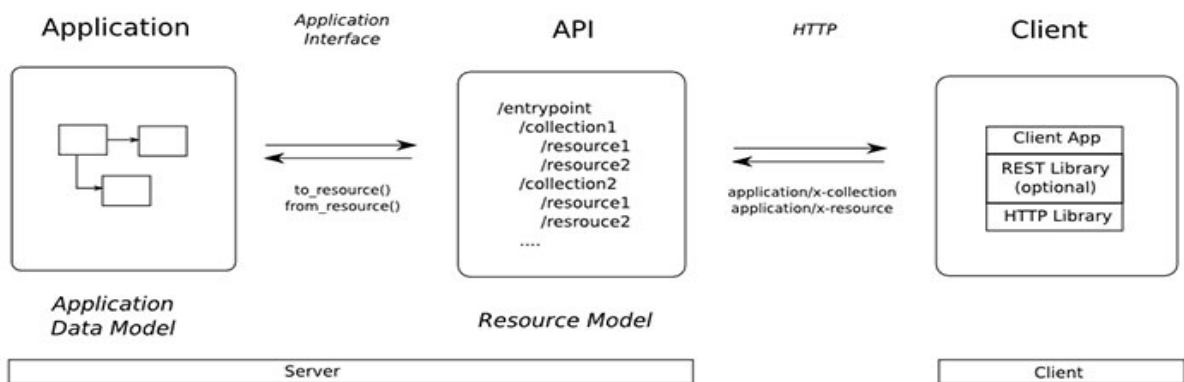
Το Rest (Representational State Transfer) είναι μια μορφή αρχιτεκτονικής για τη δημιουργία δικτυακών εφαρμογών που εισήχθη το 2000 από τον Roy Fielding. Δημιουργήθηκε για να υποστηρίξει τη διαλειτουργικότητα μεταξύ όλων των υπολογιστών του διαδικτύου. Ένας Rest Server επιτρέπει την πρόσβαση πόρων μέσω κάποιου αναγνωριστικού (π.χ. URL) και ο Rest Client απλώς λαμβάνει και τροποποιεί τα δεδομένα αυτά. Οι πόροι αυτοί αναπαρίστανται με διάφορες μορφές εκ των οποίων οι πιο διαδεδομένες είναι το κείμενο, JSON Objects, XML.

Αναλυτικότερα οι βασικές αρχές που διέπουν τη συγκεκριμένη αρχιτεκτονική είναι :

- **Ομοιόμορφη διεπαφή (Uniform Interface):** Αποτελεί θεμελιώδη αρχή οποιασδήποτε εφαρμογής του REST. Η ομοιόμορφη διεπαφή απλοποιεί και αποσυνδέει την αρχιτεκτονική πελάτη – διακομιστή, επιτρέποντας σε κάθε πλευρά να εξελιχθεί ανεξάρτητα.
- **Χωρίς κατάσταση (Stateless):** Κάθε αίτημα κάποιου πελάτη αντιμετωπίζεται ως ανεξάρτητο και περιέχει όλες τις απαραίτητες πληροφορίες που απαιτούνται για το χειρισμό του. Η κατάσταση συνεδρίας (session state) διατηρείται στον πελάτη και όχι στον εξυπηρετητή. Με αυτόν τον τρόπο απλοποιείται ο σχεδιασμός των εξυπηρετητών. Παράδειγμα stateless πρωτοκόλλων είναι το IP (internet protocol) και το HTTP (Hypertext Transfer Protocol).
- **Ένα σύστημα ιεραρχημένο με τη χρήση της cache μνήμης (Cacheable):** Πρέπει να ορίζεται αν οι αποκρίσεις μπορούν να «κρυφτούν» στη μνήμη cache ή

όχι, ώστε οι πελάτες να μην επαναχρησιμοποιούν άκυρα ή ακατάλληλα δεδομένα. Έτσι βελτιώνεται η επεκτασιμότητα και η απόδοση.

- **Διαστρωματωμένο σύστημα (Layered System):** Ο πελάτης δεν μπορεί να γνωρίζει εάν συνδέεται άμεσα με τον εξυπηρετητή ή με ένα ενδιάμεσο επίπεδο. Οι ενδιάμεσοι εξυπηρετητές μπορούν να βελτιώσουν την επεκτασιμότητα του συστήματος αλλά και την ασφάλειά του.
- **Κώδικας κατά ζήτηση (Code on Demand):** Οι διακομιστές έχουν τη δυνατότητα να παρατείνουν προσωρινά ή να προσαρμόζουν τη λειτουργία ενός πελάτη με τη μεταφορά εκτελέσιμου κώδικα.



Σχήμα 18: RESTful αρχιτεκτονική μιας εφαρμογής.

### 2.2.13 JSON Objects

Το JSON (JavaScript Object Notation) είναι μια απλή μορφή αναπαράστασης δεδομένων ως ζεύγη ιδιοτήτων/τιμών. Η σύνταξη του αποτελεί υποσύνολο της γλώσσας JavaScript, όμως κώδικας για δημιουργία και ανάλυση δομών JSON υποστηρίζεται από μεγάλο πλήθος γλωσσών προγραμματισμού. Χρησιμοποιείται ευρέως στην ασύγχρονη επικοινωνία μεταξύ browser και server και τείνει να αντικαταστήσει τη γλώσσα σήμανσης XML. Ένα αντικείμενο JSON περικλείεται σε αγκύλες `{ }` και αποτελείται από ένα σύνολο ζευγών ιδιότητας/τιμής. Σε κάθε τέτοιο ζεύγος, η ιδιότητα είναι πάντα τύπου String, ενώ η τιμή ανήκει σε έναν από τους τύπους δεδομένων που υποστηρίζει το JSON, οι οποίοι είναι οι αριθμοί (δε γίνεται διάκριση μεταξύ ακεραίων και δεκαδικών), Strings, Boolean, Arrays, Objects (συλλογή από ζεύγη ιδιοτήτων/τιμών) και το null, που αντιπροσωπεύει την κενή τιμή.

```

{ "users": [
  {
    "firstName": "Ray",
    "lastName": "Villalobos",
    "joined": {
      "month": "January",
      "day": 12,
      "year": 2012
    }
  },
  {
    "firstName": "John",
    "lastName": "Jones",
    "joined": {
      "month": "April",
      "day": 28,
      "year": 2010
    }
  }
]
}

```

*Σχήμα 19: Αναπαράσταση δεδομένων με σήμανση JSON.*

## 2.2.14 MongoDB

Η Mongo Database πρόκειται για μια βάση δεδομένων, η οποία είναι πλήρως εναρμονισμένη με τη φύση μια διαδικτυακής εφαρμογής. Είναι βάση δεδομένων που δεν υλοποιεί το σχεσιακό μοντέλο ενώ ταυτόχρονα δεν αποθηκεύει τα δεδομένα της σε πίνακα, όπως π.χ η SQL, αλλά σε JSON. Το παραπάνω είναι πολύ χρήσιμο εργαλείο αν αναλογιστεί κανείς ότι η JSON είναι μια μορφή δεδομένων που μπορεί να αποσταλεί χωρίς τροποποιήσεις μέσω του διαδικτύου. Στην παρούσα διπλωματική εργασία ως μορφή αποθήκευσης επιλέχτηκε η BSON (Binary JSON) η οποία υποστηρίζει και επιπλέον τύπους δεδομένων και είναι σχεδιασμένη για αποδοτικότερες λειτουργίες σάρωσης και αποθήκευσης.

## 2.2.15 OAuth 2.0 και Access Tokens

Το OAuth 2.0 πρόκειται για ένα πρωτόκολλο πιστοποίησης που χρησιμοποιείται στην βιομηχανία. Πρόκειται για τη δεύτερη έκδοση του πρωτοκόλλου OAuth που αρχικά δημιουργήθηκε το 2006. Συγκεκριμένα το πρωτόκολλο αυτό επιτρέπει σε τρίτες οντότητες (Applications) περιορισμένη πρόσβαση σε κάποιο HTTP service, είτε εκ μέρους του ιδιοκτήτη του πόρου (Resource owner), δημιουργώντας μια διεπαφή μεταξύ του ιδιοκτήτη και του HTTP service, είτε εκ μέρους της ίδιας της τρίτης οντότητας. Οι ρόλοι που εμπλέκονται στο παραπάνω πρωτόκολλο είναι οι εξής τέσσερις:

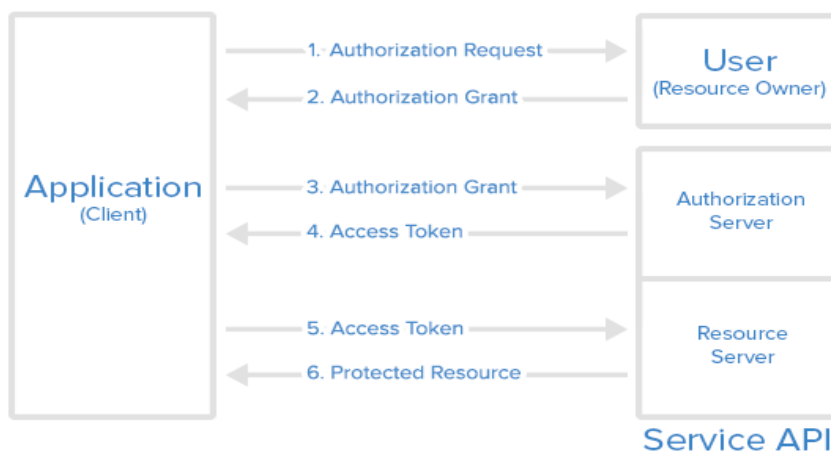
- **Ο Ιδιοκτήτης του Πόρου (Resource Owner):** Πρόκειται για μια οντότητα η οποία είναι σε θέση να επιτρέπει πρόσβαση σε έναν προστατευμένο πόρο. Εάν η οντότητα είναι κάποιο φυσικό πρόσωπο τότε ονομάζεται τελικός χρήστης (End User). Με τον όρο “προστατευμένος πόρος” εννοούμε ότι κάποιος χρήστης για να

αποκτήσει πρόσβαση στον πόρο αυτό θα χρειαστεί να του δοθεί άδεια από μία τρίτη αρχή.

- **Εξυπηρετητής του Πόρου (Resource Owner):** Πρόκειται για τον Εξυπηρετητή στην επίβλεψη του οποίου βρίσκεται ο προστατευμένος πόρος και είναι σε θέση να επιτρέψει ή να αποτρέψει και να απαντά κατάλληλα με την χρήση σύμβολο πρόσβασης (Access Token).
- **Πελάτης (Client):** Είναι η εφαρμογή η οποία ζητά πρόσβαση στον πόρο εκ μέρους του Ιδιοκτήτη του πόρου με τα στοιχεία εξουσιοδότησης του ιδιοκτήτη (συνήθως ένας συνδυασμός Username – Password). Η συγκεκριμένη οντότητα μπορεί να είναι κάποιος Server, Desktop Application ή άλλο μηχάνημα.
- **Εξυπηρετητής Εξουσιοδότησης (Authorization Server):** Ο Server ο οποίος δημιουργεί και επιστρέφει τα Access Tokens στον Πελάτη αν το αίτημα πρόσβασης του πελάτη εκ μέρους του ιδιοκτήτη ολοκληρωθεί με επιτυχία.

### 2.2.15.1 Ροή του Πρωτοκόλλου.

#### Abstract Protocol Flow



Σχήμα 20: Ροή Πρωτοκόλλου του OAuth 2.0.

Οι τέσσερις οντότητες του πρωτοκόλλου, όπως περιγράφηκαν πριν, αλληλεπιδρούν μεταξύ τους με τον εξής τρόπο:

- Ο πελάτης ζητά εξουσιοδότηση από τον ιδιοκτήτη. Η αίτηση αυτή είτε γίνεται άμεσα στον ιδιοκτήτη είτε έμμεσα μέσω του εξυπηρετητή εξουσιοδότησης (προτιμητέο για αυξημένη ασφάλεια).
- Ο πελάτης λαμβάνει μια εξουσιοδότηση (authorization grant), η οποία πρόκειται για το πιστοποιητικό που αναπαριστά τον ιδιοκτήτη και το οποίο μπορεί να εκφραστεί με τέσσερις τρόπους. Ο τρόπος που θα επιλεγεί ποικίλλει ανάλογα με ποιός πελάτης ζητά εξουσιοδότηση και από τις προδιαγραφές που υποστηρίζει ο εξυπηρετητής εξουσιοδότησης.

- Ο πελάτης στη συνέχεια παρουσιάζει τα πιστοποιητικά που του δόθηκαν από τον ιδιοκτήτη στον εξυπηρετητή εξουσιοδότησης και ζητά ένα σύμβολο πρόσβασης (Access token).
- Ο εξυπηρετητής πιστοποιεί τόσο το συγκεκριμένο πελάτη που κάνει την αίτηση όσο και τα πιστοποιητικά του ιδιοκτήτη που του δόθηκαν και αν όλα είναι έγκυρα, δημιουργεί και επιστρέφει ένα σύμβολο πρόσβασης.
- Ο πελάτης από εδώ και στο εξής ζητά τον προστατευμένο πόρο που έχει υπό την επίβλεψή του ο εξυπηρετητής εξουσιοδότησης, παρουσιάζοντας απλά το σύμβολο πρόσβασης.
- Όσο ο εξυπηρετητής πιστοποιεί το σύμβολο πρόσβασης αυτό, οι αιτήσεις γίνονται δεκτές και ο πελάτης αποκτά πρόσβαση στον πόρο.

#### **2.2.15.2 Authorization Grant**

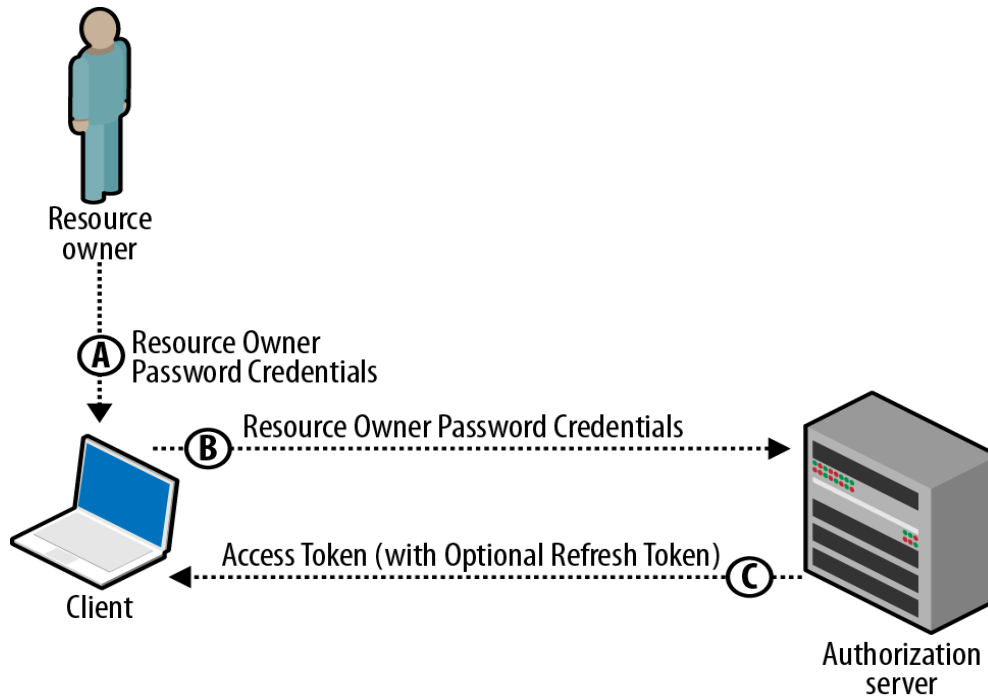
Το πιστοποιητικό εξουσιοδότησης (Authorization Grant) είναι μια απόδειξη η οποία αντιπροσωπεύει τον ιδιοκτήτη του πόρου (με σκοπό να αποκτήσει πρόσβαση στους πόρους του) και χρησιμοποιείται με σκοπό να αποκτηθεί το σύμβολο πρόσβασης. Το πρωτόκολλο OAuth 2.0 υποστηρίζει τέσσερις τύπους πιστοποιητικών. Η εν λόγω εφαρμογή έκανε χρήση του τύπου *Resource Owner Password Credential* οπότε και θα αναλυθεί περιληπτικά.

#### **2.2.15.3 Σύμβολο Πιστοποίησης (Access token)**

Τα σύμβολα πιστοποίησης είναι πιστοποιητικά τα οποία ένας πελάτης μπορεί να χρησιμοποιήσει για να αποκτήσει πρόσβαση σε προστατευμένους πόρους. Συγκεκριμένα είναι μία συμβολοσειρά χωρίς κάποια σημασία σε κανέναν εκτός από τον εξυπηρετητή. Πέραν του ότι είναι μικρά σε μέγεθος και δεν προσθέτουν καθυστερήσεις, εμποδίζουν να χρησιμοποιούνται τα πιστοποιητικά του ιδιοκτήτη συνεχώς (username και password) και έτσι αν πέσει σε λάθος χέρια δεν τα αποκτά ο κακόβουλος χρήστης. Επιπλέον έχουν περιορισμένη πρόσβαση και συνήθως έχουν ημερομηνία λήξης, ώστε όταν παρέλθει, το σύμβολο αυτό δε μπορεί να ξαναχρησιμοποιηθεί αλλά πρέπει να γίνει αίτηση για ανανέωση της ημερομηνίας (Refresh Token).

#### **2.2.15.4 Resource Owner Password Credential**

Ο συγκεκριμένος τύπος πιστοποίησης χρησιμοποιείται ως εξής: Ο πελάτης παρουσιάζει τα πιστοποιητικά του ιδιοκτήτη του πόρου (δηλαδή το username και το password) καθώς και τα δικά του πιστοποιητικά και αποκτά άμεσα το σύμβολο πρόσβασης. Επειδή ο πελάτης χρησιμοποιεί και γνωρίζει τα πιστοποιητικά του χρήστη η περίπτωση όπου ενδείκνυται να χρησιμοποιείται αυτός ο τύπος, είναι όταν υπάρχει εμπιστοσύνη μεταξύ του ιδιοκτήτη του πόρου και του πελάτη – εφαρμογή. Παρ' όλο που γνωρίζει τα πιστοποιητικά του ιδιοκτήτη, ο πελάτης τα χρησιμοποιεί μόνο μία φορά, την πρώτη, όπου και ζητά το σύμβολο πιστοποίησης και έτσι δεν χρειάζεται η αποθήκευση και η συνεχής έκθεσή τους στο διαδίκτυο.



Σχήμα 21: Πιστοποίηση τύπου Resource Owner Password Credential.

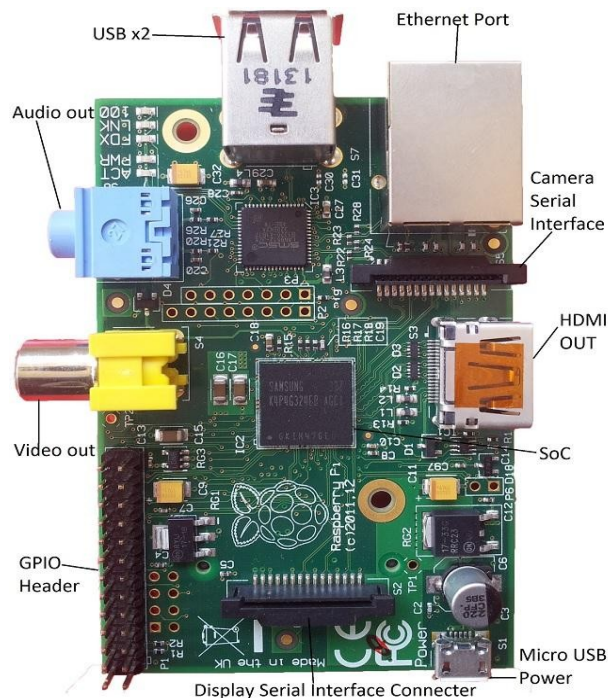
Η ροή του τύπου αυτού είναι η εξής:

- Ο Ιδιοκτήτης παρέχει το username και το password στον πελάτη.
- Ο πελάτης ζητά ένα σύμβολο πιστοποίησης από τον εξυπηρετητή πιστοποίησης, ο οποίος είθισται να είναι ίδιος με τον εξυπηρετητή του πόρου, παρουσιάζοντας τόσο τα δικά του χαρακτηριστικά (ο εξυπηρετητής πρέπει να ξέρει ποιος πελάτης είναι και αν του επιτρέπεται να εισέλθει στον πόρο ) όσο και του ιδιοκτήτη.
- Αν ο εξυπηρετητής επικυρώσει και τα δύο, επιστρέφει στον πελάτη ένα σύμβολο πρόσβασης.

## 2.2.16 Raspberry-Pi

Το Raspberry-Pi πρόκειται για ένα οικονομικό, στο μέγεθος μιας πιστωτική κάρτας, υπολογιστή ο οποίος χρειάζεται μια οθόνη υπολογιστή ή τηλεόραση, καθώς και ένα πληκτρολόγιο και ένα ποντίκι. Κατασκευάστηκε από το πανεπιστήμιο του Cambridge και εκδόθηκε από την Raspberry Pi Foundation το 2012. Είναι ένα πολύ ικανό μηχάνημα και επιτρέπει σε διάφορους ανθρώπους να έρθουν σε μια επαφή με τον προγραμματισμό. Είναι σε θέση να φέρει εις πέρας οτιδήποτε μπορεί να κάνει ένας σύγχρονος υπολογιστής, από την περιήγηση στο διαδίκτυο και την αναπαραγωγή υψηλής ποιότητας video έως και την χρήση βιντεοπαιχνιδιών, ενώ επιτρέπει χαμηλή κατανάλωση ενέργειας. Ως κανονικός υπολογιστής χρειάζεται ένα λειτουργικό σύστημα. Το λειτουργικό σύστημα που προτείνει η Raspberry Pi Foundation είναι το Raspbian, ένα Linux λειτουργικό βασισμένο στο Debian, αλλά είναι σε θέση να υποστηρίξει και άλλα λειτουργικά συστήματα. Σε όλα τα παραπάνω έρχεται να προστεθεί η σχετικά μεγάλη υπολογιστική του ικανότητα του σχετικά με τον ανταγωνισμό καθώς και η μικρή κατανάλωση ενέργειας.

Στην παρούσα διπλωματική εργασία το Raspberry-Pi θα παρέχεται στους ασθενείς μαζί με μια οθόνη, πληκτρολόγιο, ποντίκι και κάμερα, ενώ το ενσωματωμένο bluetooth που έχει θα επικοινωνεί με το οξύμετρο και η σύνδεσή του με το Internet θα πραγματοποιείται ασύρματα μέσω κάποιου WiFi.



Σχήμα 22: Raspberry Pi



## 2.2.17 Bluetooth Low Energy (Ble)

Στην παρούσα διπλωματική εργασία, η αποστολή των δεδομένων που ανιχνεύονται από το παλμικό οξύμετρο στέλνονται μέσω της τεχνολογίας Bluetooth Low Energy (Ble) στον server που είναι υπεύθυνος για τους ασθενείς και στη συνέχεια γίνονται ορατά στην οθόνη των ασθενών ή όπου αυτοί τα στέλνουν.

Το Bluetooth είναι ένα βιομηχανικό πρότυπο για ασύρματα προσωπικά δίκτυα υπολογιστών (Wireless Personal Area Networks, WPAN). Πρόκειται για μια ασύρματη τηλεπικοινωνιακή τεχνολογία μικρών αποστάσεων, η οποία μπορεί να μεταδώσει σήματα μέσω μικροκυμάτων σε ψηφιακές συσκευές. Επομένως, το Bluetooth είναι ένα πρωτόκολλο το οποίο παρέχει προτυποποιημένη, ασύρματη επικοινωνία ανάμεσα σε κινητά τηλέφωνα, φορητούς υπολογιστές, προσωπικούς υπολογιστές, εκτυπωτές, καθώς και ψηφιακές φωτογραφικές μηχανές ή ψηφιακές κάμερες, μέσω μιας ασφαλούς, φθηνής και παγκοσμίως διαθέσιμης, χωρίς ειδική άδεια, ραδιοσυχνότητας μικρής εμβέλειας. Αυτό που επιτυγχάνεται, είναι η «διασύνδεση» μεταξύ των προαναφερθεισών συσκευών χωρίς, όμως, να χρειάζεται η παρουσία καλωδίων, με χαμηλό κόστος και χαμηλή κατανάλωση ισχύος. Πρώτη η Ericsson δημοσίευσε αυτό το πρότυπο το 1999 καθότι μέχρι τότε δεν υπήρχε κάποιο ευρέως αποδεκτό πρότυπο για ασύρματα προσωπικά δίκτυα υπολογιστών (Wireless Personal Area Networks, WPAN). Το Bluetooth λειτουργεί στο «αδέσμευτο» φάσμα συχνοτήτων των 2,4 GHz, ώστε οι συσκευές που ενσωματώνουν την τεχνολογία αυτή να μπορούν να λειτουργήσουν απροβλημάτιστα σε οποιοδήποτε σημείο του πλανήτη.

Η τεχνολογία Bluetooth Low Energy (Ble), ή αλλιώς Bluetooth Smart ή αλλιώς Έκδοση (Version) 4.0+ του εγγράφου προδιαγραφών Bluetooth, είναι ένα υποσύνολο της κλασικής Bluetooth τεχνολογίας, το οποίο, όμως, έχει τα πλεονεκτήματα της χαμηλότερης κατανάλωσης ενέργειας και έτσι μπορεί να «τρέχει» για την ανάπτυξη εφαρμογών σε συσκευές που λειτουργούν με τυπικές μπαταρίες σε σχήμα νομίσματος (coin cell batteries) έως και για 2 χρόνια και τυποποιημένης αρχιτεκτονικής, που οδηγεί σε χαμηλό κόστος.

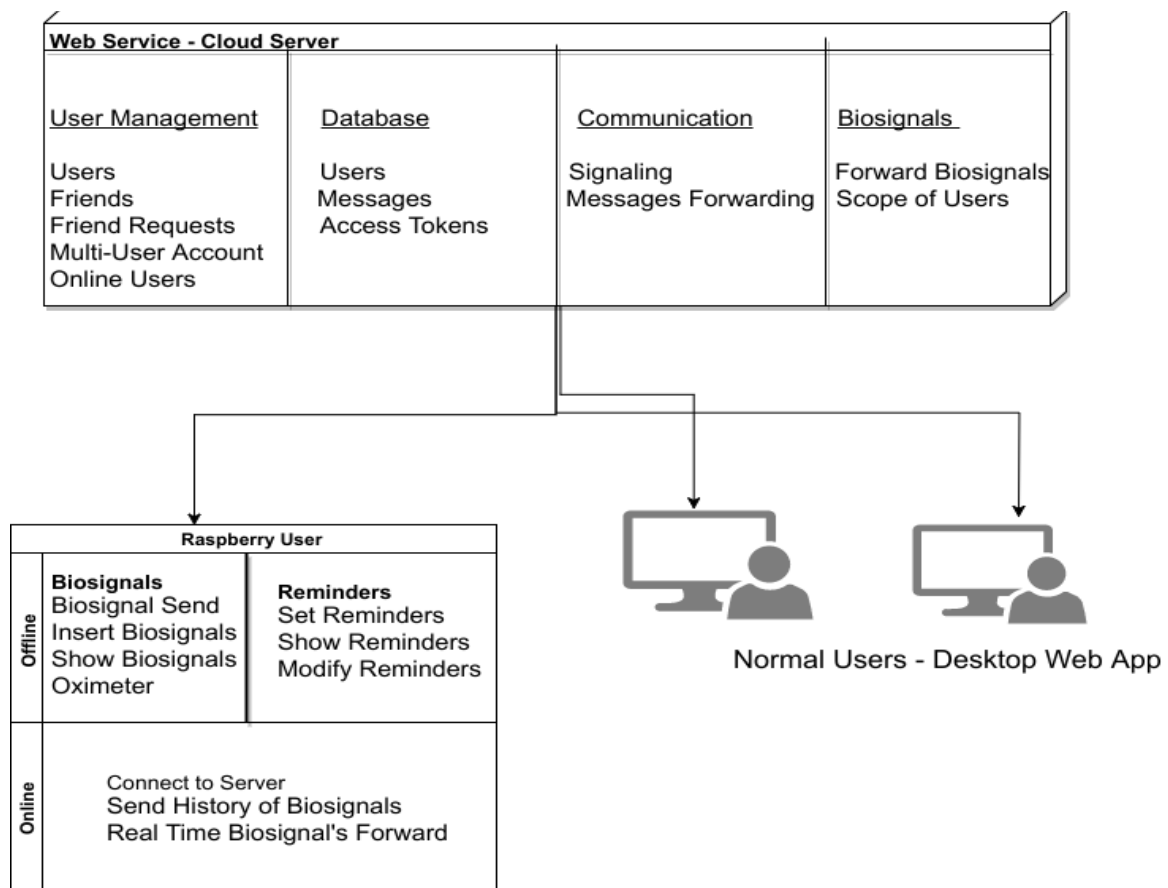
Η τεχνολογία αυτή προτάθηκε πρώτα από τη Nokia το 2006 ως μέρος του εσωτερικού της έργου «Wibree», υιοθετήθηκε, όμως, αργότερα από την εταιρεία Bluetooth SIG το 2010. Στήριξη για το Ble προσφέρουν οι περισσότερες πλατφόρμες σήμερα ]:

- iOS5+ (προτιμότερα iOS7+)
- Android 4.3+ (πολλά σφάλματα διορθώθηκαν στην έκδοση 4.4+)
- Apple OS X 10.6+
- Windows 10,8 (XP, Vista and 7 υποστηρίζουν μόνο το Bluetooth 2.1)
- GNU/Linux Vanilla BlueZ 4.93+

Η εταιρεία Bluetooth SIG προβλέπει ότι μέχρι το 2018 περισσότερα από το 90% των έξυπνων τηλεφώνων με Bluetooth θα υποστηρίζουν το Bluetooth Smart [28,29,30].

### 3 Ανάλυση και Σχεδιασμός του Συστήματος

Η παρούσα διπλωματική εργασία έχει ως σκοπό την δημιουργία μιας πλατφόρμας, βασισμένης πάνω σε browser, η οποία θα προσφέρει μια ολοκληρωμένη εμπειρία στον ασθενή – χρήστη της. Με τον όρο “ολοκληρωμένη” εννοούμε ότι ο ασθενής θα μπορεί να λαμβάνει μόνος του μετρήσεις, να παρακολουθεί το ιστορικό του, να θέτει υπενθυμίσεις και ταυτόχρονα θα του παρέχεται ένα κοινωνικό δίκτυο μέσα από το οποίο θα μπορεί να συνομιλεί εύκολα με τους δικούς του ανθρώπους ή/και τους γιατρούς του.



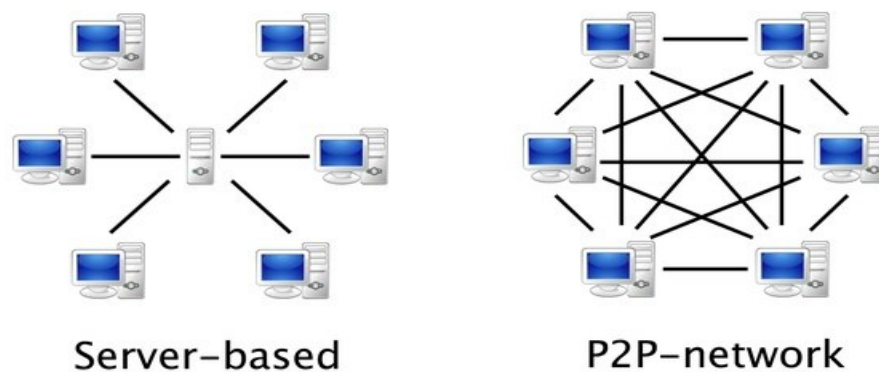
Σχήμα 23: Το Προτεινόμενο Σύστημα

#### 3.1 Γενική Περιγραφή

Σκοπός της παρούσας διπλωματικής εργασίας, είναι η δημιουργία και η συνεργασία δύο ειδών SPA εφαρμογών μέσω του διαδικτύου. Το πρώτο μέρος πρόκειται για μια SPA εφαρμογή, την οποία χρησιμοποιούν οι απλοί χρήστες μέσω του browser. Υπεύθυνος για την εξυπηρέτησή τους είναι ο κεντρικό Server. Το δεύτερο μέρος της εφαρμογής πρόκειται και πάλι για μία SPA εφαρμογή η οποία αντιπροσωπεύει τους ασθενείς και εξυπηρετείται τοπικά (localhost) από έναν δεύτερο Server όσον αφορά τις

Offline λειτουργίες, ενώ για τις Online λειτουργίες συνεργάζονται οι δύο παραπάνω εξυπηρετητές. Το περιβάλλον θα είναι και πάλι ο browser, αλλά αυτή τη φορά στη συσκευή Raspberry.

Η δόμηση της πλατφόρμας είναι στηριγμένη σε έναν κεντρικό εξυπηρετητή (Server Based Application) Παρόλο που ο κεντρικός εξυπηρετητής αναλαμβάνει ένα σημαντικό μέρος των υπολογιστικών εργασιών, εκμεταλλευόμενος τις δυνατότητες των σύγχρονων υπολογιστών, οι SPA εφαρμογές που τρέχουν στα μηχανήματα των χρηστών εκτελούν σημαντικό ποσοστό των διεργασιών που απαιτούνται και έτσι ο Server έχει λιγότερες αρμοδιότητες, επιτυγχάνοντας έτσι πιο γρήγορη απόκριση στο σύστημα ενώ η επικοινωνία με τον Server μειώνεται στον ελάχιστο δυνατό χρόνο. Έτσι αν και το σύστημα δεν μπορεί να διαθέτει λογική δικτύωσης peer-to-peer, ο κεντρικός εξυπηρετητής χρησιμοποιείται όσο λιγότερο γίνεται και έτσι οι καθυστερήσεις λόγω overhead και κατάληψη μεγάλου εύρους ζώνης είναι περιορισμένες.



Σχήμα 24: Διαφορετικές τοπολογίες ενός δικτύου.

## 3.2 Απαιτήσεις

### 3.2.1 Λειτουργικές Απαιτήσεις

Για να μπορεί να λειτουργήσει ομαλά η εφαρμογή και όλες οι υπηρεσίες τις οποίες μπορεί να προσφέρει, υπάρχουν κάποιες λειτουργικές απαιτήσεις που θα πρέπει να εκπληρώνονται, δηλαδή κάποια χαρακτηριστικά που το σύστημα οφείλει να έχει ώστε να φέρνει εις πέρας του στόχους για τους οποίους κατασκευάστηκε. Περιληπτικά θα αναφέρουμε τις πιο σημαντικές απαιτήσεις οι οποίες θα αναλυθούν στα επόμενα κεφάλαια.

Το μέρος της εφαρμογής που αφορά τους χρήστες (το κοινωνικό δίκτυο και τις υπηρεσίες του) θα πρέπει να υποστηρίζει τις παρακάτω λειτουργίες διασύνδεσης των χρηστών:

- Εγγραφή του χρήστη στο σύστημα.

- Σύνδεση του χρήστη στο σύστημα.
- Αιτήματα φιλίας (Friend Requests): Δύο συνδεδεμένοι χρήστες μπορούν να επικοινωνήσουν μόνο αν συμφωνήσουν και οι δύο. Το παραπάνω πρωτόκολλο έχει οριστεί μέσω της φιλίας, δηλαδή ο Α χρήστης στέλνει ένα αίτημα φιλίας στον Β χρήστη και αν ο τελευταίος το αποδεχθεί, τότε γίνονται φίλοι και μπορούν να επικοινωνούν μεταξύ τους.
- Εύκολη αναζήτηση άλλων χρηστών που είναι εγγεγραμμένοι στο σύστημα με σκοπό τη δημιουργία φιλίας.
- Διαγραφή φίλου.
- Εμφάνιση εικονιδίων των φίλων του χρήστη, όταν αυτός εισέρχεται στο σύστημα καθώς και την κατάσταση των φίλων αυτών (συνδεδεμένοι ή όχι).
- Επικοινωνία μεταξύ των χρηστών μέσω εικόνας, ήχου αλλά και γραπτών μηνυμάτων.
- Δυνατότητα σε έναν χρήστη να συνδεθεί στον λογαριασμό του σε παραπάνω του ενός μηχανήματα καθώς και την ομαλή λειτουργία, ενημέρωση όλων των συνδέσεων αυτών.
- Αιτήματα για λήψη βιοσημάτων από τους ασθενείς καθώς και απεικόνιση αυτών σε διαγράμματα για εύκολη και άμεση παρατήρηση (Biosignals Charts).
- Ενημερώσεις των χρηστών σε πραγματικό χρόνο για την κατάσταση των παραπάνω υπηρεσιών.
- Εμφάνιση των μετρήσεων που λαμβάνει ο ασθενής κάποια δεδομένη στιγμή κατά τη διάρκεια μιας βίντεο κλήσης σε πραγματικό χρόνο για εύκολη εποπτεία.
- Αποσύνδεση του χρήστη από το σύστημα.

Όσον αφορά τους χρήστες – ασθενείς του συστήματος, το online μέρος έχει τις ίδιες απαιτήσεις με τους απλούς χρήστες με δύο απλές διαφορές:

- Ένας ασθενής δεν μπορεί να ζητήσει τα δεδομένα από άλλους ασθενείς.
- Ο ασθενής διαθέτει μία λίστα με φίλους στους οποίους επιτρέπει να δουν τις μετρήσεις που έχει αποθηκεύσει στην βάση δεδομένων.

Το *Offline* μέρος θα πρέπει να πληροί τις παρακάτω απαιτήσεις:

- Ορισμός και επεξεργασία υπενθυμίσεων με δυνατότητα επανάληψης ημερησίως ή εβδομαδιαίως.
- Διαδραστικά διαγράμματα στα οποία προβάλλεται το ιστορικό των μετρήσεων του χρήστη και μέσω των οποίων μπορεί να επεξεργαστεί τις μετρήσεις ή να τις διαγράψει. Η προβολή των μετρήσεων μπορεί να περιλαμβάνει όλες τις μετρήσεις ή τις μετρήσεις της παρούσας ημέρας, των τριών προηγούμενων ημερών ή της τελευταίας εβδομάδας.
- Προβολή των μετρήσεων σε άξονες που λαμβάνει ο χρήστης από το οξύμετρο σε πραγματικό χρόνο καθώς και αποθήκευσης τους στη βάση δεδομένων. Ειδικός αλγόριθμος αποφασίζει πότε οι ενδείξεις του οξυμέτρου είναι σταθερές, οπότε και ενημερώνει τον χρήστη να αφαιρέσει το οξύμετρο.
- Δυνατότητα προσθήκης μέτρησης χειροκίνητα.

- Δυνατότητα αυτόματης ή μη σύνδεσης του χρήστη στο Online σύστημα κάθε φορά που εκκινεί η εφαρμογή.
- Δυνατότητα αλλαγής του κωδικού πρόσβασης του ασθενή.

### 3.2.2 Μη Λειτουργικές Απαιτήσεις

- **Στόχος Συστήματος:** Το σύστημα μπορεί να χρησιμοποιηθεί εύκολα από άπειρους χρήστες (Το UI είναι απλό και με λίγα κουμπιά) και είναι οργανωμένο έτσι ώστε να ελαχιστοποιούνται τα λάθη χρηστών (αποκλεισμός πολλαπλών αιτήσεων πριν ληφθεί απάντηση, προβολή ειδικού χωρίου για επιβεβαίωση πριν τη λήψη κάποιας απόφασης, ενημέρωση για την δυνατότητα/κατάσταση κάποιας αίτησης).
- **Ταχύτητα Συστήματος:** Η λογική του παρόντος συστήματος βασίζεται στην ασύγχρονη ενημέρωση και έτσι η απουσία polling (ερώτηση ανα τακτά χρονικά διαστήματα του server αν υπήρξε κάποια αλλαγή), καθιστά το σύστημα γρήγορο και αποδοτικό.
- **Ασφάλεια Συστήματος:** Η επικοινωνία του χρήστη με το σύστημα γίνεται πάνω από το πρωτόκολλο TLS (Transport Layer Security) και έτσι όλα τα μηνύματα κρυπτογραφούνται πριν προωθηθούν στο διαδίκτυο. Επιπλέον, οι κωδικοί των χρηστών είναι κρυπτογραφημένοι στη βάση δεδομένων του κεντρικού εξυπηρετητή και δεν υπάρχει η δυνατότητα αποκρυπτογράφησης τους.

### 3.3 Οι Χρήστες του συστήματος

Το σύστημα υποστηρίζει δύο είδη συνδεδεμένων χρηστών. Η πρώτη κατηγορία είναι οι απλοί χρήστες οι οποίοι συνδέονται στο σύστημα με σκοπό να επικοινωνήσουν με τους ασθενείς ή με άλλους απλούς χρήστες. Η δεύτερη κατηγορία είναι οι χρήστες ασθενείς. Πιο αναλυτικά έχουμε:

- **Οι ασθενείς:** Πρόκειται για τα φυσικά πρόσωπα του συστήματος που θα έχουν στη διάθεσή τους το Raspberry και στους οποίους παρέχονται δύο είδη λειτουργίας:
  - Online λειτουργία: Οι ασθενείς συνδέονται μέσω του διαδικτύου στο δίκτυο της εφαρμογής και μπορούν να συνομιλούν με άλλου συνδεδεμένους χρήστες και να προωθούν τις μετρήσεις τους.
  - Offline λειτουργία: Δεν απαιτεί σύνδεση στο διαδίκτυο και αποτελεί το μέρος της εφαρμογής που θα έχει να κάνει με την παρακολούθηση του ασθενή, δηλαδή τη λήψη μετρήσεων και την προβολή των μετρήσεων σε διαδραστικά διαγράμματα, καθώς και τον ορισμό υπενθυμίσεων. Μπορεί να θεωρηθεί ως το native μέρος της εφαρμογής των ασθενών.
- **Οι απλοί χρήστες:** Πρόκειται για τα φυσικά πρόσωπα που θα πρέπει να εγγραφούν στο σύστημα, προκειμένου να είναι σε θέση να επικοινωνήσουν είτε μεταξύ τους, είτε με κάποιον ασθενή (Γιατροί, Συγγενείς κλπ). Η σύνδεση τους στο σύστημα γίνεται μέσω του browser.

## 3.4 Αρχιτεκτονική

Σύμφωνα με το σχήμα το σύστημα για τη λειτουργία του βασίζεται σε δύο ουσιώδη υποσυστήματα: Τον κεντρικό Server (Cloud Server) και τον τοπικό Server των Raspberry. Το πρώτο είναι υπεύθυνο για την πιστοποίηση και εξουσιοδότηση και την επικοινωνία των απλών χρηστών και των ασθενών, ενώ το δεύτερο είναι υπεύθυνο για την εξυπηρέτηση των ασθενών του συστήματος. Οι πρώτοι έχουν πρόσβαση στην εφαρμογή ιστού από οποιαδήποτε συσκευή διαθέτει κάποιο browser, ενώ οι τελευταίοι εισέρχονται στο σύστημα μέσω του Raspberry Pi.

### 3.4.1 Κεντρική Υποδομή και οι Υπηρεσίες της

Ο Κεντρικός Εξυπηρετητής (Cloud Server) πρόκειται για το βασικό συστατικό (component) του συστήματος στον οποίο συνδέονται όλοι οι επιμέρους χρήστες του συστήματος. Οι κύριες αρμοδιότητες του Server είναι η εγγραφή και η αυθεντικοποίηση (authentication) των χρηστών, η σύνδεσή τους στο σύστημα, η εξυπηρέτηση των αιτημάτων τους, η επικοινωνία μεταξύ των χρηστών, καθώς και τη σηματοδότηση που απαιτείται με σκοπό να δημιουργηθούν οι κατάλληλες peer-to-peer WebRTC κλήσεις.

#### 3.4.1.1 Δημιουργία λογαριασμού – Σύνδεση στο σύστημα

Οι χρήστες για να μπορούν να χρησιμοποιήσουν τις υπηρεσίες που παρέχει ο Server θα πρέπει να έχουν δημιουργήσει έναν λογαριασμό. Για να συνδεθεί ένας χρήστης στο σύστημα απαιτούνται τα παρακάτω :

- Αν ο χρήστης δεν έχει δημιουργήσει λογαριασμό στο παρελθόν, πρέπει να δημιουργήσει έναν λογαριασμό. Για τη δημιουργία του απαιτείται να επιλέξει ένα μοναδικό όνομα (username) με το οποίο θα τον αναγνωρίζει η εφαρμογή και το οποίο θα είναι μοναδικό. Επιπλέον πρέπει να επιλέξει έναν κωδικό (password) και να παράσχει μια ηλεκτρονική διεύθυνση (e-mail).
- Από τη στιγμή που ο χρήστης έχει δημιουργήσει έναν λογαριασμό, εισέρχεται στο σύστημα παρέχοντας απλά το όνομα του (username) και τον κωδικό του (password).

#### 3.4.1.2 Αιτήματα φιλίας (Friend Requests) και φιλίες (Friendships)

Σκοπός του συστήματος είναι η παροχή υπηρεσιών στους συνδεδεμένους χρήστες. Η πιο βασική υπηρεσία του συστήματος είναι να παρέχει στους χρήστες του συστήματος εύκολη επικοινωνία. Αυτό σημαίνει ότι από την στιγμή που ένας χρήστης συνδεθεί στον λογαριασμό του θα πρέπει να είναι σε θέση να επικοινωνήσει εύκολα και γρήγορα με όσους χρήστες επιθυμεί. Η επικοινωνία που παρέχει το σύστημα πρέπει να συνδυάζεται και από ασφάλεια των δεδομένων των χρηστών ενώ παράλληλα πρέπει να παρέχει στον χρήστη ελευθερία για την επιλογή των χρηστών με τους οποίους θα μπορεί να επικοινωνήσει. Οι παραπάνω απαιτήσεις έχουν εκφραστεί μέσω της φιλίας. Δύο χρήστες του συστήματος για να επικοινωνήσουν θα πρέπει να γίνουν φίλοι

(friends). Από τη στιγμή που γίνουν φίλοι, οι δύο χρήστες θα μπορούν να ανταλλάσσουν μηνύματα, βίντεο κλήσεις καθώς και ιατρικά δεδομένα.

Η προϋπόθεση δύο χρήστες να είναι φίλοι για να επικοινωνήσουν, δημιουργεί την ανάγκη μιας διαδικασίας μέσω της οποίας δύο χρήστες που δεν είναι φίλοι και θέλουν να επικοινωνούν να μπορούν να δημιουργήσουν μια φιλία. Η παραπάνω δυνατότητα υλοποιείται μέσω των αιτημάτων φιλίας (Friend Requests), όπου ένας χρήστης εκφράζει την επιθυμία του να επικοινωνεί με έναν άλλο χρήστη. Έτσι για παράδειγμα, αν ο χρήστης A επιθυμεί να γίνει φίλος με τον χρήστη B τότε ακολουθείται η παρακάτω διαδικασία:

- Ο χρήστης A στέλνει ένα αίτημα φιλίας στον χρήστη B.
- Ο χρήστης B λαμβάνει το αίτημα φιλίας του A και μπορεί είτε να το αποδεχτεί είτε να το απορρίψει. Εάν το αποδεχτεί τότε οι δύο χρήστες γίνονται φίλοι.
- Όσο οι δύο χρήστες είναι φίλοι, τότε μπορούν εύκολα να επικοινωνούν.

Αν ένας εκ των δύο χρηστών δεν επιθυμεί να επικοινωνεί με κάποιον χρήστη τότε υπάρχει η δυνατότητα να διαγράψει τον φίλο αυτόν. Τότε για να μπορέσουν οι δύο χρήστες να επικοινωνήσουν ξανά θα πρέπει να ξαναδημιουργηθεί φιλία μεταξύ τους. Κάθε φορά που ένας χρήστης συνδέεται στο σύστημα, τότε αυτό είναι σε θέση να εντοπίζει ποιοί είναι οι φίλοι του και στη συνέχεια να τους εμφανίζει στην οθόνη του. Αυτό είναι μια απαραίτητη λειτουργία του συστήματος, ώστε να μπορεί ο χρήστης να επικοινωνεί.

#### **3.4.1.3 Πολλαπλοί Χρήστες Συνδεδεμένοι σε Ίδιο Λογαριασμό**

Το σύστημα δίνει τη δυνατότητα σε έναν χρήστη να εισέρχεται σε έναν λογαριασμό από δύο διαφορετικά μηχανήματα. Έτσι μπορεί να βρίσκεται για παράδειγμα συνδεδεμένος ταυτόχρονα από ένα tablet και από έναν browser στον υπολογιστή. Ο Cloud Server θα πρέπει να είναι σε θέση να καταλαβαίνει ότι πρόκειται για τον ίδιο χρήστη αλλά ταυτόχρονα να μπορεί να διαχωρίζει τα δύο μηχανήματα σε όσες λειτουργίες απαιτείται, παρόλο που θα είναι συνδεδεμένοι στο ίδιο όνομα. Θα λέμε ότι υπάρχουν παραπάνω από μία συνδέσεις σε έναν λογαριασμό.

#### **3.4.1.4 Online Χρήστες**

Για να μπορέσει να επιτευχθεί επικοινωνία μεταξύ δύο χρηστών πρέπει αυτή τη δεδομένη στιγμή οι δύο χρήστες να είναι συνδεδεμένοι στο σύστημα. Αυτό καθιστά απαραίτητο το σύστημα να μπορεί να διαπιστώσει πότε ένας χρήστης είναι συνδεδεμένος. Θα λέμε ότι ένας χρήστης είναι online αν είναι την τρέχουσα στιγμή έχει συνδεθεί στο λογαριασμό του. Ο Cloud Server είναι υπεύθυνος για να διατηρεί την κατάσταση των χρηστών του συστήματος, δηλαδή αν είναι συνδεδεμένοι ή όχι. Κάθε φορά λοιπόν που κάποιος χρήστης συνδέεται στο σύστημα, τότε το τελευταίο θα ενημερώνεται ώστε να τον συμπεριλάβει στους συνδεδεμένους χρήστες.

### 3.4.1.5 Ασύγχρονη Ενημέρωση του Συστήματος

Μία μέθοδος που μειώνει ιδιαίτερα τα μηνύματα μεταξύ εξυπηρετητή και πελάτη είναι η ασύγχρονη επικοινωνία. Έτσι άμα συμβεί κάποιο γεγονός το οποίο αφορά έναν χρήστη ο οποίος είναι συνδεδεμένος, τότε ο Server τον ενημερώνει ασύγχρονα και η εφαρμογή που τρέχει στον λογαριασμό του πελάτη λαμβάνει την εκάστοτε απόφαση. Οι ενημερώσεις του συστήματος που πραγματοποιούνται στο σύστημα είναι οι εξής: αποδοχή ή απόρριψη αιτήματος φιλίας, αποστολή/ακύρωση αιτήματος, αποστολή/διαγραφή μηνύματος κειμένου, διαγραφή φίλου και αλλαγή κατάστασης κάποιου χρήστη.

Ενδεικτικά έχουμε:

- **Αποδοχή/Απόρριψη αιτήματος:** Ο Server ελέγχει αν το μήνυμα έχει ακυρωθεί από τον αποστολέα ή εαν έχει συμβεί κάποιο άλλο λάθος και στη συνέχεια στέλνει σε όλες τις συνδέσεις του αποστολέα μήνυμα ώστε να ενημερωθούν.
- **Ακύρωση αιτήματος:** Ο Server ελέγχει αν ο παραλήπτης έχει αποδεχθεί το αίτημα ή αν έχει συμβεί κάποιο άλλο λάθος και στη συνέχεια στέλνει κατάλληλο μήνυμα σε όλους τους παραλήπτες ώστε να ενημερωθούν.
- **Αλλαγή κατάστασης:** Αν κάποιος χρήστης εξέλθει από το σύστημα ή εισέλθει και δεν υπάρχει άλλη σύνδεση στον ίδιο λογαριασμό, τότε ενημερώνει όλες τις συνδέσεις των φίλων του να αλλάξουν την κατάσταση σε *offline/online* αντίστοιχα.

### 3.4.1.6 Μηνύματα Κειμένου (Text Messages)

Ένας από τους τρόπους επικοινωνίας που παρέχεται στους χρήστες είναι μέσω γραπτών μηνυμάτων. Προϋπόθεση φυσικά για την ανταλλαγή μηνυμάτων κειμένου μεταξύ δύο χρηστών είναι να είναι φίλοι. Τα μηνύματα που ανταλλάσσονται αποθηκεύονται από τον Server για οποιοδήποτε ζεύγος φίλων του συστήματος υπάρχει, και έτσι κάθε φορά που ένας χρήστης εισέρχεται στο σύστημα, μπορεί να δει όλα τα μηνύματα που έχει ανταλλάξει με όλους τους φίλους του. Επιπλέον και οι δύο χρήστες μιας συνομιλίας έχουν τη δυνατότητα να διαγράψουν κάποιο μεμονωμένο μήνυμα καθώς και ολόκληρη την συνομιλία. Η αλλαγή αυτή επηρεάζει και τους δύο χρήστες, δηλαδή αν ένα μήνυμα διαγραφεί από την συνομιλία διαγράφεται και για τους δύο χρήστες.

### 3.4.1.7 Σηματοδότηση (Signaling) και WebRTC

Ο Cloud Server είναι υπεύθυνος για τη σηματοδότηση (signaling) του συστήματος η οποία είναι απαραίτητη για την υλοποίηση του WebRTC. Ο Server με τις πληροφορίες που έχει αποθηκεύσει είναι υπεύθυνος για την μεταφορά των μηνυμάτων που είναι απαραίτητα για να μπορέσουν δύο χρήστες να πραγματοποιήσουν βίντεο κλήση. Επιπλέον, το σύστημα είναι σε θέση να ελέγξει αν ένας χρήστης είναι συνδεδεμένος, αν βρίσκεται στην μέση μιας κλήσης ή αν υπάρχουν πολλαπλές συνδέσεις σε έναν λογαριασμό και να στείλει τα ανάλογα μηνύματα σε κάθε μία από τις περιπτώσεις. Επιπλέον, είναι υπεύθυνος να ενημερώσει τους χρήστες για τυχόν αλλαγές ή σφάλματα που έχουν προκύψει. Στην συνέχεια παρουσιάζονται τα βασικότερα Use Cases που έχουν διαμορφωθεί στην συγκεκριμένη υλοποίηση [32].



- Ο χρήστης A καλεί τον χρήστη B

Μετασυνθήκες:

- Οι χρήστες A και B βρίσκονται σε κλήση είτε
- Ο χρήστης έχει ενημερωθεί ότι η κλήση με τον χρήστη B δεν μπορεί να πραγματοποιηθεί.

Προϋποθέσεις:

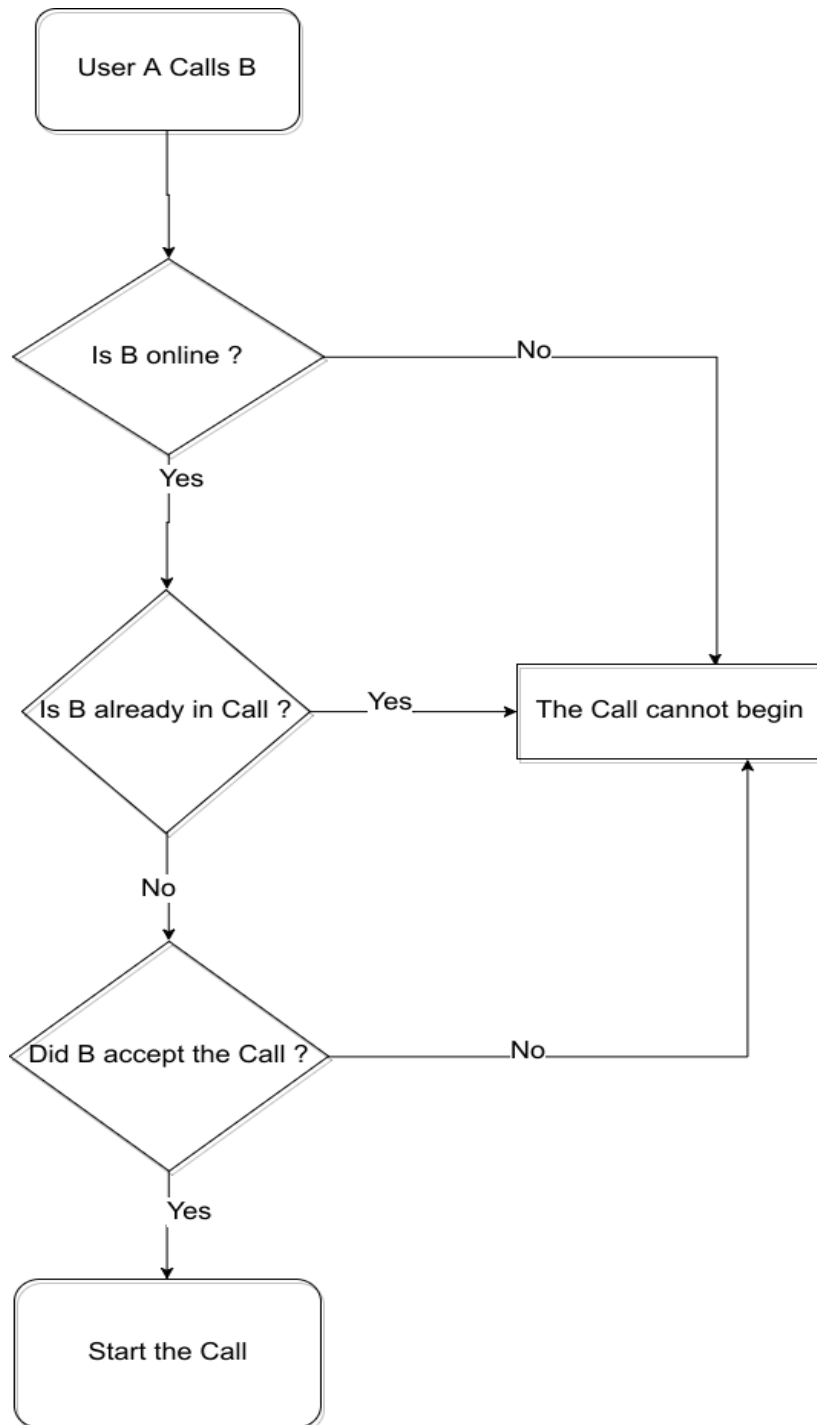
- Τόσο ο χρήστης A όσο και ο χρήστης B είναι συνδεδεμένοι στο σύστημα.
- Οι χρήστες A,B (όλες οι τρέχουσες συνδέσεις) δεν βρίσκονται την τρέχουσα στιγμή σε κλήση.

Ενεργοποίηση της υπηρεσίας:

- Ο χρήστης A επιλέγει το εικονίδιο του χρήστη B και επιλέγει κλήση.

Βασική Ροή :

- Το σύστημα παρέχει έναν εικονίδιο για τους φίλους του χρήστη A. Ο χρήστης B πρέπει να είναι ένα εξ' αυτών.
- Ο χρήστης A αν θέλει να καλέσει τον χρήστη B επιλέγει το εικονίδιο και πατάει το κουμπί κλήσης.
- Το σύστημα λαμβάνει την ενέργεια αυτή και ενημερώνει τον χρήστη B για την εισερχόμενη κλήση.
- Αν ο χρήστης B αποδεχτεί την κλήση, το σύστημα πραγματοποιεί τις απαραίτητες ενέργειες ώστε να ξεκινήσει η επικοινωνία μέσω βίντεο κλήσης, διαφορετικά
- Ενημερώνει τον χρήστη A ότι ο χρήστης B δεν είναι διαθέσιμος για βίντεο κλήση.



Σχήμα 25: Ροή ελέγχου για πραγματοποίηση κλήσης

- Ο χρήστης A είναι σε κλήση με τον χρήστη B και η κλήση περατώνεται.

Μετασυνθήκες:

- Η κλήση περατώνεται.

Προϋποθέσεις:

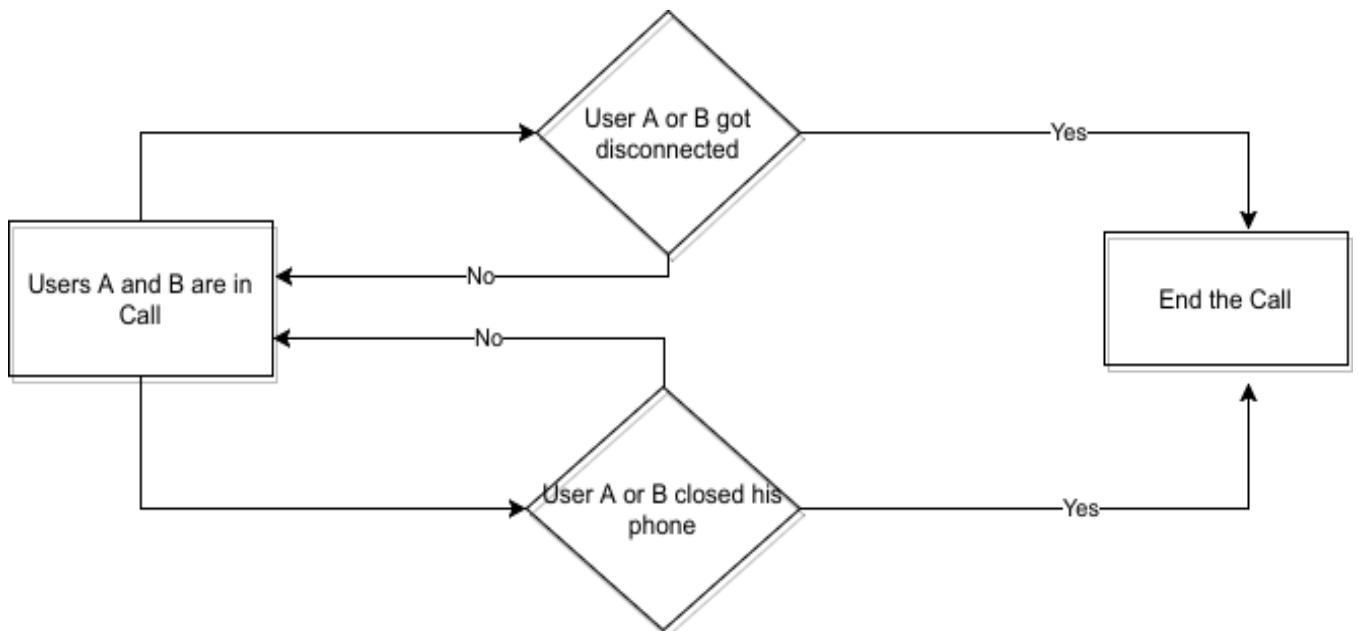
- Τόσο ο χρήστης A όσο και ο χρήστης B είναι συνδεδεμένοι στο σύστημα και βρίσκονται σε κλήση.

Ενεργοποίηση της υπηρεσίας:

- Ένας εκ των χρηστών A,B τερματίζει την κλήση ή αποσυνδέεται.

Βασική Ροή :

- Οι χρήστες βρίσκονται σε κλήση.
- Ένας από τους χρήστες A, B επιλέγει το κουμπί 'τερματισμός' που παρέχει το σύστημα ή αποσυνδέεται από το σύστημα.
- Το σύστημα ενημερώνει τον συνομιλητή του χρήστη, ότι η κλήση έχει πλέον τερματιστεί, και κλείνει το βίντεο,



Σχήμα 26: Ροή ελέγχου τερματισμού κλήσης.

### 3.4.1.8 Μεταφορά (Αποστολή) Βιοσημάτων

Το τελευταίο μέρος της εφαρμογής που αφορά την κεντρική υποδομή του συστήματος είναι οι υπηρεσίες που παρέχει το σύστημα σχετικά με τα βιοσήματα. Όπως αναφέρεται στα παρακάτω, τα βιοσήματα είναι οι μετρήσεις που λαμβάνει ο χρήστης - ασθενής μέσω του οξυμέτρου που διαθέτει. Τα βιοσήματα που μπορούν να στείλουν οι χρήστες – ασθενείς είναι η οξυγόνωση του αίματος και οι καρδιακοί παλμοί. Το σύστημα επιτρέπει στους απλούς χρήστες να ζητήσουν από τους χρήστες ασθενείς του συστήματος τα αποθηκευμένα βιοσήματα που έχουν λάβει στον παρελθόν ή να δουν σε πραγματικό χρόνο τις μετρήσεις που λαμβάνει ο ασθενής μια δεδομένη στιγμή, αν οι δύο χρήστες βρίσκονται σε βίντεο – κλήση.

Τα Use Cases της συγκεκριμένης υπηρεσίας είναι:

- Ο χρήστης A ζητά τα αποθηκευμένα βιοσήματα του χρήστη – ασθενή B

Μετασυνθήκες:

- Ο χρήστης A μπορεί να δει τα αποθηκευμένα βιοσήματα.
- Ο χρήστης A δεν μπορεί να δει τα αποθηκευμένα βιοσήματα

Προϋποθέσεις:

- Τόσο ο χρήστης A πρέπει να είναι συνδεδεμένοι στο σύστημα, ενώ ο χρήστης B πρέπει να επιτρέψει στον χρήστη A να δει τις μετρήσεις του. Επιπλέον οι δύο χρήστες πρέπει να είναι φίλοι.

Ενεργοποίηση της υπηρεσίας:

- Ο χρήστης A πατά το κουμπί για την αποστολή αίτησης να δει το ιστορικό των βιοσημάτων του χρήστη B.

Βασική ροή :

- Ο χρήστης A πατά το κουμπί με το οποίο ζητά τα δεδομένα του χρήστη B
- Ο χρήστης B ειδοποιείται για το αίτημα του χρήστη A και δέχεται ή όχι να στείλει τα δεδομένα στον A.
- Αν δεχτεί στην οθόνη του χρήστη A εμφανίζονται τα βιοσήματα του B σε μορφή διαγράμματος, διαφορετικά στον χρήστη A δεν επιτρέπεται να δει τα βιοσήματα του B.

- Οι χρήστες A,B είναι σε τηλεφωνική κλήση και ο A αρχίζει να λαμβάνει μετρήσεις μέσω του οξυμέτρου.

Μετασυνθήκες:

- Οι χρήστες A,B μπορούν να δουν τις μετρήσεις του χρήστη A όπως αυτός τις λαμβάνει από το οξύμετρο.

Προϋποθέσεις:

- Ο χρήστης A είναι χρήστης – ασθενής.
- Οι χρήστες A,B βρίσκονται σε κλήση.

Ενεργοποίηση της υπηρεσίας:

- Ο χρήστης A ξεκινά να χρησιμοποιεί το οξύμετρο και λαμβάνει μετρήσεις.

Βασική ροή :

- Ο χρήστης A αρχίζει να λαμβάνει μετρήσεις μέσω του οξυμέτρου.
- Οι δύο χρήστες ξεκινάνε να βλέπουν τις μετρήσεις του χρήστη A.

### 3.4.2 Υποδομή του Raspberry

Στην προηγούμενη παράγραφο εξηγήσαμε τις κεντρικές υπηρεσίες που παρέχει το σύστημα στους χρήστες του. Οι υπηρεσίες αναφέρονται σε όλους τους χρήστες, δηλαδή τόσο στους απλούς χρήστες του συστήματος όσο και στους ασθενείς. Το σύστημα παρέχει όμως κάποιες επιπλέον υπηρεσίες όσο αφορά τους χρήστες ασθενείς και οι οποίες έχουν να κάνουν με υπηρεσίες υπεύθυνες για την παρακολούθηση της υγείας των ασθενών. Για τον σκοπό αυτό οι χρήστες ασθενείς εξυπηρετούνται από έναν δεύτερο τοπικό Server, ο οποίος εξυπηρετεί τοπικά τους ασθενείς ως μια SPA μέσω του browser. Η φύση της εφαρμογής μπορεί να χωριστεί σε δύο μέρη. Το πρώτο είναι οι υπηρεσίες οι οποίες απαιτούν σύνδεση στο διαδίκτυο και έχουν να κάνουν με το κοινωνικό δίκτυο που παρέχεται από τον κεντρικό Εξυπηρετητή. Το δεύτερο μέρος πρόκειται για τις υπηρεσίες που δεν απαιτούν σύνδεση στο διαδίκτυο και έχουν να κάνουν με την παρακολούθηση του ασθενή. Θα αναφερόμαστε στο πρώτο μέρος ως Online λειτουργίες και στο δεύτερο ως Offline. Για να μπορέσει ο χρήστης ασθενής να συνδεθεί στο κοινωνικό δίκτυο που παρέχει το σύστημα θα πρέπει να συνεργαστούν και οι δύο Server, καθώς πλέον τώρα ο ασθενής έχει επιπρόσθετες λειτουργίες για τις οποίες είναι υπεύθυνος ο τοπικός Server που τον εξυπηρετεί. Προφανώς η ιδιότητά του ως ασθενή έχει δημιουργήσει κάποιες διαφορές σε σχέση με τον απλό χρήστη και για τις οποίες συνεργάζονται και οι δύο Server.

Συγκεκριμένα οι δύο κύριες διαφορές τους είναι:

- Ο ασθενής μπορεί να εισέλθει στο σύστημα μόνο μέσω του Raspberry και όχι από τη δημόσια διεύθυνση ενώ ο απλός χρήστης δε μπορεί να εισέλθει από την εφαρμογή που τρέχει στο Raspberry.
- Ο ασθενής είναι σε θέση να προωθεί τις μετρήσεις του σε κανονικούς χρήστες, αλλά δεν είναι σε θέση να λαμβάνει μετρήσεις από άλλους ασθενείς.

Όλες οι άλλες υπηρεσίες του συστήματος όσον αφορά το κοινωνικό δίκτυο και τις λειτουργίες του είναι κοινές για τους δύο χρήστες και είναι στην αρμοδιότητα αποκλειστικά του κεντρικού εξυπηρετητή (Cloud Server). Παρακάτω αναλύονται οι κυριότερες λειτουργίες που παρέχονται στους ασθενείς – χρήστες.

#### 3.4.2.1 Σύνδεση στο Δίκτυο

Όπως και οι κανονικοί χρήστες του συστήματος ο ασθενής θα πρέπει να εγγραφεί στο σύστημα. Η εγγραφή πραγματοποιείται με τον ίδιο τρόπο με τους απλούς χρήστες. Σε αντίθεση όμως με τους απλούς χρήστες όταν η εγγραφή ολοκληρωθεί με επιτυχία τότε ο χρήστης – ασθενής δεν χρειάζεται να εισάγει κάθε φορά το όνομα και τον κωδικό του αλλά θα γίνεται αυτόματα από την εφαρμογή. Έτσι όταν εκκινεί την εφαρμογή του αυτή θα βρίσκει τα στοιχεία του και θα τον συνδέει στο σύστημα όταν αυτός επιθυμεί.

### 3.4.2.2 Υπενθυμίσεις

Οι ασθενείς του συστήματος μπορούν να θέτουν υπενθυμίσεις. Μία υπενθύμιση είναι κάποιο κείμενο το οποίο οι ασθενείς ορίζουν να εμφανιστεί μετά από κάποια χρονική περίοδο την οποία αυτοί έχουν επιλέξει, για παράδειγμα κάποιο φάρμακο που πρέπει να ληφθεί ή κάποια συνάντηση με έναν γιατρό. Οι ασθενείς μπορούν να ορίζουν το κείμενο της υπενθύμισης, καθώς και την μέρα /ώρα που επιθυμούν αυτή να εμφανιστεί. Μια επιπλέον ρύθμιση που μπορούν να ελέγξουν στον ορισμό των υπενθυμίσεων είναι η επανάληψη ανά τακτά χρονικά διαστήματα. Οι δυνατότητες που δίνονται είναι να μην υπάρχει επανάληψη, να επαναλαμβάνεται η υπενθύμιση ημερησίως ή τέλος εβδομαδιαίως.

Οι υπενθυμίσεις που έχουν οριστεί στον χρήστη θα είναι διαθέσιμες σε έναν πίνακα από τον οποίο θα μπορεί τόσο να τις παρακολουθεί όσο και να τις τροποποιεί. Η τροποποίηση θα αναφέρεται σε όλες τις παραμέτρους μιας υπενθύμισης δηλαδή στο κείμενο, στην ώρα/ημέρα καθώς και στην περιοδικότητά της. Επιπλέον ο χρήστης θα μπορεί να διαγράψει μια υπενθύμιση από τον πίνακα με τις υπενθυμίσεις. Ο χρήστης δεν θα είναι σε θέση να δει υπενθυμίσεις οι οποίες αναφέρονται σε προηγούμενες χρονικές στιγμές εκτός αν αυτές έχουν οριστεί να εμφανίζονται κατ' επανάληψη. Όταν η υπενθύμιση εμφανιστεί στον χρήστη αυτός μπορεί να την αποκρύψει από την οθόνη ή μπορεί να θέσει η επανάληψη να επαναληφθεί σε δέκα λεπτά.

### 3.4.2.3 Βιοσήματα

Τα Βιοσήματα είναι η βασική παράμετρος μέσω της οποίας ο ασθενής, οι γιατροί καθώς και το σύστημα θα μπορεί να παρακολουθεί την κατάσταση της υγείας του. Στην συγκεκριμένη εφαρμογή τα βιοσήματα θα αντιπροσωπεύουν τους καρδιακούς παλμούς του ασθενή και την οξυγόνωση του αίματος. Οι τιμές αυτές στην συνέχεια θα αποθηκεύονται στο σύστημα. Οι τρόποι εισαγωγής και αποθήκευσης και εισαγωγής και αποθήκευσης θα είναι οι εξής δύο :

- Χειροκίνητα.
- Μέσω του οξυμέτρου που θα διαθέτουν οι ασθενείς.

Εκτός από την λήψη και αποθήκευσης των βιοσημάτων οι ασθενείς θα είναι σε θέση να προβάλλουν τα βιοσήματα, να τα τροποποιούν, να τα διαγράφουν αλλά και να τα στέλνουν σε άλλους χρήστες του συστήματος.

#### 3.4.2.3.1 Εισαγωγή βιοσημάτων στο Σύστημα

Όπως αναφέραμε ο ασθενής θα μπορεί να αποθηκεύσει ένα βιοσήμα στο σύστημα με δύο τρόπους χειροκίνητα και αυτόματα μέσω του οξυμέτρου.

#### - Χειροκίνητη Εισαγωγή

Ο χρήστης μπορεί να εισάγει ένα βιοσήμα χειροκίνητα. Για να ολοκληρωθεί η παραπάνω διαδικασία πρέπει να διαλέξει μία από τις δύο κατηγορίες οι οποίες είναι διαθέσιμες. Το σύστημα αυτόματα θα επιλέξει ως ημερομηνία και ώρα εισαγωγής στο σύστημα την δεδομένη στιγμή που ο χρήστης ολοκλήρωσε τη διαδικασία. Αν ο χρήστης επιθυμεί να ορίσει μια διαφορετική ώρα τότε πρέπει να επιλέξει το κουμπί 'Επιλογή ώρα και μέρα' και να τα εισάγει χειροκίνητα.

#### - Εισαγωγή με Χρήση Οξυμέτρου

Ο δεύτερος τρόπος που μπορεί ο χρήστης να αποθηκεύσει βιοσήματα είναι μέσω του οξυμέτρου. Αρχικά ο χρήστης πρέπει να τοποθετήσει το οξύμετρο στο δάκτυλο και ξεκινήσει αυτό να λαμβάνει μετρήσεις. Αυτόματα ο χρήστης θα μπορεί να δει σε πραγματικό χρόνο τις μετρήσεις αυτές σε ένα διάγραμμα. Αφού το σύστημα βεβαιωθεί ότι οι ενδείξεις του οξυμέτρου έχουν σταθεροποιηθεί τότε αρχίζει να αποθηκεύει τα βιοσήματα. Τέλος ενημερώνει τον χρήστη ότι η μέτρηση ολοκληρώθηκε με επιτυχία και ότι μπορεί να αφαιρέσει το οξύμετρο.

### 3.4.2.3.2 Προβολή και επεξεργασία βιοσημάτων

Για την καλύτερη εποπτεία του ιστορικού του ασθενή τα δεδομένα των δύο βιοσημάτων θα εμφανίζονται σε ένα κοινό διαδραστικό διάγραμμα. Εκεί ο χρήστης θα μπορεί να παρακολουθεί το ιστορικό του εύκολα και επιλέγοντας μία μεμονωμένη μέτρηση θα μπορεί να δει πληροφορίες γι' αυτή καθώς και να την διαγράψει ή να τροποποιήσει την τιμή της. Επιπλέον ο χρήστης θα μπορεί να ορίσει το χρονικό εύρος στο οποίο επιθυμεί να παρακολουθεί τα βιοσήματα του. Έτσι θα μπορεί τα αποθηκευμένα βιοσήματα τα οποία έχουν ληφθεί:

- Εντός του τελευταίου εικοσιτετραώρου.
- Εντός των προηγούμενων τριών ημερών.
- Εντός της προηγούμενης εβδομάδας.
- Εντός του προηγούμενου μήνα ή,
- Από την αρχή χρήσης της υπηρεσίας.

### 3.4.2.3.3 Αποστολή Βιοσημάτων

Η τελευταία δυνατότητα σχετικά με τα βιοσήματα είναι η αποστολή του ιστορικού του ασθενή σε εξουσιοδοτημένους χρήστες καθώς και η αποστολή σε πραγματικό χρόνο μετρήσεων που λαμβάνει ο ασθενής μια δεδομένη στιγμή αν αυτός βρίσκεται σε βίντεο – κλήση.



## - Αποστολή Ιστορικού

Σκοπός αυτής της υπηρεσίας είναι ο ασθενής να μπορεί να στείλει τις μετρήσεις του σε άλλους χρήστες του συστήματος με σκοπό να παρακολουθήσουν το ιστορικό του εξ' αποστάσεως. Τα βιοσήματα του ασθενή αποτελούν εμπιστευτικά και ευαίσθητα δεδομένα και έτσι σε θέση να λάβουν είναι μόνο όσοι χρήστες του συστήματος επιθυμεί ο ασθενής. Όταν γίνει αίτηση από έναν χρήστη ο οποίος δεν είναι εξουσιοδοτημένος από τον ασθενή να δει τα βιοσήματά του, το σύστημα ζητά την άδεια του ασθενή να αποστείλει τα βιοσήματα του. Αν ο ασθενής απαντήσει θετικά στην αίτηση, το σύστημα προσθέτει τον αιτούντα στους εξουσιοδοτημένους χρήστες και στο εξής μπορεί να ζητά τις μετρήσεις αυτόματα. Αν ο ασθενής δεν επιτρέψει στον χρήστη να λάβει τις μετρήσεις τότε ο δεύτερος δεν θα τις λάβει. Ο ασθενής είναι επίσης σε θέση να σταματήσει να στέλνει δεδομένα σε κάποιον χρήστη που είχε εξουσιοδοτήσει παλιότερα. Στη συνέχεια ο χρήστης αυτός θεωρείται μη εξουσιοδοτημένος.

## - Αποστολή Μετρήσεων σε Πραγματικό Χρόνο Κατά τη Διάρκεια Βίντεο – Κλήσης

Αν ένας ασθενής βρίσκεται σε κλήση με κάποιον χρήστη του συστήματος τότε θα μπορεί να στείλει τις μετρήσεις του που λαμβάνει από το οξύμετρο σε πραγματικό χρόνο. Για να ξεκινήσει η αποστολή των μετρήσεων αρκεί να τοποθετήσει το οξύμετρο στο χέρι και αυτό θα ξεκινήσει τη λειτουργία του.

### 3.4.2.4 Ρυθμίσεις Λογαριασμού

Ο ασθενής θα μπορεί να τροποποιεί κάποιες ρυθμίσεις της εφαρμογής του. Οι ρυθμίσεις που έχει την δυνατότητα να τροποποιεί είναι οι εξής :

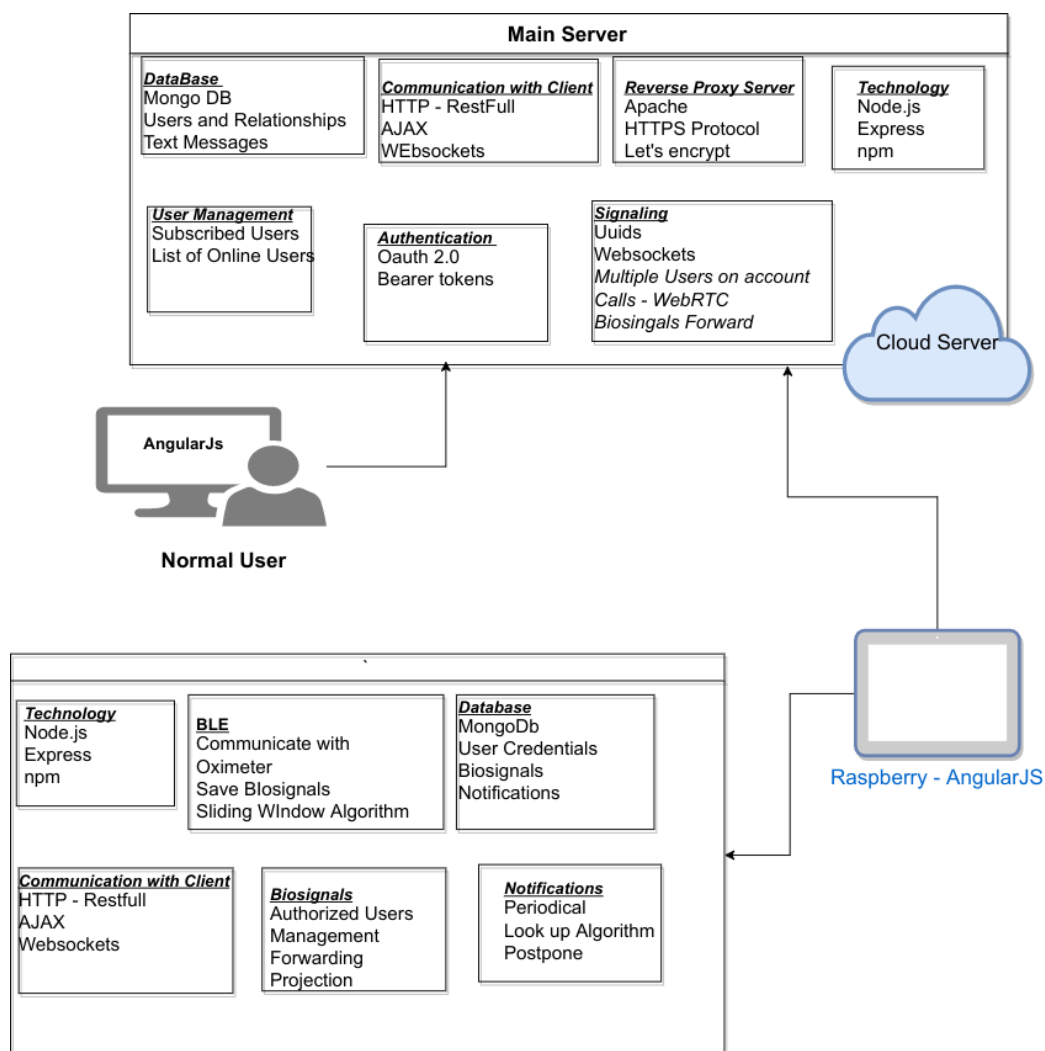
- *Κωδικό Πρόσβασης* : Ο ασθενής είναι σε θέση να αλλάξει τον κωδικό πρόσβασης που είχε εισάγει κατά την εγγραφή στο σύστημα ή που τυχόν άλλαξε στο παρελθόν. Για την ολοκλήρωση της παραπάνω υπηρεσίας αρκεί να εισάγει τον προηγούμενο κωδικό καθώς και τον καινούριο κωδικό που επιθυμεί.
- *Τρόπος σύνδεσης*: Ο ασθενής όπως αναφέραμε δεν χρειάζεται να εισάγει κάθε φορά τα στοιχεία του. Αντιθέτως η εφαρμογή τα βρίσκει και τα στέλνει στον Cloud Server. Οι δυνατότητες που υπάρχουν σχετικά με τη σύνδεση στο δίκτυο είναι η σύνδεση να πραγματοποιείται αυτόματα όταν ο χρήστης εκκινεί την εφαρμογή ή να ορίζει ρητά πότε επιθυμεί να εισέλθει στο σύστημα. Αν επιλέξει το δεύτερο όταν εκκινεί η εφαρμογή θα πρέπει να επιλέξει χειροκίνητα το κουμπί 'Σύνδεση'
- *Εξουσιοδοτημένοι Χρήστες*: Αναφέραμε ότι ο ασθενής μπορεί να επιλέξει ποιούς χρήστες θα βλέπουν τις μετρήσεις του. Μέσω των εξουσιοδοτημένων χρηστών θα μπορεί να παρακολουθεί σε ποιούς χρήστες έχει παράσχει άδεια καθώς και να διαγράψει κάποιον χρήστη στο οποίο έχει παράσχει άδεια στο παρελθόν.

### 3.4.3 Επικοινωνία Μεταξύ των Υποσυστημάτων

Το σύστημα όπως είδαμε αποτελείται από έναν κεντρικό εξυπηρετητή, από μία εφαρμογή SPA που τρέχει σε έναν οποιονδήποτε browser όσον αφορά τους απλούς χρήστες και για κάθε έναν από τους ασθενείς υπάρχει μια εφαρμογή SPA καθώς και ένας υπεύθυνος εξυπηρετητής γι' αυτήν. Έτσι οι απλοί χρήστες επικοινωνούν αποκλειστικά με τον κεντρικό Server ο οποίος είναι υπεύθυνος να στείλει τα κατάλληλα μηνύματα σε άλλους χρήστες. Όσον αφορά την επικοινωνία μεταξύ της εφαρμογής SPA των χρηστών – ασθενών ακολουθείται η παρακάτω σχεδιαστική επιλογή: Για όλες τις offline λειτουργίες συνεργάζονται αποκλειστικά η SPA εφαρμογή και ο τοπικός Server ενώ για την σύνδεση στο δίκτυο η εφαρμογή SPA επικοινωνεί άμεσα τόσο με τον κεντρικό εξυπηρετητή όσο με τον τοπικό εξυπηρετητή. Έτσι οι δύο εξυπηρετητές δεν επικοινωνούν μεταξύ τους άμεσα, μειώνοντας έτσι την χρήση τους στο ελάχιστο δυνατό.

## 4 Υλοποίηση του Συστήματος

Στο παρόν κεφάλαιο αναλύονται και εξετάζονται πρακτικά θέματα καθώς και οι τεχνικές που προτάθηκαν για ορισμένες δυσκολίες. Ειδική μέριμνα έχει δοθεί στον τρόπο που το σύστημα πληροί τις απαιτήσεις όπως αυτές παρουσιάστηκαν στο προηγούμενο κεφάλαιο. Επιπλέον θα γίνει αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν και σε μεθόδους που χρησιμοποιήθηκαν. Τέλος θα παρουσιαστούν μερικά μέρη από κώδικα για καλύτερη κατανόηση των τεχνικών και των στρατηγικών (snippets) καθώς και μία παρουσίαση της διεπαφής τόσο των απλών χρηστών όσο και των ασθενών.



Σχήμα 27: Το Υλοποιημένο Σύστημα

## 4.1 SPA Layout

Η AngularJS για την ενημέρωση και παρακολούθηση των μοντέλων (model) μιας HTML εφαρμογής διαθέτει τους controllers. Συγκεκριμένα περιέχουν μια μεταβλητή, την \$scope, και όποια δεδομένα δοθούν κάτω από αυτή ο controller τα συνδέει με το DOM, τα παρακολουθεί και ενημερώνει την HTML όποτε αυτά αλλάζουν. Έτσι όλο το User Interface (UI) του χρήστη αποτελείται από μεταβλητές οι οποίες βρίσκονται κάτω από την επίβλεψη κάθε φορά ενός controller, ο οποίος τις παρακολουθεί και ενημερώνει ασύγχρονα της HTML σελίδα.

### 4.1.1 SPA Layout των Απλών Χρηστών

Τόσο το User Interface όσο και οι λειτουργίες της εφαρμογής των απλών χρηστών του συστήματος έχουν χωριστεί σε δύο κατηγορίες: την εισαγωγική σελίδα που εμφανίζεται προκειμένου ο χρήστης να εισέλθει στο σύστημα και στην εξουσιοδοτημένη σελίδα η εμφανίζεται αφού ο χρήστης έχει εισέλθει επιτυχώς στο σύστημα. Με αυτόν τον τρόπο είναι πιο εύκολο να χωριστούν τα δύο βασικά μέρη. Η μετάβαση από την εισαγωγική στην εξουσιοδοτημένη σελίδα γίνεται μόνο όταν ο χρήστης εισέλθει επιτυχώς στο σύστημα.

Η εισαγωγική σελίδα χωρίζεται σε δύο λογικά μέρη, ένα για την σύνδεση ήδη εγγεγραμμένου χρήστη στο σύστημα και ένα για τη δημιουργία λογαριασμού ενός χρήστη ώστε να μπορέσει να συνδεθεί στο σύστημα.

Το εξουσιοδοτημένο μέρος, δηλαδή οι υπηρεσίες που παρέχει το σύστημα, χωρίζεται σε πέντε λογικά μέρη τα οποία επιτελούν έναν ξεχωριστή λειτουργία το καθένα. Συγκεκριμένα έχουμε :

- **Μηνύματα κειμένου:** Πρόκειται για το μέρος που είναι υπεύθυνο για την αποθήκευση, συγγραφή, αποστολή, παρουσίαση των μηνυμάτων και την επικοινωνία με τον Server.
- **Βίντεο κλήσεις:** Πρόκειται για το μέρος της εφαρμογής που είναι υπεύθυνο για την ομαλή έκβαση των βίντεο κλήσεων και το οποίο ελέγχει αν π.χ ένας χρήστης είναι σε κλήση, με ποιόν μιλάει, την κατάσταση του συνομιλητή κ.α.
- **Βιοσήματα:** Είναι το μέρος της εφαρμογής που ελέγχει τα αιτήματα για αποστολή βιοσημάτων στο σύστημα, καθώς και την αποθήκευση και παρουσίαση στον χρήστη.
- **Φίλοι:** Είναι το μέρος που είναι υπεύθυνο για την αποθήκευση των φίλων ενός χρήστη, την απεικόνιση τους στην σελίδα καθώς και την ενημέρωση της κατάστασής τους, δηλαδή αν βρίσκονται ή όχι συνδεδεμένοι στο σύστημα.
- **Αναζήτηση φίλων και αιτήματα φιλίας:** Είναι το μέρος της εφαρμογής το οποίο προτείνει στον χρήστη πιθανούς χρήστες του συστήματος που θέλει να προσθέσει ως φίλους, ενώ είναι υπεύθυνο και για την αποθήκευση, προώθηση και απεικόνιση της κατάστασης των αιτημάτων φιλίας του χρήστη.

## 4.1.2 SPA Layout των Χρηστών Ασθενών

Ομοίως με τους απλούς χρήστης το SPA Layout τόσο το UI όσο και η διαμέριση του κώδικα χωρίζεται σε διαφορετικά μέρη τα οποία συνεργάζονται για να παρέχουν στον χρήστη – ασθενή όλες τις υπηρεσίες. Τα τέσσερα αυτά μέρη αυτά είναι :

- **Online Services:** Πρόκειται για το μέρος της εφαρμογής το οποίο είναι υπεύθυνο για την σύνδεση (ή εγγραφή αντίστοιχα) του ασθενή στο σύστημα.
- **Υπενθυμίσεις:** Πρόκειται για το μέρος της εφαρμογής που ελέγχει, τροποποιεί, διαγράφει και ενημερώνει τον ασθενή τις ειδοποιήσεις του.
- **Βιοσήματα:** Πρόκειται για το μέρος της εφαρμογής που αποθηκεύει τα βιοσήματα του χρήστη, τον ενημερώνει για αυτά, τα προβάλλει σε άξονα και τα τροποποιεί.
- **Ρυθμίσεις:** Πρόκειται για το μέρος της εφαρμογής το οποίο είναι υπεύθυνο για την αλλαγή των ρυθμίσεων στο προφίλ του χρήστη.

## 4.2 Ο Κεντρικός Εξυπηρετητής (Cloud Server)

Ο εξυπηρετητής έχει υλοποιηθεί σε λειτουργικό σύστημα Linux Ubuntu 16.04.4 LTS και φιλοξενείται στον “Ωκεανό. Το framework που χρησιμοποιήθηκε είναι το Nodejs σε συνδυασμό με το module express. Η δημόσια διεύθυνση από την οποία είναι διαθέσιμος είναι η <https://MyApplication.com/node>. Η βάση δεδομένων, με την οποία επικοινωνεί ο Server για την αποθήκευση των δεδομένων, έχει υλοποιηθεί με mongo database. Τα δεδομένα που θα αποθηκεύονται θα είναι πληροφορίες για τους χρήστες του συστήματος όπως :

- Το όνομα και ο κωδικός πρόσβασης που έχουν επιλέξει.
- Οι φίλοι τους, τα αιτήματα φιλίας που έχουν στείλει και λάβει.
- Τα μηνύματα που έχουν ανταλλάξει με άλλους χρήστες καθώς και
- Τα σύμβολα πιστοποίησης που έχουν δημιουργηθεί, απαραίτητα για την πρόσβαση των πόρων του εξυπηρετητή.

Το πρωτόκολλο https που χρειάζεται ώστε να λειτουργούν οι βίντεο κλήσεις καθώς και για να παρέχεται ασφάλεια και εμπιστευτικότητα υλοποιήθηκε μέσω ενός reverse proxy server υλοποιημένου με apache Server, ενώ η απόκτηση του πιστοποιητικού έγινε μέσω της εφαρμογής Certbot.

### 4.2.1 Τρόπος επικοινωνίας Μεταξύ SPA και των Servers

Η επικοινωνία μεταξύ χρήστη-εξυπηρετητή ακολουθεί την εξής λογική: Οι λειτουργίες που αναφέρονται σε λειτουργίες οι οποίες προϋποθέτουν ο χρήστης να είναι συνδεδεμένος στο σύστημα (π.χ. κλήση φίλου, αποστολή μηνύματος, ποιού φίλοι ενός χρήστη είναι συνδεδεμένοι, αποστολή βιοσημάτων κλπ) καθώς και σε λειτουργίες

υπεύθυνες για ασύγχρονη ανανέωση των SPA των χρηστών, γίνονται μέσω των Websockets. Συγκεκριμένα, κάθε χρήστης θα εγκαθιστά μία σύνδεση με Websockets με τον κεντρικό εξυπηρετητή, από την οποία θα επικοινωνεί μαζί του μέχρι να αποσυνδεθεί από το σύστημα. Οι υπόλοιπες λειτουργίες θα γίνονται μέσω απλών AJAX αιτημάτων προς τον Server.

#### 4.2.1.1 AJAX – HTTP Αιτήματα

Εφόσον ο χρήστης – ασθενής χρησιμοποιεί 2 διαφορετικά URL, σημαίνει ότι (τουλάχιστον) ένα από αυτά είναι σε διαφορετικό domain από το δικό του. Έτσι, στον κεντρικό Server ενεργοποιήθηκε η δυνατότητα CORS ώστε να μπορεί να εξυπηρετεί και τον ασθενή-χρήστη.

```
var app = express();
app.use(function(req, res, next) {
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Methods",
    "GET, HEAD, OPTIONS, POST, PUT, DELETE");
  res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With,
    Content-Type, Accept, Authorization");
  next();
});
```

Οι πόροι του Server είναι προστατευμένοι και για να επιτραπεί στον χρήστη να εισέλθει στο σύστημα πρέπει να ακολουθηθεί ο τύπος πιστοποίησης Resource Owner Password Credential, όπως αυτός έχει οριστεί από το πρωτόκολλο OAuth 2.0. Έτσι οι χρήστες μπορούν να αποκτήσουν πρόσβαση στο σύστημα μόνο αποκτώντας ένα access token. Η υλοποίηση της παραπάνω διαδικασίας έγινε μέσω των modules oauth2orize, passport, passport-oauth2-client-password και passport-http-bearer του npm.

Οι αντίστοιχοι κώδικες είναι:

```
✓ passport.use(new
  BearerStrategy('accessToken', function(access_token, done) {
    token.findOne({value:access_token}, function(err, token_found){
      if(err)
        return done(err);
      else if(!token_found)
        return done(null, false);
      else if(token_found.expirationDate <= new Date() )
        return(done(null, false));
      else {
        users.findOne({_id: token_found.userId}, function (err,
found_user) {
          if (err)
            return done(err);
          if (!found_user)
```

```

        return done(null, false);
      done(null, found_user);
    })
  }
});

```

- ✓ `passport.use(new ClientPasswordStrategy(function(client_id, client_secret, done) {
 client.findOne({ClientId: client_id}, function (err, client) {
 if (err) {return done(err);
 } if (!client) {
 console.log('No client with username : ' + client_id);
 return done(null, false, {message: 'No user with this name'});
 }
 if (client_secret !== client.Client_Secret) {
 return done(null, false, {message: 'incorrect password'});
 }
 return done(null, client);
 })
 }));`
- ✓ `server.exchange(oauth2orize.exchange.password(function (client, username, password, category, done) {
 users.findOne({username: username}, function (err, user_found) {
 if (err) {
 return done(err);
 }
 if (!user_found) {
 //User not found
 return done(null, false);
 }
 user_found.verifyPassword(password, function (err, isMatch) {
 if (err)
 return done(null, false);
 else if (!isMatch)
 return done(null, false);
 //confirmed password now confirm category
 //Create and save new token
 var token_value = utils.uid(256);
 var token = new Token({
 value: token_value,
 clientId: client.ClientId,
 userId: user_found._id,
 expirationDate: new Date(new Date().getTime() + (24 * 3600 * 1000)) // Each token is Valid for one Day
 });
 token.save(function (err) {
 if (err) return done(err);
 return done(null, token_value);
 });
 });
 });`

```

    });
  });
});

```

Για τον καλύτερο διαχωρισμό των λειτουργιών ορίστηκαν 5 βασικά URLs τα οποία έχουν να κάνουν με διαφορετικές λειτουργίες:

- <https://MyApplication.com/node/login>: Η συγκεκριμένη διεύθυνση έχει να κάνει με τους πιστοποιημένους χρήστες και είναι υπεύθυνη για τα αιτήματα φιλίας και την ενημέρωση της κατάστασης της φιλίας μεταξύ των χρηστών. Όλοι οι πόροι κάτω από το συγκεκριμένο URL είναι προστατευμένοι. Στο συγκεκριμένο URL θα γίνονται τόσο απλά αιτήματα HTTP όσο και AJAX. Τα δεδομένα που ανταλλάσσονται μεταξύ αυτής της URL και της εφαρμογής είναι σε μορφή JSON objects.

Ενδεικτικά παραθέτουμε το URL που επιστρέφει τους φίλους ενός εγγεγραμμένου χρήστη:

```

router.get('/GetFriends', passport.authenticate('bearer', {session:
false}), function (req, res, next) {
  relationship.findOne({user: req.user.username}, function (err, result)
  {
    if (err)
      res.send({'message': err});
    else {
      res.send(result.friends);
    }
  });
});

```

- <https://MyApplication.com/node/subscribe>: Όπως υποδηλώνει και το όνομά της, η συγκεκριμένη URL είναι υπεύθυνη για την εγγραφή χρηστών στο σύστημα. Στο συγκεκριμένο URL θα γίνονται τόσο απλά αιτήματα HTTP όσο και AJAX. Τα δεδομένα που ανταλλάσσονται μεταξύ αυτή της URL και της εφαρμογής είναι σε μορφή JSON objects.

```

router.post('/', function (req, res, next) {
  console.log('Received Subscribe Application ');
  var newUser = new user(
    {
      username: req.body.username,
      password: req.body.password, email: req.body.email,
      category: req.body.category
    });
  console.log(newUser);
  var Relationship = new relationship({
    user: req.body.username
  });
  newUser.save(function (err) {
    if (err) {
      res.send(err);
    }
    else
    {

```



```

        Relationship.save(function (err, result) {
            if (err) {
                res.send(err);
                return;
            }
            next();
        })
    }
});
}, getToken);

```

- <https://MyApplication.com/node/messages>: Πρόκειται για το URL που είναι υπεύθυνο για την απόκτηση των συνομιλιών ενός χρήστη, την αποθήκευση και προώθηση μηνυμάτων καθώς και τη διαγραφή κάποιου συγκεκριμένου μηνύματος.

Παρατίθεται ο κώδικας της διαγραφής μηνύματος:

```

router.delete('/', passport.authenticate('bearer', {session:
false}), function(req, res, next){ // We have to delete message from database
    var uuid = req.query.uuid;
    var Userid = req.query.uuidUser;
    var Array=[req.query.target, req.user.username];
    conversation.findOne({Participants:{$all:Array}}, function(err, result){
        if(err)
            res.send({message:err});
        else{
            var ConvId = result.ConversationId;
            message.update({ConvesrationId:ConvId},{ $pull:{message:
{uniqueId:uuid}}}, function(err, result){
                if(err)
                    res.send({message:err});
                else{
                    console.log(result);
                    res.send({message:'Message was deleted'});
                }
            })
        }
    });
    Wssdelete(req.user.username, Userid, uuid, req.query.target);
});

```

- <https://MyApplication.com/node/biosignals>: Το συγκεκριμένο URL είναι υπεύθυνο για τον έλεγχο του εάν κάποιος χρήστης είναι ασθενής, οπότε και προωθεί το μήνυμα για να αποκτήσει τα βιοσήματα του συγκεκριμένου ασθενή. Αν ο χρήστης-στόχος δεν είναι ασθενής, δεν έχει νόημα η υπηρεσία. Στο συγκεκριμένο URL πραγματοποιούνται αποκλειστικά αιτήματα AJAX, ενώ τα δεδομένα που ανταλλάσσονται είναι σε μορφή JSON objects.
- <https://MyApplication.com/node/autocomple>: URL το οποίο ρωτά τη βάση δεδομένων και επιστρέφει ονόματα εγγεγραμμένων χρηστών τα οποία αντιστοιχίζονται στην αναζήτηση που κάνει ο χρήστης, και τον ενημερώνει σε

πραγματικό χρόνο. Στο συγκεκριμένο URL πραγματοποιούνται αποκλειστικά αιτήματα AJAX, ενώ τα δεδομένα που ανταλλάσσονται είναι σε μορφή JSON objects.

#### 4.2.1.2 Websockets

Τα Websockets αποτελούν τη δεύτερη δίοδο επικοινωνίας μεταξύ εξυπηρετητή και πελάτη. Συγκεκριμένα η αμφίδρομη κατεύθυνση που επιτρέπουν δίνει τη δυνατότητα στον Εξυπηρετητή να λαμβάνει, να επεξεργάζεται και στη συνέχεια να προωθεί μηνύματα πληροφοριών στους χρήστες του συστήματος. Από εδώ και στο εξής θα αναφέρουμε ως WebSocket Server το μέρος του Cloud Server το οποίο είναι υπεύθυνο για την επικοινωνία μέσω των Websockets. Τα Websockets έχουν υλοποιηθεί μέσω του module websocket του npm. Η διεύθυνση που δέχεται τις συνδέσεις Websockets είναι η **wss://MyApplication.com/sockets**.

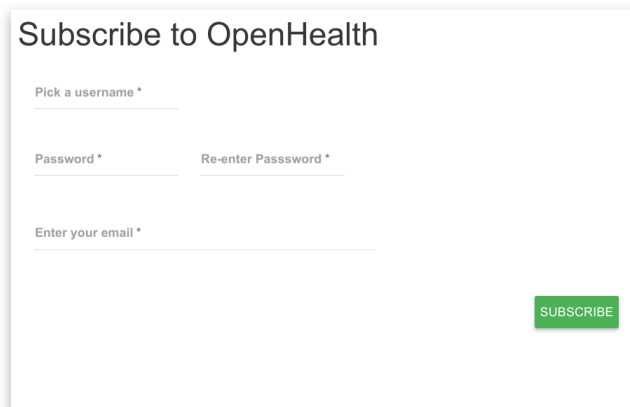
Οι λειτουργίες που επιτελούν τα Websockets είναι :

- Κατάσταση σύνδεσης των φίλων των χρηστών.
- Signaling για τη σύνδεση των απομακρυσμένων χρηστών με σκοπό τη δημιουργία βίντεο κλήσης (WebRTC).
- Προώθηση μηνυμάτων απαραίτητων για τη λειτουργία της εφαρμογής σε πραγματικό χρόνο, όπως μηνυμάτων κειμένου, ασύγχρονης ενημέρωσης των χρηστών π.χ. αν άλλαξε κατάσταση κάποιου αιτήματος φιλίας, φιλίας κλπ.

### 4.3 Υλοποίηση των Απαιτήσεων Απλών Χρηστών του Συστήματος

#### 4.3.1 Χρήστες και Σύνδεση στο Σύστημα

Όπως αναφέραμε, για την επιτυχή σύνδεση του χρήστη απαιτείται η ύπαρξη κάποιου λογαριασμού. Αφού δημιουργηθεί ο λογαριασμός, τότε ο χρήστης απλά συνδέεται με το όνομα και των κωδικό πρόσβασής του. Όταν τα στοιχεία επαληθευτούν από τον Server, τότε στον χρήστη δίνεται ένα bearer token με το οποίο μπορεί πλέον να αιτείται HTTP Requests, καθώς και να εγκαθιδρύει τη σύνδεση με τον WebSocket Server αφού όλα τα resources όπως είπαμε είναι προστατευμένα και το token είναι απαραίτητο. Το token αυτό αποθηκεύει η εφαρμογή του χρήστη καθ'όλη τη διάρκεια που παραμένει συνδεδεμένος στο σύστημα.



Subscribe to OpenHealth

Pick a username \*

Password \* Re-enter Password \*

Enter your email \*

SUBSCRIBE

Σχήμα 28: Η διεπαφή χρήστη (UI) στην εισαγωγική σελίδα

Κάθε εγγεγραμμένος χρήστης αντιπροσωπεύεται στη βάση δεδομένων ως μια εγγραφή της συλλογής Users η οποία περιέχει το όνομα του χρήστη, τον κωδικό πρόσβασης σε κρυπτογραφημένη μορφή, τους φίλους του, καθώς και τα αιτήματα τα οποία έχει στείλει και λάβει. Όταν ένας χρήστης εισέρχεται επιτυχώς στο σύστημα τότε η εφαρμογή πελάτη ζητά από τον server να της αποστείλει πληροφορίες για τους φίλους του, και τα αιτήματα φιλίας. Όπως θα εξηγήσουμε παρακάτω όλες οι πληροφορίες του χρήστη αναπαρίστανται ως Singleton Objects μέσω των Services που παρέχει η AngularJS.

#### 4.3.2 Ασύγχρονη Ενημέρωση του Συστήματος και Angular Services.

Όπως έχουμε αναφέρει υπάρχουν λειτουργίες οι οποίες απαιτούν να υπάρχει ενημέρωση του συστήματος και των εφαρμογών των χρηστών, για παράδειγμα αν κάποιος δεχτεί ένα αίτημα φιλίας τότε οι δύο χρήστες θα πρέπει να γίνουν φίλοι και να ενημερωθούν οι εφαρμογές σε πραγματικό χρόνο. Όπως αναφέραμε οι ενημερώσεις αυτό θα γίνονται ασύγχρονα σε όλο το σύστημα για εξοικονόμησης χρόνου και bandwidth. Ο τρόπος που γίνεται η ασύγχρονη επικοινωνία είναι μέσω των Websockets. Αν ο χρήστης προβεί σε κάποια δράση/αλλαγή η οποία απαιτεί να ενημερωθούν και άλλοι χρήστες τότε η εφαρμογή ενημερώνουν τον Server και στην συνέχεια αυτός στέλνει μηνύματα ενημέρωσης στους χρήστες που το συγκεκριμένο γεγονός αφορά. Οι ενέργειες που αφορούν αυτή την κατηγορία είναι :

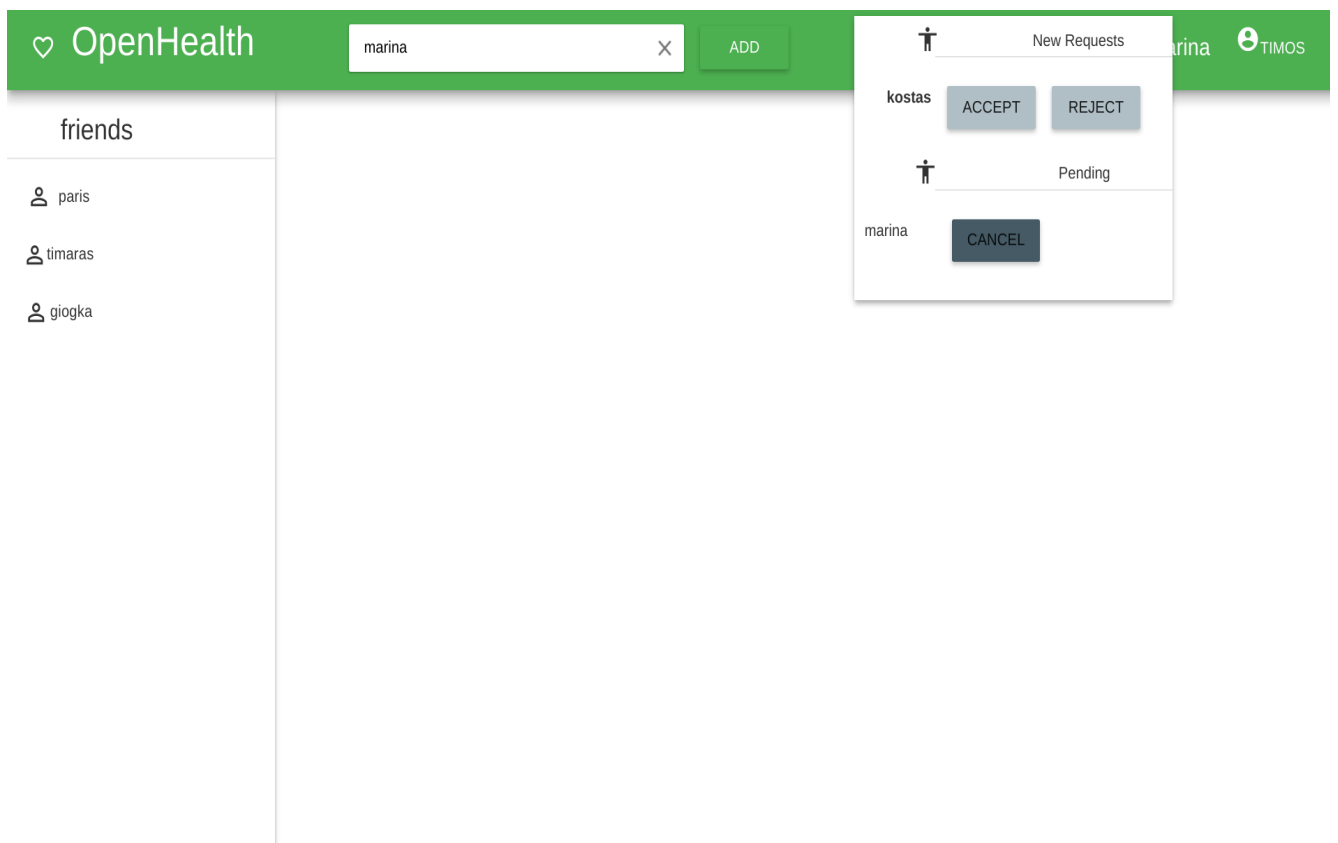
- Διαγραφή φίλου.
- Αποδοχή/ Απόρριψη/ Ακύρωση αιτήματος φιλίας.
- Λήψη/ Αποστολή/ Διαγραφή μηνυμάτων.
- Αποστολή/ Αποδοχή βιοσημάτων.

Για να είναι σε θέση η εφαρμογή του πελάτη να το ενημερώσει και αυτή με την σειρά της ασύγχρονα τα δεδομένα θα πρέπει να έχει έναν μηχανισμό/τρόπο για να τα επιτύχει. Η λογική που ακολουθείται για τις παραπάνω περιπτώσεις είναι η εξής: Αρχικά απαιτείται μία υποδομή η οποία θα αποθηκεύει την τρέχουσα κατάσταση του χρήστη (τους φίλους του, αν είναι συνδεδεμένοι ή όχι, τα μηνύματά του, τα αιτήματα φιλίας κλπ). Η υποδομή αυτή έχει δημιουργηθεί μέσω των Services της AngularJS τα οποία είναι singleton objects και τα οποία είναι υπεύθυνα για δεδομένα που χρειάζονται σε διαφορετικά μέρη της εφαρμογής SPA. Τα παραπάνω περιέχουν όλες τις πληροφορίες που είναι απαραίτητες για τη λειτουργία της εφαρμογής. Κάθε Service λοιπόν αντιπροσωπεύει ένα μέρος της εφαρμογής και έχει τόσο τις πληροφορίες που χρειάζεται (π.χ τα μηνύματα προς κάποιο χρήστη) καθώς και μεθόδους για την αποθήκευση, επεξεργασία και επιστροφή των δεδομένων αυτών. Έτσι όταν η εφαρμογή λάβει ένα μήνυμα από τον Server το οποίο απαιτεί κάποια ενημέρωση, μέσω των κατάλληλων μεθόδων ενημερώνει τα δεδομένα των Services, ενώ μέσω της ασύγχρονης μεθόδου ενημέρωσης όλης της εφαρμογής `$rootScope.$emit` στέλνει ασύγχρονα ένα μήνυμα στον κατάλληλο controller οτι πρέπει να ενημερώσει τις τοπικές μεταβλητές. Αφού ολοκληρωθούν, η εφαρμογή του χρήστη είναι πλέον ενημερωμένη.

### 4.3.3 Αιτήματα φιλίας

Στον εξυπηρετητή τα αιτήματα φιλίας είναι μέρος της συλλογής των χρηστών. Για ευκολότερη λειτουργικότητα και παρακολούθηση τα αιτήματα βρίσκονται σε μία από τις παρακάτω τρεις κατηγορίες :

- Pending – Το αίτημα έχει αποσταλεί και δεν έχει απαντηθεί από τον παραλήπτη.
- Accepted – Το αίτημα έχει γίνει αποδεκτό και οι χρήστες έχουν γίνει φίλοι.
- Cancelled by Sender – Ο αποστολέας ακύρωσε το αίτημα.



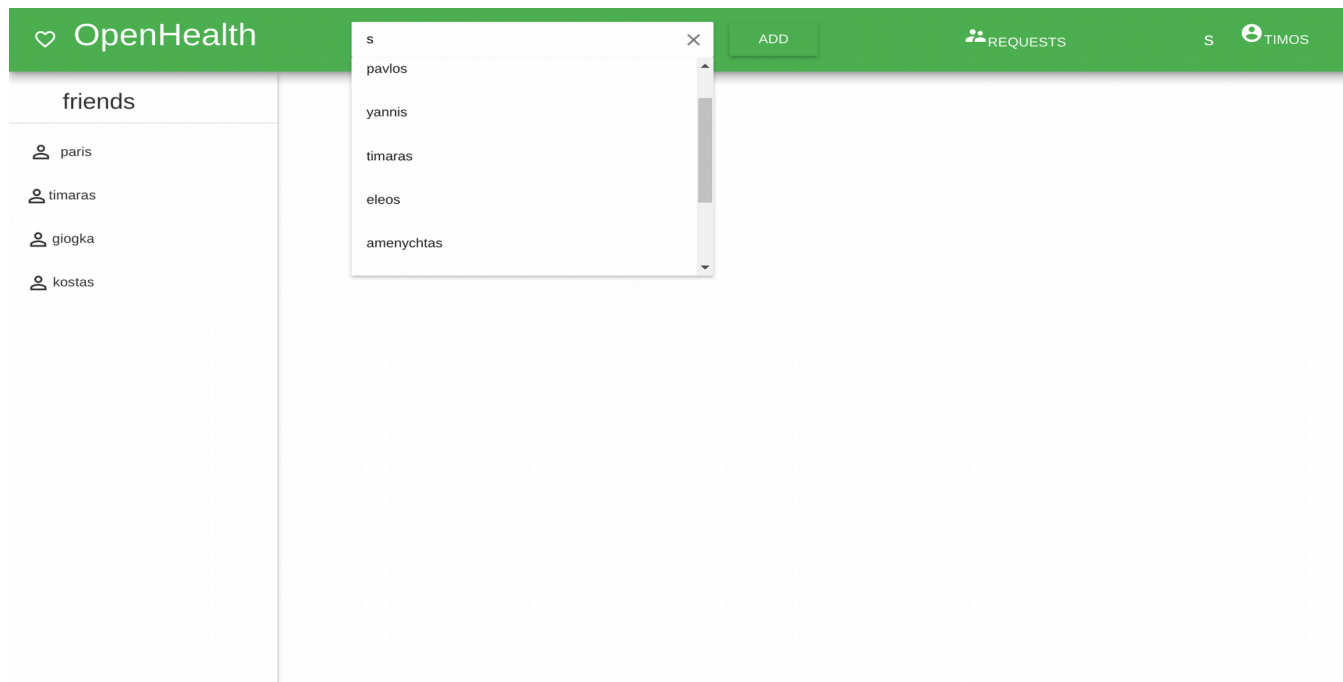
Σχήμα 29: Εισερχόμενα και εξερχόμενα αιτήματα φιλίας

Οι εφαρμογές των χρηστών του συστήματος διαθέτουν και αυτές δομές για την τοπική αποθήκευση και προβολή των αιτημάτων στους χρήστες, οι οποίες και πάλι επιλέχθηκαν να είναι τα Services της AngularJS. Έτσι κάθε χρήστης διαθέτει τα εισερχόμενα και εξερχόμενα αιτήματα φιλίας.

#### 4.3.4 Αυτόματη Συμπλήρωση Ονομάτων Φίλων.

Όπως είναι κατανοητό, τα αιτήματα φιλίας είναι ο μόνος τρόπος ένα χρήστης να δηλώσει ότι επιθυμεί να γίνει φίλος με κάποιον άλλον χρήστη του συστήματος. Αυτό όμως απαιτεί όταν ο χρήστης A θέλει να γίνει φίλος με τον χρήστη B να γνωρίζει την ακριβή ορθογραφία του ονόματος με το οποίο έχει εγγραφεί ο χρήστης. Αυτό θα μπορούσε να δημιουργήσει δυσκολίες στην εύρεση του ονόματος. Για την αντιμετώπιση του προβλήματος αυτού δημιουργήθηκε μία υπηρεσία η οποία θα προτείνει ονόματα στον χρήστη ανάλογα με τα γράμματα που έχει εισάγει έως τώρα. Συγκεκριμένα κάθε φορά που πληκτρολογεί ένα πλήκτρο ο χρήστης, στέλνεται ένα αίτημα AJAX που περιέχει ότι έχει πληκτρολογήσει ο χρήστης. Αφού ο Server λάβει το μήνυμα αυτό, αναζητά στη βάση αν υπάρχει κάποιο όνομα που να περιέχει ως συμβολοσειρά τη

μέχρι τώρα λέξη που πληκτρολόγησε ο χρήστης, και στέλνει πίσω ως μια λίστα το αποτέλεσμα της αναζήτησης αυτής σε μορφή JSON.



Σχήμα 30: Αναζήτηση και Αυτόματη Συμπλήρωση κάποιου χρήστη που περιέχει το γράμμα 's'

#### 4.3.5 Φίλοι και Κατάσταση Σύνδεσης των Φίλων ενός Χρήστη (State of Users – Online / Offline)

Οι φίλοι του χρήστη στην εφαρμογή SPA των πελατών είναι και αυτοί αποθηκευμένοι ως ένα Service. Συγκεκριμένα το Service αυτό, έχει αποθηκευμένο τα ονόματα των φίλων του χρήστη καθώς και την τρέχουσα κατάστασή τους, δηλαδή αν είναι συνδεδεμένοι ή όχι. Ο έλεγχος και η ενημέρωση ολόκληρου του συστήματος, σχετικά με τους χρήστες που είναι συνδεδεμένοι στο σύστημα κάθε στιγμή, γίνεται αποκλειστικά μέσω των websockets. Το σύστημα ελέγχει και ενημερώνει τις εφαρμογές των χρηστών όταν ένας χρήστης εισέρχεται ή αποσυνδέεται από το σύστημα. Έτσι δημιουργήθηκαν οι τρεις παρακάτω κατηγορίες για να ανταπεξέρχεται το σύστημα σε αυτές τις ενημερώσεις:

- Κάθε φορά που ένας χρήστης συνδέεται επιτυχώς στο σύστημα, τότε στέλνει στον Websocket Server το token που έχει λάβει και αν είναι έγκυρο ο Server τον προσθέτει στο σύστημα. Για κάθε χρήστη που εισέρχεται, ο server αποθηκεύει το token του, το όνομά του, τη σύνδεση μέσω της οποίας θα μπορεί να επικοινωνήσει μαζί του, το είδος του χρήστη, καθώς και ένα μοναδικό αναγνωριστικό (uuid), το οποίο είναι απαραίτητο καθώς το σύστημα δεν αρκεί να αναγνωρίζει έναν χρήστη από το όνομα του, καθώς μπορεί να υπάρχουν πολλαπλές συνδέσεις σε έναν

λογαριασμό, πράγμα που καθιστά απαραίτητο ο Server να είναι σε θέση να επιλέξει την κατάλληλη σύνδεση. Στη συνέχεια προσθέτει τον χρήστη στη λίστα.

```
var OnlineUser = {
  username: username,
  PortIp: portIp,
  token: token,
  SameAccount: Same,
  Id: id,
  category: category
};
this.list.push(OnlineUser);
```

- Όταν ο χρήστης εισέλθει επιτυχώς, ο Server αναζητεί τους φίλους του οι οποίοι βρίσκονται μέσα στην λίστα των online Users και τους στέλνει όσες βρήκε ώστε να φαίνονται ενεργοί στον χρήστη.
- Όταν ένας χρήστης αποσυνδέεται από το σύστημα, ο Websocket Server απομακρύνει τον χρήστη από τη λίστα και στέλνει ένα μήνυμα σε όλους χρήστες βρίσκονται online να αλλάξουν κατάσταση στον χρήστη.

```
var message = {type: 'UserGotOffline', name: Username}
```

Ειδική μέριμνα πρέπει να δοθεί στην περίπτωση που δύο ή παραπάνω συνδέσεις έχουν γίνει στον ίδιο λογαριασμό. Για κάθε μία από τις προηγούμενες περιπτώσεις έγιναν οι εξής τροποποιήσεις:

- Ένας χρήστης εισέρχεται στο σύστημα και το token που παρέχει είναι έγκυρο. Σε κάθε User προστίθεται ένα πεδίο που υποδηλώνει πόσες συνδέσεις υπάρχουν στο ίδιο όνομα. Αν δεν υπάρχει άλλη σύνδεση τότε η μεταβλητή τίθεται ίση με μηδέν. Διαφορετικά, αφού ο Server βρει πόσες συνδέσεις υπάρχουν, αποθηκεύει το πλήθος που βρήκε συν ένα σε όλες τις συνδέσεις.

```
var Same = 0;
for (var j = 0; j < this.list.length; j++) {
  if (this.list[j].username === username) {
    Same++;
    this.list[j].SameAccount++; // We increase all our Users by one
  }
}
```

- Ο Server αναζητεί τους συνδεδεμένους φίλους ενός χρήστη μόνο με βάση το όνομα του και όχι με το πλήθος των συνδέσεων σε έναν λογαριασμό επειδή μπορεί να υπάρχουν πάνω από μία.

- Όταν κάποιος χρήστης αποσυνδεθεί από το σύστημα, τότε ο Server κάνει αναζήτηση αν υπάρχουν άλλες συνδέσεις στον ίδιο λογαριασμό στο σύστημα. Αν υπάρχουν, τότε μειώνει το πλήθος των συνδέσεων που υπάρχουν στον ίδιο λογαριασμό κατά ένα σε κάθε μία από τις συνδέσεις που βρήκε. Αν αυτή είναι η μοναδική σύνδεση, τότε ενημερώνει τους φίλους του λογαριασμού ότι ο συγκεκριμένος χρήστης αποσυνδέθηκε από το σύστημα.

```

If (userLeft.SameAccount > 0) {
    for (var j = 0; j < this.list.length; j++) {
        if (Username === this.list[j].username)
            this.list[j].SameAccount--;
    }
}
else {
    relationship.findOne({user: Username}, function (err, result) {
        if (!result)
            return;
        else {
            for (var i = 0; i < _this.list.length; i++) {
                for (var j = 0; j < result.friends.length; j++) {
                    if (_this.list[i].username === result.friends[j] &&
                        _this.list[i].username !== Username) {
                        _this.list[i].PortIp.send(JSON.stringify({type:
                            'UserGotOffline', name: Username}));
                        break;
                    }
                }
            }
        }
    });
}
}

```

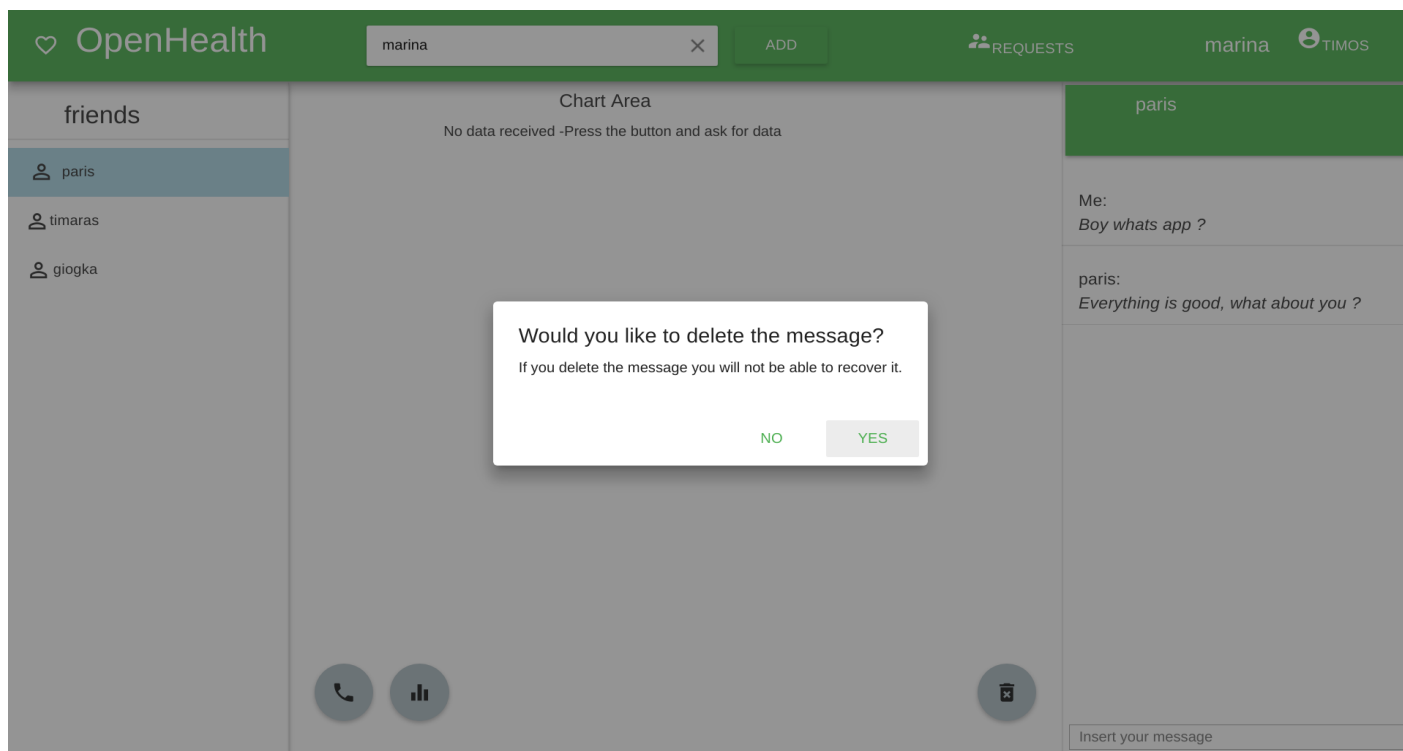
#### 4.3.6 Μηνύματα Γραπτά Κειμένου

Τα μηνύματα ήταν μία άλλη απαίτηση του συστήματος. Λόγω της φύσης τους τα μηνύματα μπορούν να αυξηθούν πάρα πολύ σε αριθμό. Αυτό δημιούργησε την ανάγκη να υλοποιηθεί ένας αποδοτικός τρόπος αναζήτησης, εισαγωγής τροποποίησης και απεικόνισης των μηνυμάτων στον χρήστη. Στον Server η αποθήκευση των μηνυμάτων ακολούθησε την παρακάτω λογική αποθήκευσης. Όταν δύο χρήστες γίνονται φίλοι τότε δημιουργείται μια νέα συλλογή η οποία περιέχει τους δύο χρήστες και ένα μοναδικό αναγνωριστικό (Conversation Id). Τα μηνύματα μεταξύ των δύο συγκεκριμένων χρηστών είναι αποθηκευμένα σε μια συνομιλία, την οποία ο Server αναζητεί με το μοναδικό αυτό αναγνωριστικό. Έτσι αποφεύγουμε ο Server να ελέγξει ολόκληρη την βάση ενώ ταυτόχρονα, τα CRUD ερωτήματα υλοποιούνται πολύ πιο αποδοτικά.

Ως συλλογή τα μηνύματα επιπλέον περιέχουν δύο πληροφορίες για την εύκολη διαχείριση: την κατεύθυν τους, δηλαδή τον αποστολέα και τον παραλήπτη, καθώς και ένα μοναδικό αναγνωριστικό το οποίο χρησιμοποιείται για την αναζήτηση και διαγραφή μηνυμάτων. Στην εφαρμογή του χρηστών τα μηνύματα αποθηκεύονται και αυτά ως ένα service. Επειδή ακριβώς το service πρέπει να ξέρει κάθε στιγμή προς ποιόν χρήστη αναφέρεται ένα μήνυμα, υπάρχει ένας πίνακας με τα μηνύματα για κάθε έναν από τους φίλους του. Κάθε φορά που ο χρήστης επιλέγει έναν φίλο του, εμφανίζονται και τα



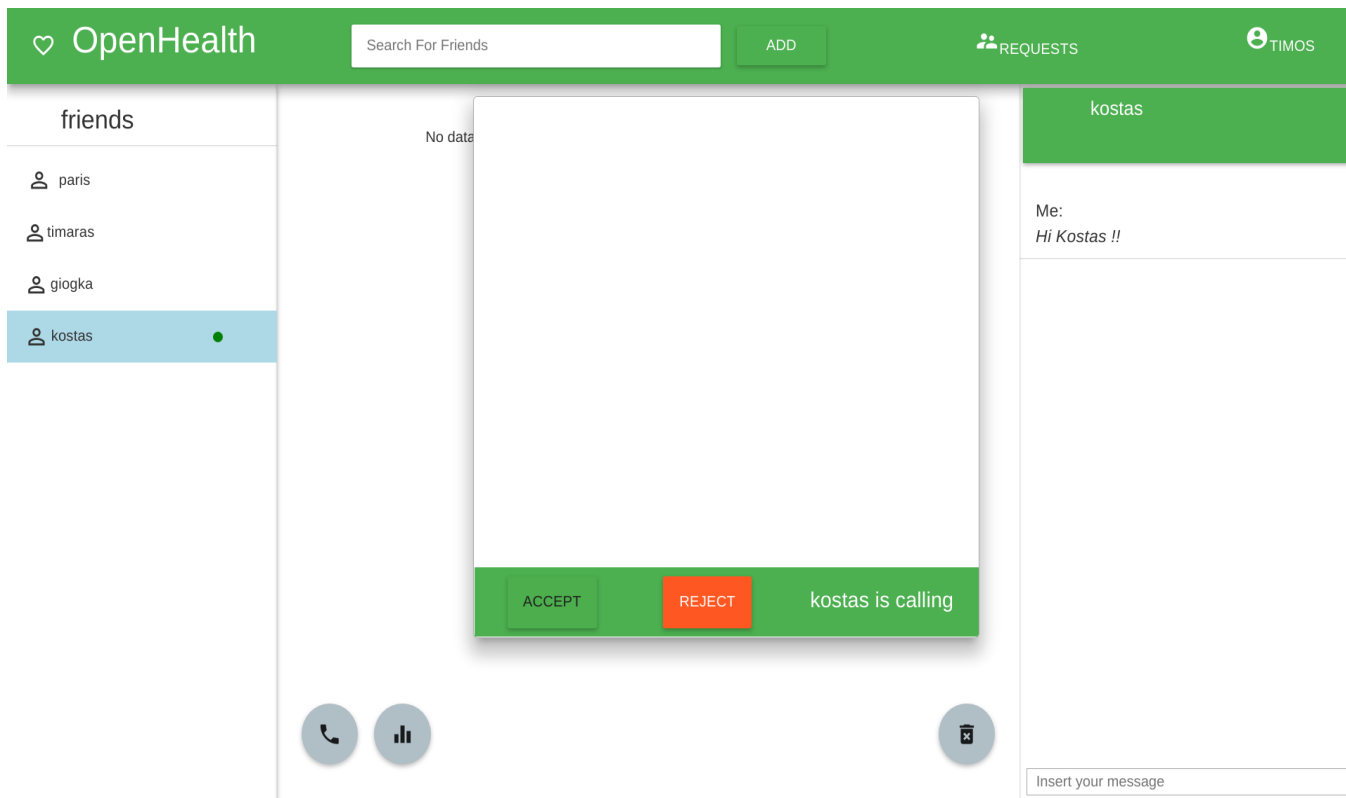
αντίστοιχα μηνύματα σχετικά με αυτόν. Κάθε στιγμή ο χρήστης μπορεί να βλέπει τα μηνύματα με μόνο έναν φίλο του. Έτσι αν επιλέξει πάνω στο εικονίδιο ενός φίλου του, εμφανίζονται τα μηνύματα που έχει ανταλλάξει με τον συγκεκριμένο χρήστη. Επιλέγοντας πάνω σε ένα συγκεκριμένο μήνυμα ο χρήστης μπορεί να το διαγράψει. Πριν διαγραφεί ένα μήνυμα η εφαρμογή ζητά επιβεβαίωση από τον χρήστη.



Σχήμα 31: Επιλογή μηνύματος και αναδυόμενο παράθυρο διαγραφής

#### 4.3.7 Σηματοδότηση – Singalling και WebRTC (Σχεδιαστικές Αποφάσεις )

Όπως εξηγήσαμε η σηματοδότηση είναι απαραίτητη ώστε να πραγματοποιηθούν οι βίντεο κλήσεις. Έτσι ο Server μας είναι υπεύθυνος για τη μεταβίβαση κάθε φορά του μηνύματος στον κατάλληλο αποδέκτη. Ο εξυπηρετητής δεν γνωρίζει αν κάποιος χρήστης είναι ήδη σε κλήση ή εάν μία εγκατάσταση άλλης κλήσης βρίσκεται σε εξέλιξη. Για όλες αυτές τις αρμοδιότητες είναι υπεύθυνες οι εφαρμογές SPA που τρέχουν στα μηχανήματα των χρηστών. Για παράδειγμα, οι εφαρμογές είναι υπεύθυνες να καθορίζουν αν ο χρήστης βρίσκεται σε άλλη κλήση, αν υπάρχει αναμονή για άλλη κλήση, ομαλή έκβαση της κλήσης κλπ. Έτσι ο Server έχει λιγότερες εργασίες, πράγμα που επιταχύνει την ταχύτητα εξυπηρέτησης του συστήματος. Η παραπάνω απαίτηση έχει υλοποιηθεί και αυτή με χρήση των Services της AngularJS. Παραδείγματα πληροφοριών που περιέχει το service αυτό είναι αν ο χρήστης βρίσκεται σε κλήση, τα ονόματα των συνομιλητών, τα μοναδικά αναγνωριστικά που τους χαρακτηρίζουν κ.α .



Σχήμα 32: Εισερχόμενη κλήση από φίλο του χρήστη

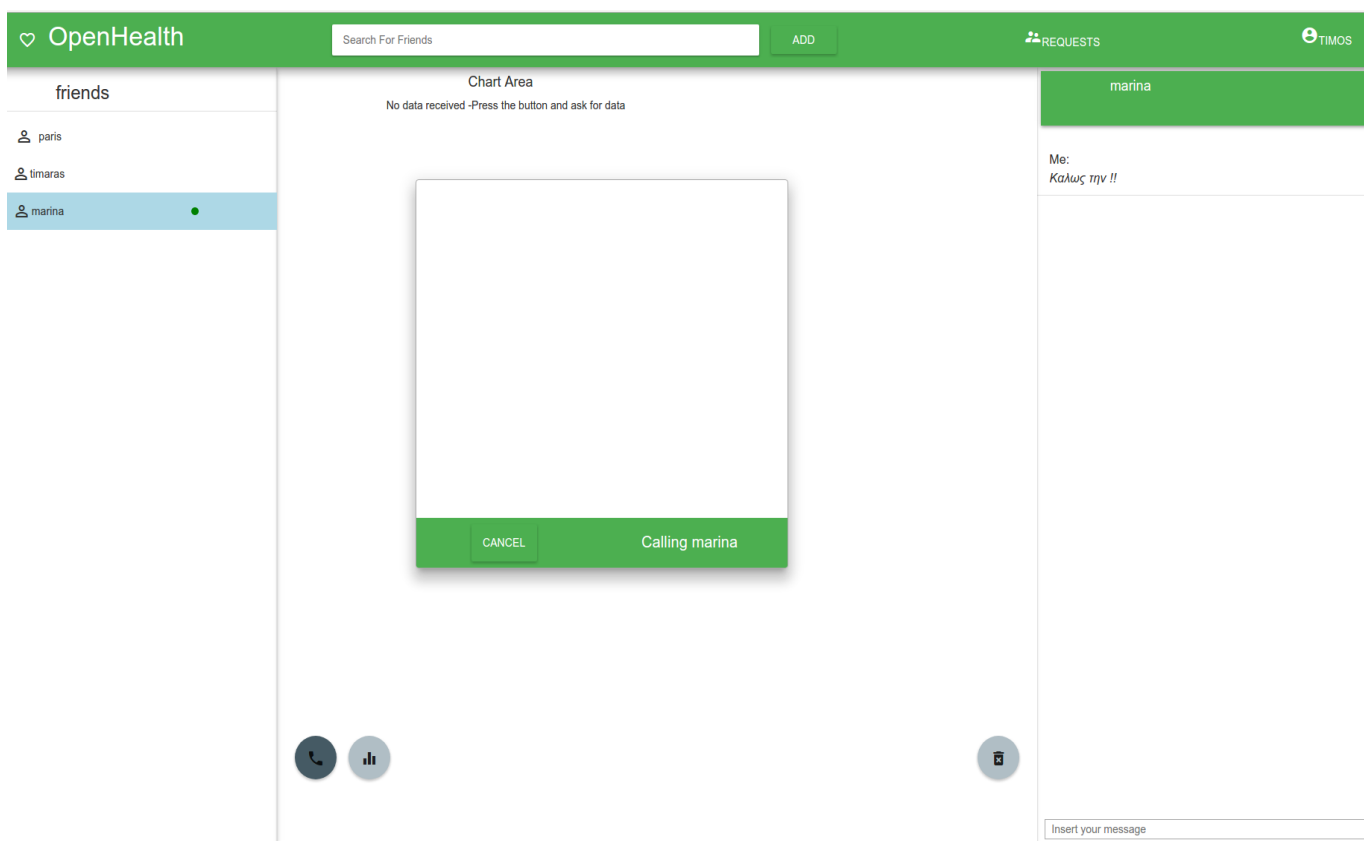
Τα μηνύματα που προωθεί ο εξυπηρετητής και αφορούν το Singalling χωρίζονται σε δυο βασικές κατηγορίες :

- Τα μηνύματα που θέλουμε να πάμε σε μια συγκεκριμένη σύνδεση, π.χ ο λογαριασμός του χρήστη A έχει δύο τρέχουσες συνδέσεις και μία από αυτές καλεί τον χρήστη B. Η απάντηση του B τότε στην εισερχόμενη κλήση πρέπει να εμφανιστεί στην συγκεκριμένη σύνδεση που τον κάλεσε.
- Τα μηνύματα που θέλουν να πάνε σε όλες τις συνδέσεις σε έναν λογαριασμό. Για παράδειγμα, ο χρήστης A καλεί τον B και όλες οι συνδέσεις του B θέλουμε να ενημερωθούν ότι υπάρχει εισερχόμενη κλήση . Έτσι αν κάποιος αποδέχεται μια κλήση, πρέπει να προσθέσει το δικό του αναγνωριστικό ενώ αν κάποιος ξεκινά μια κλήση δεν το προσθέτει. Στα παρακάτω δύο μηνύματα φαίνεται η διαφορά μεταξύ των δύο κατηγοριών.

```
var message = {
  type: 'video-response',
  target: VideoServices.getTarget(),
  source: my_name,
  answer: 'yes',
  sourceId: VideoServices.getMyid(),
  targetId: VideoServices.getTargetid()
};
```

```
var message = {
  type: 'video-start',
  source: my_name,
  target: username
};
```

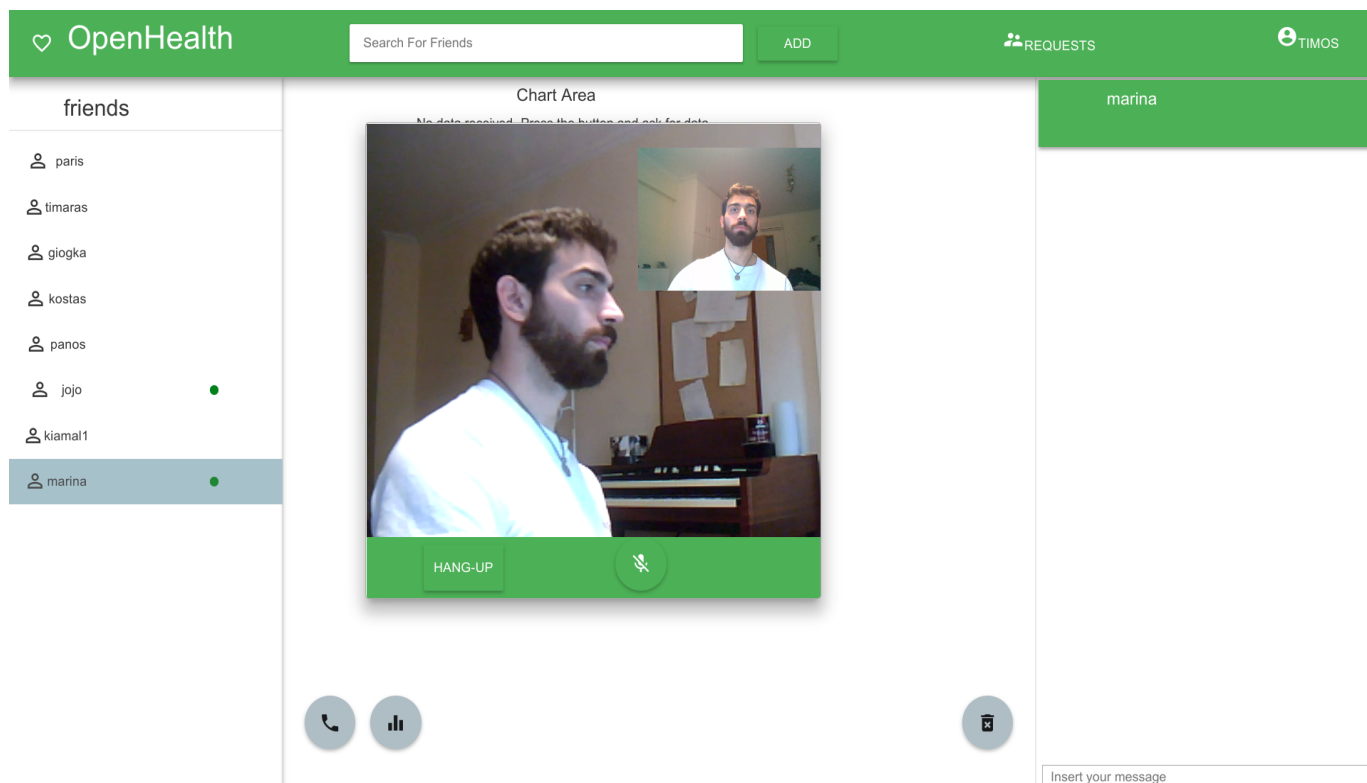
Η πραγματοποίηση και η ενεργοποίηση της σηματοδότησης και κατ' επέκταση της κλήσης ακολουθεί τα παρακάτω βήματα: Αρχικά ο χρήστης A επιλέγει το εικονίδιο του φίλου B που επιθυμεί να συνομιλήσει και πατάει το κουμπί για την κλήση. Στην οθόνη του χρήστη A εμφανίζεται ένα αναδυόμενο παράθυρο μέσα στο οποίο θα πραγματοποιηθεί η κλήση. Μήνυμα προωθείται στον Server περιέχοντας τους δύο χρήστες που θα συνομιλήσουν. Ο Server στέλνει το μήνυμα σε όλες τις συνδέσεις που υπάρχουν στον όνομα του χρήστη B και εμφανίζεται και σε αυτούς ένα αναδυόμενο παράθυρο που τους ενημερώνει ότι υπάρχει κλήση. Αν κάποια από τις συνδέσεις βρίσκεται σε άλλη κλήση τότε στέλνει μήνυμα πίσω στο Server να ενημερώσει τον καλών ότι δεν είναι διαθέσιμος. Διαφορετικά αν μια σύνδεση του χρήστη B αποδεχτεί ή απορρίψει την κλήση, αφού προσθέσει το μοναδικό της αναγνωριστικό, στέλνει ένα μήνυμα στον Server να ενημερώσει τις άλλες συνδέσεις του B ότι η κλήση απαντήθηκε ή απορρίφθηκε αντίστοιχα. Στη συνέχεια ενημερώνει τον χρήστη A ότι ο B αποδέχτηκε την κλήση, οπότε αρχίζει η αποστολή των κατάλληλων πληροφοριών



Σχήμα 33: Ένας χρήστης του συστήματος ξεκινά μία κλήση

Η κλήση στο σύστημά μας αντιπροσωπεύεται από το Object `RTCPeerConnection()`, που παρέχεται από τους browsers. Το Object αυτό περιέχει πληροφορίες που είναι απαραίτητες για να εγκατασταθεί η κλήση, όπως τα `IceCandidates` τα οποία καθορίζουν την διαδρομή που θα ακολουθούν τα δεδομένα στο διαδίκτυο, το μέγεθος των πακέτων

και άλλες παραμέτρους, καθώς και τα streams, δηλαδή την εικόνα και τον ήχο που πρέπει να μεταφερθεί μεταξύ των δύο χρηστών. Αφού το object δημιουργήσει όλες αυτές τις παραμέτρους, μέσω των μεθόδων που περιέχει τότε η εφαρμογή SPA, στέλνει στον Server τα μηνύματα που περιέχουν και ο Server τα προωθεί καταλλήλως.



Σχήμα 34: Δύο χρήστες του συστήματος βρίσκονται σε κλήσης

#### 4.3.8 Μεταφορά και Απεικόνιση Βιοσημάτων.

Η απεικόνιση των βιοσημάτων έγινε μέσω framework nvd3. Η χρήση του framework καθιστά απαραίτητο να σταλούν μόνο οι τιμές των βιοσημάτων. Έτσι μέσα στο δίκτυο μεταφέρονται αποκλειστικά δύο πίνακες, ένας για την οξυγόνωση του αίματος και ένας για τους καρδιακούς παλμούς. Όταν ένας χρήστης επιθυμεί να δει τα δεδομένα ενός ασθενή, του στέλνει ένα αίτημα και αν ο ασθενής το δεχτεί, οι παραπάνω δύο πίνακες στέλνονται μέσω του δικτύου, σε μορφή JSON. Οι μετρήσεις ζητούνται με ένα προκαθορισμένο εύρος (τελευταίες 24 ώρες). Δίνεται επίσης η δυνατότητα ο χρήστης να ζητήσει άλλο εύρος βιοσημάτων (π.χ τρεις μέρες ή μια εβδομάδα) για ακριβέστερη ή εκτενέστερη εποπτεία. Όταν συμβαίνει αυτό, νέο αίτημα πάει στο χρήστη ασθενή και αυτός απαντά στο αίτημα στέλνοντας τις νέες μετρήσεις ενώ οι προηγούμενες διαγράφονται. Ο λόγος που σχεδιάστηκε έτσι, ήταν η σχεδιαστική απλότητα καθώς και η ταχύτητα που τη συνοδεύει. Ο απλός χρήστη φυσικά δεν έχει τη δυνατότητα τροποποίησης των δεδομένων που λαμβάνει, αλλά μόνο τη εποπτεία τους.



Σχήμα 35: Τα βιοσήματα που έχει λάβει ένας χρήστης από έναν ασθενή του συστήματος

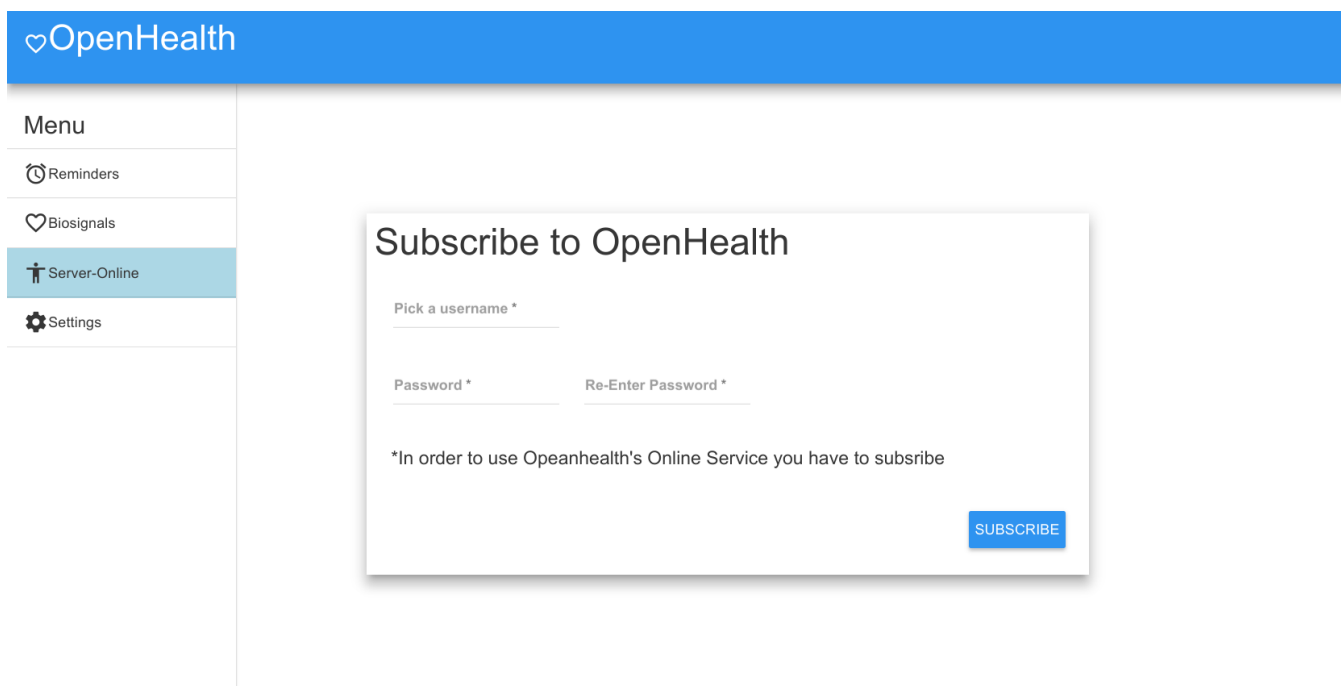
## 4.4 Υλοποίηση των απαιτήσεων χρηστών – ασθενών του συστήματος

Η εφαρμογή με την οποία οι χρήστες ασθενείς θα συνδέονται στο σύστημα είναι και αυτή ένας συνδυασμός ενός node.js τοπικού εξυπηρετητή (local Server), του front-end framework AngularJS και μίας βάσης mongo Database και θα τρέχει πάνω σε μία συσκευή Raspberry Pi. Η εφαρμογή του front-end μιλάει απευθείας και με τους δύο Servers, τόσο με http αιτήματα όσο και μέσω websockets. Όσον αφορά τις offline λειτουργίες αυτές θα εξυπηρετούνται από τον τοπικό server, οι ασύγχρονες λειτουργίες μέσω σύνδεσης Websockets και οι άλλες μέσω AJAX αιτημάτων, ενώ η πλειονότητα των online λειτουργιών θα εξυπηρετούνται από τον cloud Server. Ο λόγος που σχεδιάσαμε ο Client να μιλάει απευθείας με τον cloud – server ήταν για λόγους ευκολίας και εκμετάλλευσης του ήδη υπάρχοντος API, καθώς και ότι το UI του ασθενή σε επίπεδο online λειτουργιών είναι σχεδόν ταυτόσημο.

### 4.4.1 Σύνδεση στο Δίκτυο

Η σύνδεση στο δίκτυο απαιτεί ο ασθενής να εγγραφεί αρχικά στο σύστημα. Από τη στιγμή που ο χρήστης εγγραφεί επιτυχώς το σύστημα, ο τοπικός εξυπηρετητής έχει αποθηκευμένα στην βάση το όνομα και τον κωδικό πρόσβασης και αυτόματα τα στέλνει

στην εφαρμογή του ασθενή, κάθε φορά που εκκινεί την εφαρμογή του. Έτσι δεν χρειάζεται να τα εισάγει ο ίδιος χειροκίνητα.



The screenshot shows the OpenHealth application interface. At the top is a blue header with the OpenHealth logo. On the left is a vertical menu with options: Menu, Reminders, Biosignals, Server-Online (highlighted), and Settings. The main content area displays a 'Subscribe to OpenHealth' form. The form has three input fields: 'Pick a username \*', 'Password \*', and 'Re-Enter Password \*'. Below the fields is a note: '\*In order to use Opeanhealth's Online Service you have to subscribe'. A blue 'SUBSCRIBE' button is located at the bottom right of the form.

Σχήμα 36: Πρώτη εγγραφή του ασθενή για την σύνδεση στο σύστημα

#### 4.4.2 Υπενθυμίσεις

Όπως αναφέραμε στο κεφάλαιο 3 υπάρχουν δύο κατηγορίες υπενθυμίσεων: οι περιοδικές και οι μη περιοδικές. Οι υπενθυμίσεις είναι αποθηκευμένες στη βάση ως μία συλλογή. Ο τρόπος με τον οποίο γίνεται η αναζήτηση είναι ο εξής: Όταν ο local Server εκκινεί, μια συνάρτηση ορίζεται να τρέχει κάθε 20 δευτερόλεπτα (daemon), και η οποία κάνει αναζήτηση στη βάση δεδομένων για πιθανές υπενθυμίσεις που πρέπει να εμφανιστούν. Συγκεκριμένα, αν εντοπίσει μία υπενθύμιση η οποία έχει χρόνο εμφάνισης με διαφορά από την τωρινή ώρα λιγότερο από 15 δευτερόλεπτα, τότε στέλνει κατάλληλο μήνυμα μέσω της websokcet σύνδεσης που υπάρχει στον ασθενή και ενημερώνει την εγγραφή ώστε να μην την ξαναεμφανίσει. Ειδικά πεδία χρειάστηκαν για να γνωρίζει η συνάρτηση αν μία περιοδική υπενθύμιση έχει εμφανιστεί στο χρήστη ή αν μία υπενθύμιση έχει τεθεί σε αναστολή και πρέπει να εμφανιστεί αργότερα. Για την ενεργοποίηση και τροποποίηση των υπενθυμίσεων υπάρχει ένα εύκολο και κατανοητό UI όπως φαίνεται παρακάτω.

Menu

Reminders

Biosignals

Server-Online

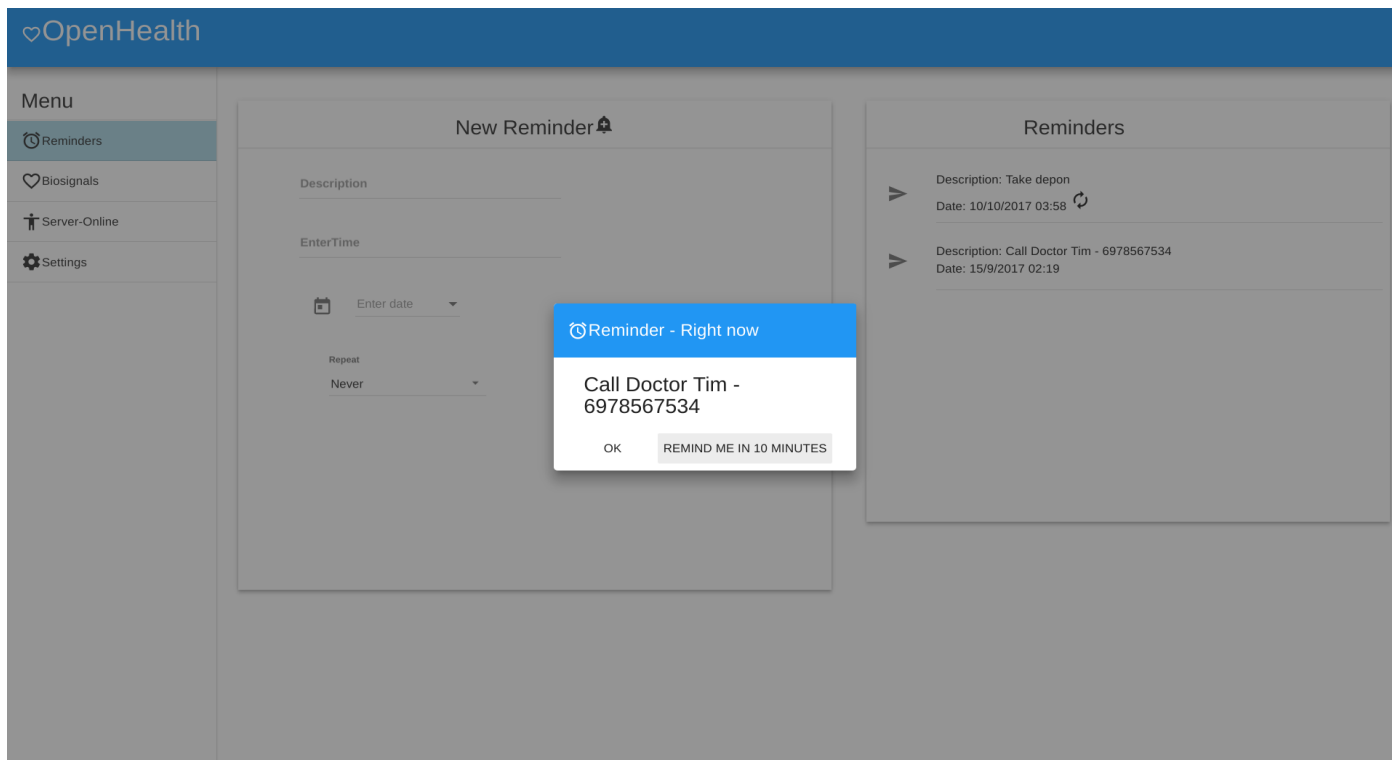
Settings

The image shows two side-by-side screenshots of the OpenHealth application's reminder management interface. On the left is the 'New Reminder' form, which includes a 'Description' text input, an 'EnterTime' text input, a date selection field with a calendar icon and a dropdown arrow, and a 'Repeat' dropdown menu currently set to 'Never'. A blue checkmark is visible at the bottom right of this form. On the right is the 'Reminders' list view, showing a single reminder with a right-pointing arrow, a description 'Take depon', and a date 'Date: 10/9/2017 03:58' with a refresh icon.

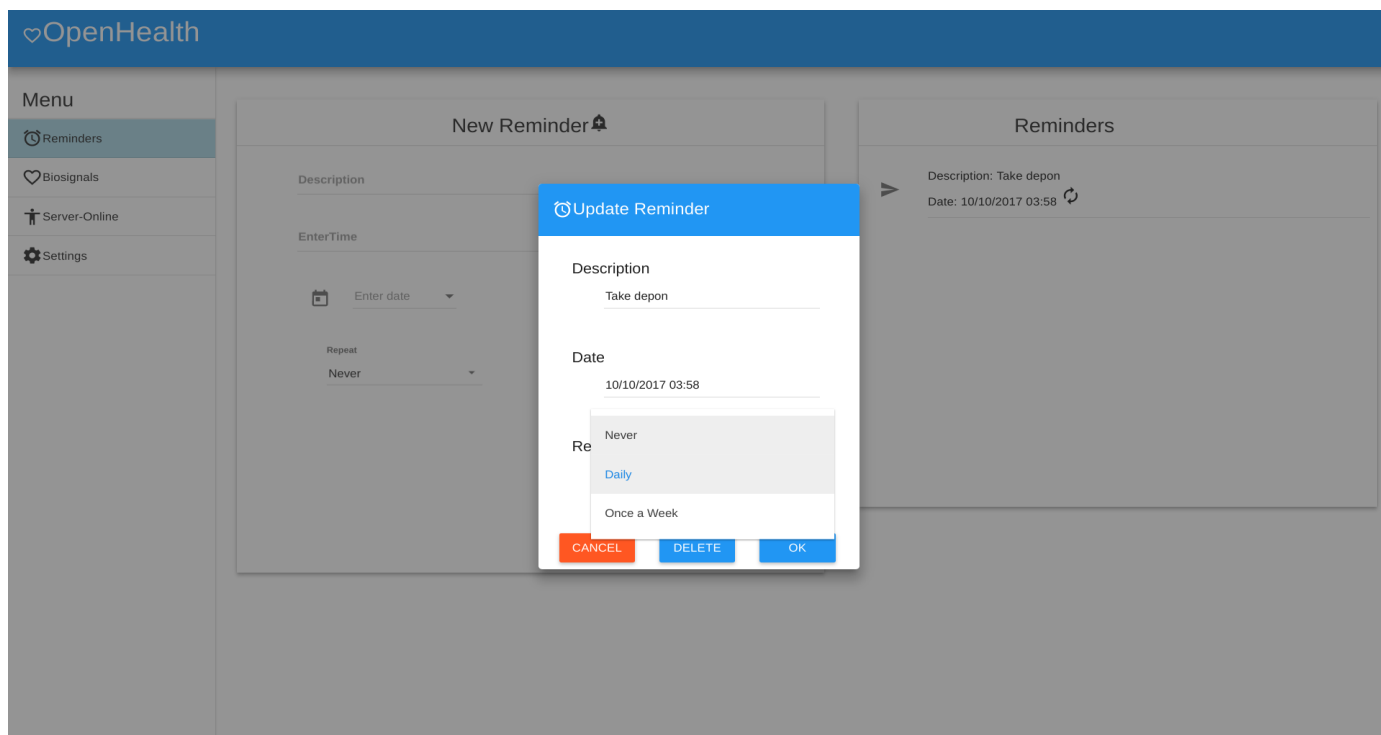
Σχήμα 37: User Interface για τις υπενθυμίσεις

Όσον αφορά τις υπενθυμίσεις στην εφαρμογή του ασθενή, αναπαρίστανται και αυτές ως ένα service, το οποίο έχει πληροφορίες όπως κείμενο, ώρα εμφάνισης, περιοδικότητα, αν βρίσκεται σε αναστολή. Η παραπάνω μορφή αποθήκευσης διευκολύνει την εύκολη αποθήκευση και τροποποίησή τους, αφού έχει χωρίσει τις παραμέτρους των υπενθυμίσεων, ενώ για την εμφάνιση και αλληλεπίδραση με τον χρήστη βρίσκονται κάτω από την επίβλεψη ενός αντίστοιχου controller (Reminder Controller).

Ο χρήστης μπορεί επιπλέον να τροποποιεί τις υπενθυμίσεις που έχει θέσει στο παρελθόν. Επιλέγοντας πάνω σε μια από τις υπενθυμίσεις, εμφανίζεται ένα νέο παράθυρο που περιέχει τις πληροφορίες της συγκεκριμένης υπενθύμισης. Η συγκεκριμένη λειτουργία δίνει στον χρήστη την δυνατότητα να τροποποιήσει όλες τις παραμέτρους της υπενθύμισης, το κείμενο της, την ώρα εμφάνισης, την περιοδικότητα της αλλά και να την διαγράψει από το σύστημα. Αφού ολοκληρώσει τα πεδία που επιθυμεί να τροποποιήσει ο χρήστης, στέλνει ένα AJAX request στον εξυπηρετητή και αν λάβει θετική απάντηση ενημερώνει και την εφαρμογή του πελάτη. Ο λόγος που χρησιμοποιήθηκε AJAX request αντί για αποστολή μέσω των websockets είναι ακριβώς αυτή η δυνατότητα της εφαρμογής να ενημερώνεται άμεσα για τυχόν σφάλματα που μπορεί να συμβούν και να μην περιμένει για ασύγχρονη απάντηση μέσω των websockets.



Σχήμα 38: Ενημέρωση του χρήστη για μια υπενθύμιση



Σχήμα 39: Τροποποίηση υπενθύμισης



```

var notification = new Schema({
  description:{
    type:'String',
    required:true
  },
  date:{
    type:"date",
    required:true
  },
  repeat:{
    type:'String',
    enum:[
      'Daily',
      'Never',
      'Weekly'
    ]
  },
  showedToday:{
    type:'bool',
    required:true
  },
  readyTochange:{
    type:'bool',
    required:true
  },
  uniqueId:{
    type:'String',
    required:'true'
  },
  remindFlag:{
    type:'bool',
    required:'true'
  },
  remindDate:{
    type:'date',
    required:'true'
  },
},

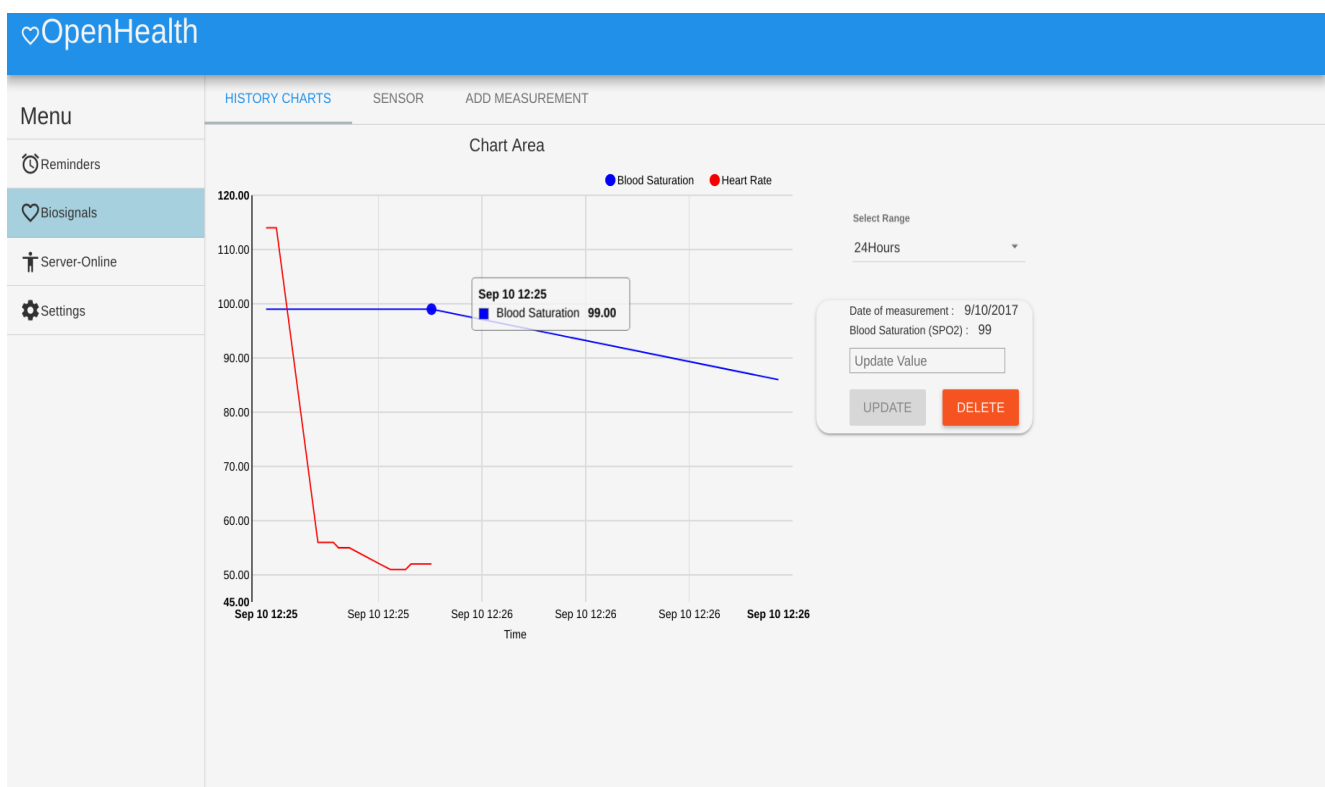
```

Σχήμα 40: Η συλλογή υπενθυμίσεων του ασθενή

### 4.4.3 Διαχείριση Βιοσημάτων

- **Απεικόνιση και Επεξεργασία Βιοσημάτων**

Τα βιοσήματα είναι και αυτά μια συλλογή αποθηκευμένη στη βάση δεδομένων του τοπικού Server ενώ όσον αφορά την εφαρμογή του χρήστη αναπαρίστανται και αυτά ως ένα Angular service και ελέγχονται από έναν κατάλληλο controller (BiosignalController). Η απεικόνισή τους γίνεται και εδώ μέσω του framework nvd3.js, έτσι κάθε φορά που ο χρήστης επιθυμεί να δει τα βιοσήματα του τότε η εφαρμογή ενημερώνει τον local Server και αυτός στέλνει τις μετρήσεις από τη βάση στον ασθενή με το προεπιλεγμένο εύρος. Για διαφορετικό εύρος βιοσημάτων γίνεται κάθε φορά νέο ερώτημα στη βάση και τα παλιά δεδομένα ομοίως διαγράφονται από την εφαρμογή του ασθενή.

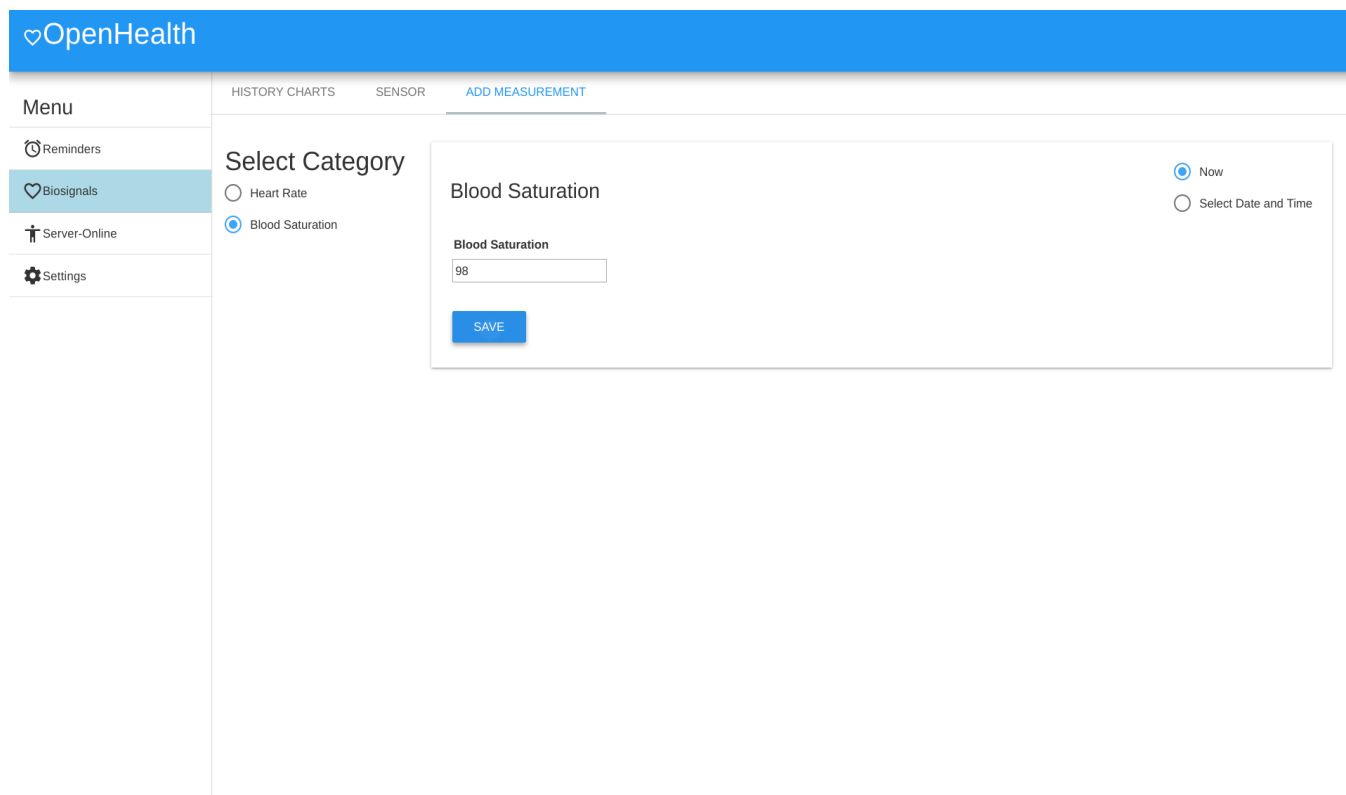


Σχήμα 41: Απεικόνιση και επεξεργασία βιοσημάτων`

- **Εισαγωγή Βιοσημάτων στη Βάση**

Η εισαγωγή των βιοσημάτων στο σύστημα μπορεί να γίνει, όπως έχουμε ήδη αναφέρει, με την χρήση του οξυμέτρου ή χειροκίνητα. Στην χειροκίνητη εισαγωγή δεδομένων ο χρήστης επιλέγει το κουμπί 'add measurement' το οποίο τον μεταφέρει στην κατάλληλη σελίδα. Εκεί έχει τη δυνατότητα να επιλέξει την κατηγορία του βιοσήματος που επιθυμεί να προσθέσει (οξυγόνωση ή καρδιακοί παλμοί). Αφού επιλέξει την κατηγορία πρέπει να επιλέξει και την ημερομηνία που το συγκεκριμένο βιοσήμα λήφθηκε. Η προεπιλεγμένη επιλογή είναι η δεδομένη στιγμή που

πραγματοποιείται η διαδικασία, αλλά μπορεί να επιλέξει και κάποια άλλη στιγμή χειροκίνητα. Αφού ολοκληρώσει τα παραπάνω, το σύστημα τον ενημερώνει για την επιτυχή ή όχι αποθήκευση του βιοσήματος.

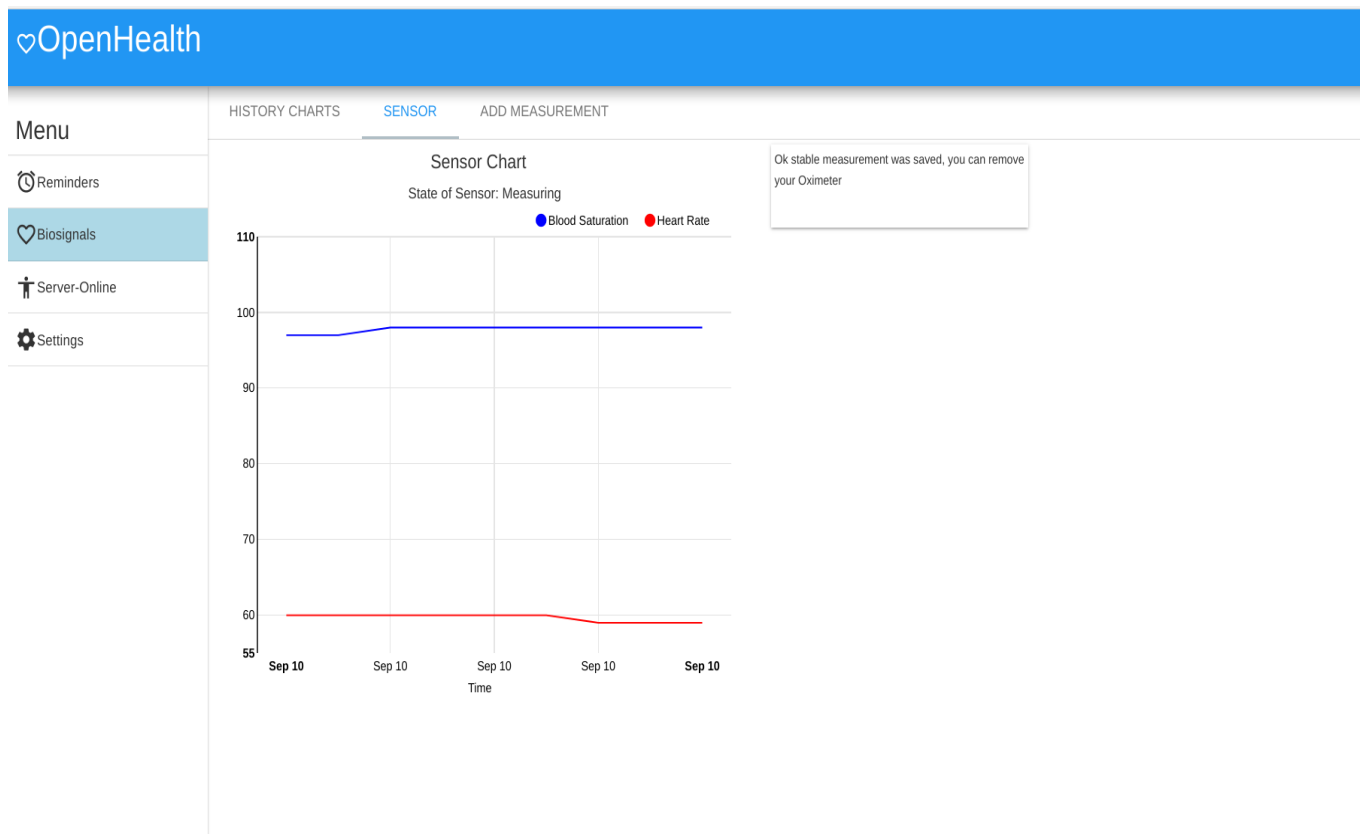


The screenshot shows the 'OpenHealth' application interface. On the left is a 'Menu' with options: Reminders, Biosignals (highlighted), Server-Online, and Settings. The main area has three tabs: 'HISTORY CHARTS', 'SENSOR', and 'ADD MEASUREMENT' (active). Under 'ADD MEASUREMENT', there is a 'Select Category' section with radio buttons for 'Heart Rate' and 'Blood Saturation' (selected). To the right, the 'Blood Saturation' form has a text input field containing '98' and a 'SAVE' button. At the top right of the form, there are two radio buttons: 'Now' (selected) and 'Select Date and Time'.

Σχήμα 42: Χειροκίνητη προσθήκη βιοσήματος στο σύστημα

Μία από τις λειτουργίες που θεωρήθηκε αξιόλογο να αναφέρουμε στην εισαγωγή μετρήσεων είναι ο τρόπος αποθήκευσης μέσω του οξυμέτρου. Συγκεκριμένα, όταν ο ασθενής εισάγει στο δάκτυλό του οξυμέτρο, τότε επικοινωνεί μέσω BLE με τον τοπικό εξυπηρετητή και στη συνέχεια αυτός στέλνει τις μετρήσεις που λαμβάνει στην εφαρμογή του ασθενή μέσω της websockets σύνδεσης. Στην συνέχεια η εφαρμογή απεικονίζει σε διάγραμμα τις τελευταίες δέκα μετρήσεις. Όμως, για την αποθήκευση μετρήσεων απαιτείται οι μετρήσεις να έχουν σχετικά μικρή απόκλιση μεταξύ τους για να θεωρηθούν έγκυρες. Η παραπάνω απαίτηση έχει υλοποιηθεί μέσω ενός αλγορίθμου γνωστού και ως Sliding Window. Αφού λάβει μετρήσεις ο Server τις αποθηκεύει προσωρινά σε έναν πίνακα και υπολογίζει κάθε φορά το μέσο όρο των τελευταίων πέντε μετρήσεων. Αν ο μέσος όρος των αποθηκευμένων μετρήσεων και της νέας μέτρησης απέχουν το πολύ κατά μία μονάδα, τότε η νέα μέτρηση θεωρείται σταθερή και αποθηκεύεται. Διαφορετικά η διαδικασία συνεχίζεται μέχρις ότου βρεθεί σταθερή ένδειξη. Επιπλέον όταν η σταθερή αυτή μέτρηση αποθηκευτεί τότε ο ασθενής δεν χρειάζεται να συνεχίσει να λαμβάνει μετρήσεις. Έτσι υλοποιήθηκε ασύγχρονα ο Server ώστε να ενημερώνει τον ασθενή ότι

αποθηκεύτηκε μέτρηση στην βάση και ότι μπορεί πλέον να σταματήσει να χρησιμοποιεί το οξύμετρο. Η ασύγχρονη αυτή επικοινωνία γίνεται φυσικά μέσω της Websocket σύνδεσης.



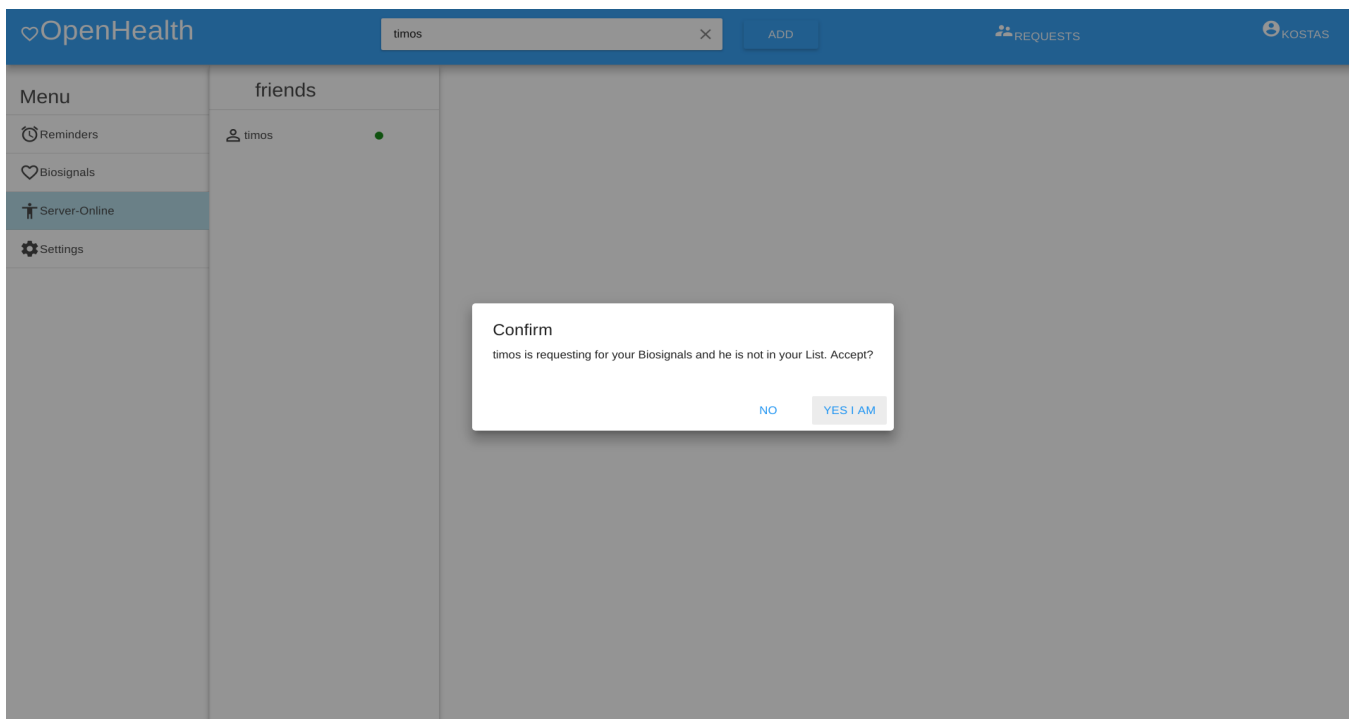
Σχήμα 43: Αποθήκευση μέτρησης μέσω οξύμετρου και ενημέρωση αποθήκευσης

### • Αποστολή Μετρήσεων Βιοσημάτων σε Πραγματικό Χρόνο

Τέλος υλοποιήθηκε και η αποστολή βιοσημάτων που λαμβάνει ο ασθενής την τρέχουσα χρονική στιγμή από μια υπάρχουσα βιντεοκλήση. Έτσι αν ο χρήστης χρησιμοποιεί το οξύμετρο και ταυτόχρονα βρίσκεται σε μία κλήση, τότε ο τοπικός εξυπηρετητής στέλνει τις μετρήσεις για απεικόνιση στην εφαρμογή του ασθενή και η εφαρμογή αυτή ασύγχρονα, μέσω της σύνδεσης Websocket που έχει με τον Cloud Server, στέλνει την ίδια μέτρηση στον συνομιλητή του. Έτσι και οι δύο βλέπουν στο πάνω αριστερά μέρος της βίντεο κλήσης τους τις μετρήσεις σε πραγματικό χρόνο. Κάθε νέα μέτρηση που στέλνει το οξύμετρο στον Server προωθείται αντίστοιχα και στους δύο χρήστες, έτσι και οι δύο βλέπουν κάθε στιγμή την τελευταία τιμή των βιοσημάτων.

Κάθε χρήστης έχει αποθηκευμένα στη βάση δεδομένων του τοπικού του εξυπηρετητή μια λίστα με τους χρήστες στους οποίους επιτρέπει να δουν τις μετρήσεις του. Κάθε φορά που εκκινεί η εφαρμογή, ζητά από τη βάση του Local Server τους χρήστες αυτούς και τους αποθηκεύει σε ένα service. Όταν έρθει ένα αίτημα για λήψη βιοσημάτων, τότε ο εφαρμογή ελέγχει αν το τρέχον όνομα υπάρχει στο service. Αν

υπάρχει, τότε στέλνει αυτόματα τις μετρήσεις. Αλλιώς εμφανίζεται μήνυμα στο χρήστη που τον ρωτά αν αποδέχεται ή όχι να στείλει τις μετρήσεις του στο συγκεκριμένο χρήστη. Αν ο ασθενής διαγράψει ή προσθέσει έναν χρήστη στην λίστα εξουσιοδοτημένων χρηστών, τότε αυτός διαγράφεται ή προστίθεται αντίστοιχα τόσο στην βάση του Server όσο και στο service της εφαρμογής πελάτη.



Σχήμα 45: Αίτημα λήψης βιοσημάτων από μη-εξουσιοδοτημένο χρήστη.

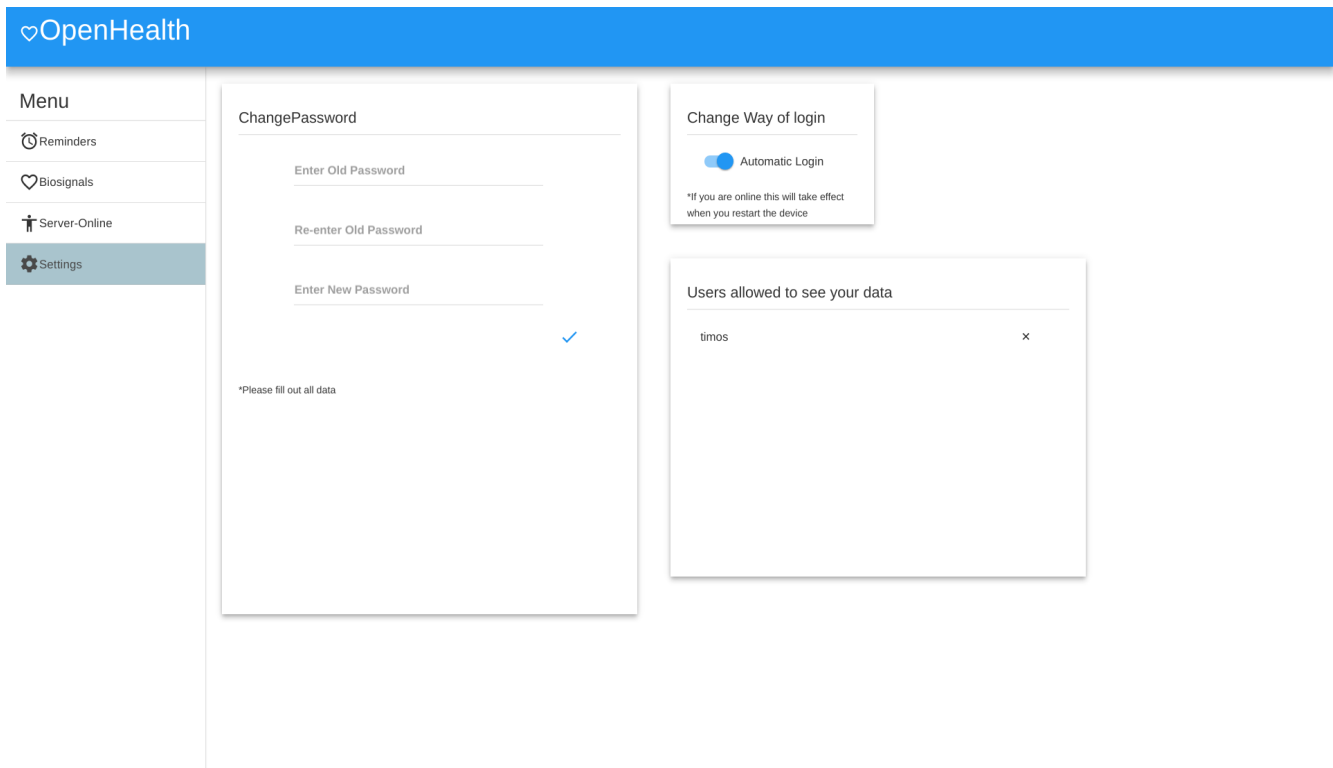
#### 4.4.4 Ρυθμίσεις Λογαριασμού

Μια ειδική σελίδα υπάρχει στην εφαρμογή ώστε να μπορεί να αλλάζει κάποιες ρυθμίσεις του λογαριασμού του.

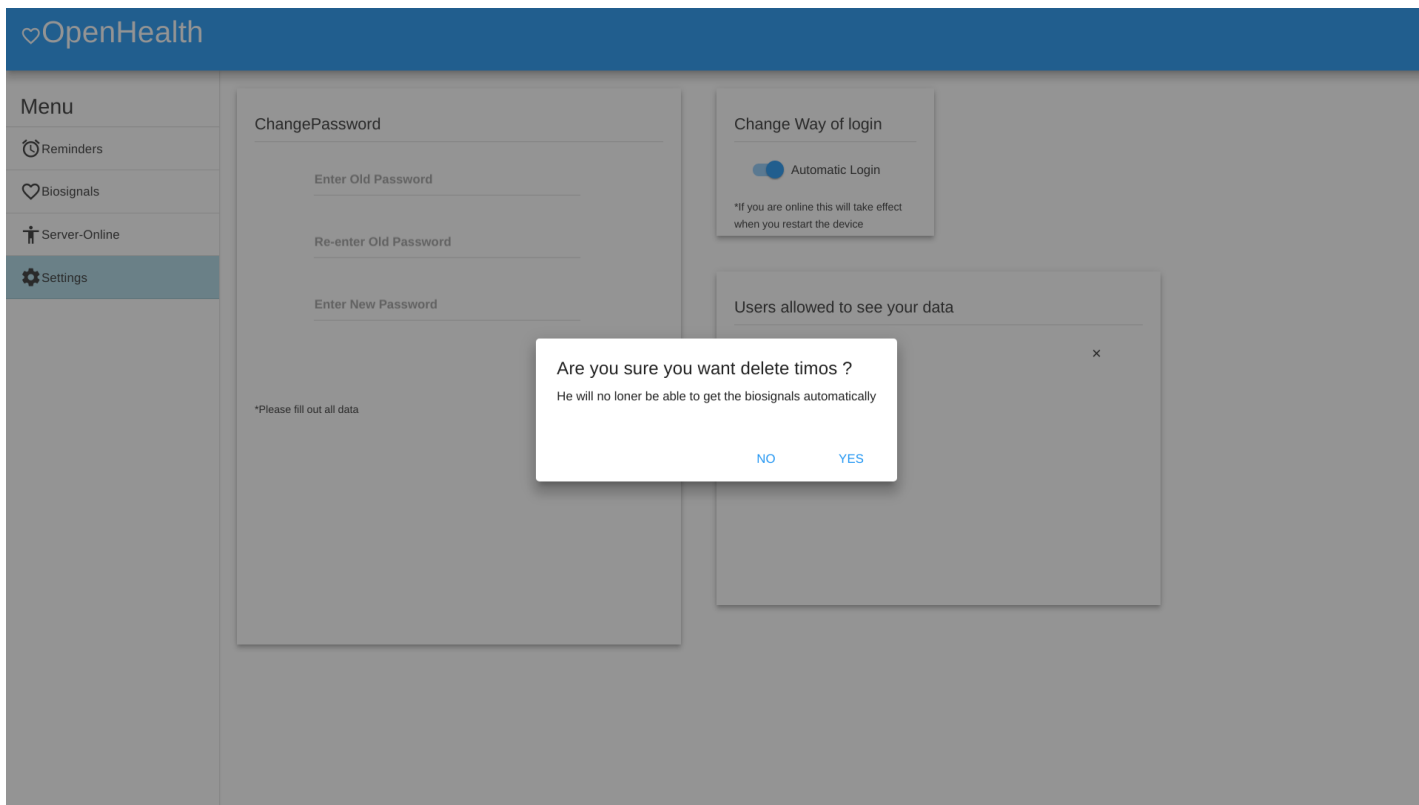
- Η πρώτη ρύθμιση δίνει τη δυνατότητα ο χρήστης να αλλάξει τον κωδικό πρόσβασης του. Συγκεκριμένα αφού πληκτρολογήσει τον νέο κωδικό, τον στέλνει στον Cloud Server και εάν είναι σωστός τότε ο κωδικός και ο χρήστης ενημερώνονται.
- Η δεύτερη ρύθμιση ελέγχει τον τρόπο εισαγωγής του ασθενή στο σύστημα. Οι επιλογές γίνονται αυτόματα ή χειροκίνητα. Κάθε μία από τις παραπάνω επιλογές αποθηκεύεται από τον τοπικό Server και έτσι κάθε φορά που εισάγεται ο χρήστης στο σύστημα, ακολουθείται η κατάλληλη επιλογή. Για ευκολία, σε επίπεδο χρήσης ως προεπιλογή έχει τεθεί ο αυτόματη σύνδεση στο δίκτυο. Σε κάθε εκκίνηση της εφαρμογής ο Server ελέγχει την επιλογή αυτή και αντίστοιχα είτε συνδέεται αυτόματα

είτε ο χρήστης επιλέγει να συνδεθεί. Σε καμία από τις δύο περιπτώσεις δεν απαιτείται ο χρήστης να πληκτρολογήσει τον κωδικό και το όνομά του, αλλά γίνονται αυτόματα.

- Η τρίτη και τελευταία ρύθμιση είναι η εποπτεία και ο έλεγχος των χρηστών στους οποίους ο χρήστης επιτρέπει να δούν τις μετρήσεις του. Οι χρήστες είναι αποθηκευμένοι στη βάση δεδομένων του τοπικού Server, ενώ αναπαρίστανται ως ένα Service στην εφαρμογή του ασθενή.



Σχήμα 46: Ρυθμίσεις λογαριασμού του απλού χρήστη.



Σχήμα 47: Διαγραφή χρήστη από την λίστα εξουσιοδοτημένων χρηστών.

Φυσικά ο χρήστης πρέπει να είναι σε θέση να διαγράψει κάποιον χρήστη από τη λίστα των εξουσιοδοτημένων χρηστών που διαθέτει. Η παραπάνω διαδικασία ολοκληρώνεται πατώντας πάνω στο όνομα του χρήστη που επιθυμεί να διαγράψει και εγκρίνοντας στη συνέχεια την προτροπή του αναδυόμενου παραθύρου.

## **5. Επίλογος - Συμπεράσματα**

### **5.1 Το Τελικό Σύστημα**

Στην παρούσα διπλωματική εργασία έγινε μια προσπάθεια να υλοποιηθεί ένα σύστημα απομακρυσμένης επικοινωνίας και παρακολούθησης ασθενών ή ατόμων περιορισμένης κινητικότητας. Το σύστημα σχεδιάστηκε με τέτοιο τρόπο ώστε να είναι εύχρηστο και γρήγορο στον ασθενή, πράγμα στο οποίο συνετέλεσαν τόσο οι τεχνολογίες που χρησιμοποιήθηκαν όσο και οι αρχιτεκτονικές και σχεδιαστικές αποφάσεις. Η εξέλιξη πλέον των frameworks, του διαδικτύου, αλλά και των δυνατοτήτων των καθημερινών υπολογιστών είναι τέτοιες που η ανάπτυξη εφαρμογών δεν απαιτεί πολλά έξοδα, ενώ οι εφαρμογές δεν στερούνται τίποτα σε ταχύτητα ή πολυπλοκότητα. Επιπρόσθετα, η εκμάθηση των frameworks είναι πλέον σχετικά σύντομη και εύκολη, ενώ οι δυνατότητες που παρέχουν είναι πολύ εξελιγμένες και έτσι πολλές εφαρμογές μπορούν να παραχθούν σχετικά εύκολα και αποδοτικά. Μέσω αυτών των δυνατοτήτων το υλοποιημένο σύστημα μπόρεσε να παράσχει στους ίδιους τους ασθενείς τη δυνατότητα να παρακολουθούν μόνοι τους (ή με βοήθεια τρίτων) την υγεία τους, ενώ δόθηκε και ένα ολοκληρωμένο κοινωνικό δίκτυο με σκοπό την επικοινωνία και παρακολούθηση του ασθενή. Το χαμηλό κόστος αγοράς και το μικρό μέγεθος του Raspberry-Pi, από το οποίο οι χρήστες – ασθενείς εισέρχονται στο σύστημα, καθιστά προσβάσιμη την απόκτησή του και τη χρήση του σε οποιαδήποτε τοποθεσία διαθέτει ένα wifi ή τρόπο σύνδεσης στο διαδίκτυο. Η απλή επίσης σχεδιαστική τεχνοτροπία βοηθάει τους άπειρους χρήστες να μάθουν γρήγορα το σύστημα, χωρίς έτσι να απαιτείται μεγάλος χρόνος εκμάθησης. Η δυνατότητα υπενθυμίσεων, η παρακολούθηση, επεξεργασία και λήψη βιοσημάτων αυτόματα και αξιόπιστα, είναι μια καλή αρχή για την συστηματική παρακολούθηση της υγείας ασθενών και ηλικιωμένων ανθρώπων.

### **5.2 Παρατηρήσεις – Προσφερόμενες Υπηρεσίες**

Το πλήθος των εφαρμογών για απομακρυσμένη ή μη παρακολούθηση της κατάστασης ασθενών με χρόνιες παθήσεις ή ηλικιωμένων ανθρώπων έχει αυξηθεί δραματικά τα τελευταία χρόνια. Αυτό έχει δημιουργήσει ένα τεράστιο σύνολο διαφορετικών υπηρεσιών συχνά αποκλειόμενων μεταξύ τους. Στην παρούσα διπλωματική εργασία έγινε μια προσπάθεια να δοθεί η δυνατότητα στον ασθενή να συνδυάσει την ιατρική του παρακολούθηση, τις καθημερινές ανάγκες ενώ παράλληλα να παράσχει εύκολη επικοινωνία με γιατρούς ή άλλα πρόσωπα μέσω του διαδικτύου. Στηριζόμενοι πάνω σε αυτό το σύστημα, ο ασθενής μπορεί να συνδυάσει αυτές τις, μέχρι στιγμής, ανεξάρτητες υπηρεσίες σε μια ενιαία υπηρεσία εκμεταλλευόμενος τη διάδοση του διαδικτύου και τη ραγδαία εξέλιξη των web browsers και των οικιακών προσωπικών υπολογιστών. Έτσι για παράδειγμα, η επικοινωνία με γιατρούς και η απομακρυσμένη παρακολούθηση των βιοσημάτων είναι πλέον πολύ εύκολη αφού παρέχονται όλες οι δυνατότητες μαζί. Φυσικά υπάρχουν πλατφόρμες οι οποίες είναι πολύ εξειδικευμένες σε κάποιους τομείς, αλλά η συγκεκριμένη εφαρμογή προσπαθεί να θέσει μια βάση για την ενσωμάτωση όλων αυτών σε ένα γενικότερο πλαίσιο. Σχηματικά



θα μπορούσαμε να παρουσιάσουμε το κέρδος ενός ασθενή από τη χρήση του συστήματος ως εξής:

<b>Υπηρεσίες που παρέχονται στον ασθενή</b>	<b>Με το σύστημα</b>	<b>Χωρίς το σύστημα</b>
Επικοινωνία (Βίντεο, μηνύματα)	Ενσωματωμένο μέσα στο σύστημα.	Ανάγκη για τρίτες υπηρεσίες ή κινητή τηλεφωνία.
Παρακολούθηση Υγείας	Καθημερινές Υπενθυμίσεις, εισαγωγή και εποπτεία των βιοσημάτων του.	Απαιτείται παρακολούθηση μέσω γιατρού ή άλλου προσώπου ή άλλης υπηρεσίας
Αποστολή Βιοσημάτων	Αποστολή Αποθηκευμένων Βιοσημάτων καθώς και αποστολή σε πραγματικό χρόνο.	Δεν είναι εφικτό.
Κοινωνικό Δίκτυο	Ενσωματωμένο στο σύστημα.	Μόνο μέσω άλλων υπηρεσιών.
Προαπαιτούμενα	Προσωπικός Υπολογιστής και σύνδεση στο διαδίκτυο	—

Όμως το σύστημα δεν μπορεί να θεωρηθεί ένα πλήρες σύστημα τηλεπαρακολούθησης ασθενών αλλά σίγουρα αποτελεί μια καλή βάση ώστε πάνω σε αυτή να αναπτυχθούν και να προστεθούν νέες δυνατότητες.

### **5.3 Θέματα για Περαιτέρω Διερεύνηση**

Το παραπάνω σύστημα μπορεί να χρησιμοποιηθεί ως μια βάση ώστε να εμπλουτιστεί με νέες δυνατότητες οι οποίες να παρέχουν στο χρήστη μια πιο ολοκληρωμένη εμπειρία σχετικά με ιατρικά θέματα. Μερικά μελλοντικά θέματα προς διερεύνηση θα μπορούσαν να είναι:

- Προσθήκη επιπλέον βιοσημάτων ή μετρήσεων πραγματικών παραμέτρων για μια πιο ολοκληρωμένη εποπτεία της υγείας του ασθενή, για παράδειγμα ασύρματη ζυγαριά για παρακολούθηση του βάρους, πιεσόμετρο για παρακολούθηση της πίεσης του αίματος, ψυχολογική κατάσταση, σωματική δραστηριότητα και άλλα.
- Η δυνατότητα στο σύστημα να αποθηκεύονται και να στέλνονται ιατρικές εξετάσεις που έλαβε ο ασθενής από κάποιον ιατρό ή νοσοκομείο ώστε ο γιατρός να μπορεί να τις παρακολουθεί απομακρυσμένα.
- Δυνατότητα για βίντεο κλήσεις που να περιέχουν πάνω από δύο συνομιλητές ώστε να μπορεί για παράδειγμα ο ασθενής να συνομιλήσει με δύο γιατρούς ή κάποιο συγγενικό του πρόσωπο να ακούει τι οδηγίες – συμβουλές δίνει ο γιατρός στον ασθενή.

- Άμα το σύστημα εντοπίσει κάποιο πολύ σοβαρό πρόβλημα υγείας (πολύ χαμηλούς παλμούς ή οξυγόνωση του αίματος) να επικοινωνεί αυτόματα με υπηρεσίες υγείας.
- Προσθήκη εφαρμογών – παιχνιδιών οι οποίες να ψυχαγωγούν τον ασθενή ώστε να βρίσκεται σε καλή ψυχική κατάσταση, το οποίο είναι καίριας σημασίας για την υγεία κάποιου ανθρώπου.
- Να παρέχει στους ασθενείς νέα για τις τελευταίες ιατρικές εξελίξεις ώστε να είναι ενήμεροι για τα ιατρικά δρώμενα καθώς και να παρέχει οδηγίες για διάφορα θέματα υγείας.

Πέρα από την προσθήκη όμως ιατρικών λειτουργιών και υπηρεσιών στο σύστημα, τροποποιήσεις θα μπορούσαν να υπάρξουν και στο υπόλοιπο μέρος της εφαρμογής για μια πιο ολοκληρωμένη πλατφόρμα. Μερικές από αυτές είναι :

- Προσθήκη και δεύτερου εξυπηρετητή για καλύτερο διαμοιρασμό των αιτήσεων και καλύτερο χρόνο εξυπηρέτησης.
- Δυνατότητα υπενθύμισης του κωδικού ή του ονόματος του χρήστη μέσω e-mail καθώς και η δυνατότητα αλλαγής του ονόματος των χρηστών.
- Λήψη και αποστολή αρχείων καθώς και φωτογραφιών μέσα στην εφαρμογή.
- Προσθήκη multi-threading στις εφαρμογές για καλύτερη εκμετάλλευση της πολυνηματικότητας των σύγχρονων υπολογιστών.

## Βιβλιογραφία

- [1] Wikipedia, [Online]. Available: <https://el.wikipedia.org/wiki/Καρδιά>.
- [2] Wikipedia, [Online]. Available: [https://en.wikipedia.org/wiki/Oxygen\\_saturation\\_\(medicine\)](https://en.wikipedia.org/wiki/Oxygen_saturation_(medicine)). Yahoo Answers, [Online]. Available: <https://answers.yahoo.com/question/index?qid=20101009083024AAAd197>.
- [3] Allnurses, [Online]. Available: <http://allnurses.com/nursing-student-assistance/spo2-vs-sao2-266863.html>.
- [4] Pneumonologos, [Online]. Available: <http://www.pneumonologos.net/#!/faqoxymetry/c1e9p>.
- [5] Department of Engineering Science, The University of Oxford, [Online]. Available: [https://www.robots.ox.ac.uk/~neil/teaching/lectures/med\\_elec/notes6.pdf](https://www.robots.ox.ac.uk/~neil/teaching/lectures/med_elec/notes6.pdf).
- [6] Wikipedia, [Online]. Available: [https://en.wikipedia.org/wiki/Pulse\\_oximetry](https://en.wikipedia.org/wiki/Pulse_oximetry).
- [7] Amperor Direct USA, [Online]. Available: <https://www.amperordirect.com/pc/helppulse-oximeter/z-what-is-pi.html>.
- [8] Wikipedia [Online] Available: [https://el.wikipedia.org/wiki/Web\\_browser](https://el.wikipedia.org/wiki/Web_browser).
- [9] W3schools [Online] Available: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp).
- [10] [Online]. Available : <https://www.w3.org/Style/CSS20/history.html>.
- [11] Oauth [Online]. Available: <https://auth0.com/blog/a-brief-history-of-javascript/>.
- [12] Techopedia [Online]. Available: <https://www.techopedia.com/definition/23240/browser-compatibility>.
- [13] Wikipedia [Online]. Available: [https://en.wikipedia.org/wiki/Ajax\\_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming)) .
- [14] «Google Trends - Ενδιαφέρον για Αναζήτηση Ιστού: adobe flex, extjs, gwt, vaadin - Παγκοσμίως, 2004 - παρόν» [Online]. Available: <http://www.google.com/trends/explore#q=adobe++flex,extjs,gwt,vaadin>.
- [15] Wikipedia [Online]. Available: [https://en.wikipedia.org/wiki/Single-page\\_application](https://en.wikipedia.org/wiki/Single-page_application).
- [16] [17] Wikipedia [Online]. Available : <https://en.wikipedia.org/wiki/WebSocket>.
- [17] AngularJS [Online]. Available: <https://docs.angularjs.org/guide/concepts>.
- [18] Wikipedia [Online]. Available: <https://en.wikipedia.org/wiki/XMLHttpRequest>.
- [19] Douglas Crockford “JavaScript: The Good Parts” ,Yahoo Press 2008
- [20] Wikipedia [Online]. Available: <https://el.wikipedia.org/wiki/Nodejs>.
- [21] Nodejs [Online]. Available: <https://nodejs.org/en/>.
- [22] Voidcanvas [Online]. Available: <http://voidcanvas.com/describing-node-js/>.
- [23] Leonard Richardson & Sam Ruby ” RESTfull Web Services “ O’Reilly Media, 2007
- [24] Nodejs [Online]. Available: [https://www.tutorialspoint.com/nodejs/nodejs\\_restful\\_api.html](https://www.tutorialspoint.com/nodejs/nodejs_restful_api.html)
- [25] Nodejs [Online]. Available: <https://oauth.net/2/>.
- [26] Wikipedia [Online]. Available: [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi).
- [27] Waqas Anwaar, “Energy Efficient Computing: A Comparison of Raspberry PI with Modern Devices”, International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 04, Issue 02, March 2015.

- [28] Adafruit, [Online]. Available: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction>.
- [29] ShopperTrack, [Online]. Available: <http://www.shoppertrak.com/what-is-bluetooth-low-energy-ble-bluetooth-smart/>.
- [30] Bluetooth.com, [Online]. Available: <https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technologybasics/low-energy>.
- [31] Wikipedia, [Online]. Available: [https://en.wikipedia.org/wiki/Use\\_case](https://en.wikipedia.org/wiki/Use_case)
- [32] Wikipedia, [Online]. Available: [https://en.wikipedia.org/wiki/Proxy\\_server](https://en.wikipedia.org/wiki/Proxy_server)
- [33] Wikipedia, [Online]. Available: <https://el.wikipedia.org/wiki/HTTPS>
- [34] Certbot, [Online]. Available: <https://certbot.eff.org/>
- [35] Wikipedia, [Online]. Available: [https://en.wikipedia.org/wiki/Apache\\_HTTP\\_Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server)
- [36] Kaniusas Eugenijus, "Biomedical Signals and Sensors I", Springer 2012
- [37] SangJoon Lee, Jungkuk Kim, & MyoungHo Lee, 'The Design of the m-Health Service Application Using a Nintendo DS Game Console' Telemedicine and e-Health, March 2011, 17(2): 124-130.
- [38] Yong-Gyu Lee, Won Sig Jeong, and Gilwon Yoon "Smartphone-Based Mobile Health Monitoring', Telemedicine and e-Health. October 2012, 18(8): 585-590.
- [39] Yusuke Sakaue & Masaaki Mikiawa, "Development of Wireless Biosignal Monitoring Device " [Information Technology Applications in Biomedicine, Nov. 2007](#), pp. 2168-2194
- [40] Farhad Abtahi, Jonathan Snall, Benjamin Aslami and Kaj Lindecrantz, "Biosignal PI, an Affordable Open-Source ECG and Respiration Measurement System" Sensors (Basel) 2014 Dec 23, 15(1): 93-109.