



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάπτυξη εργαλείου κοινωνικής δικτύωσης στον τομέα της υγείας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αγγελική Δ. Χρόνη

Επιβλέπων : Δημήτρης Ασκούνης
Καθηγητής Ε.Μ.Π.

Αθήνα, Μαρτίου 2020



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάπτυξη εργαλείου κοινωνικής δικτύωσης στον τομέα της υγείας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αγγελική Δ. Χρόνη

Επιβλέπων : Δημήτρης Ασκούνης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11^η Μαρτίου 2020.

.....
Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

.....
Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.

.....
Χρυσόστομος Δούκας
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2020

.....

Αγγελική Δ. Χρόνη

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

Copyright © ΑΓΓΕΛΙΚΗ ΧΡΟΝΗ, 2020

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 η ανάπτυξη της Ιατρικής τις τελευταίες δεκαετίες είναι ραγδαία, το πρόβλημα της επαρκούς και αποτελεσματικής περίθαλψης παραμένει. Η διαχείριση των ιατρικών δεδομένων εκκρεμεί καθώς δεν υπάρχει μια ασφαλής και εύκολα προσβάσιμη βάση δεδομένων στην οποία να καταγράφεται το ιστορικό των ασθενών. Η ιδιωτικότητα αυτών των δεδομένων είναι ζωτικής σημασίας και αυτό είναι και το σημαντικότερο πρόβλημα. 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 τεχνολογία Blockchain. Η τελευταία έχει την δυνατότητα να διασφαλίσει την ιδιωτικότητα και να πιστοποιήσει την εγκυρότητα των δεδομένων.

Βασικός σκοπός της διπλωματικής αυτής εργασίας, είναι η ανάλυση, η σχεδίαση και η ανάπτυξη ενός πληροφοριακού συστήματος υγείας, το οποίο περιέχει μια σειρά από δικτυακές εφαρμογές ώστε να διευκολυνθεί η καταγραφή των ιατρικών δεδομένων των ασθενών και συνεπώς και η έρευνα και η εργασία των ιατρών. Με την βοήθεια της τεχνολογίας Blockchain διασφαλίζονται όλοι οι όροι ιδιωτικότητας. Η εφαρμογή αυτή στοχεύει στην βελτίωση του τρόπου παρακολούθησης αλλά και στην αύξηση της συμμόρφωσης των ασθενών με τις οδηγίες που τους αναθέτονται από το ιατρικό προσωπικό. Η διαδικτυακή εφαρμογή παρέχει στο ιατρικό προσωπικό μια πληθώρα δυνατοτήτων και χρήσιμων ενεργειών. Αυτές οι ενέργειες και δυνατότητες έχουν να κάνουν κυρίως με την ανάθεση οδηγιών σε ασθενείς και την παρακολούθηση της πορείας της υγείας τους. Η διαδικτυακή εφαρμογή αποτελείται από ένα εύχρηστο και φιλικό προς τον χρήστη περιβάλλον, εφοδιασμένο με τα κατάλληλα εργαλεία ώστε ο ασθενής να έχει την δυνατότητα να ενημερώνει το σύστημα για την εξέλιξη της υγείας του, καταχωρώντας τις μετρήσεις που του ζητούνται. Παρέχει επίσης δυνατότητες επεξεργασίας των προφίλ των ασθενών και των γιατρών, εισάγοντας ο καθένας τα στοιχεία του. Όσον αφορά τις τεχνολογίες που χρησιμοποιήθηκαν, ο κύριος εξυπηρετητής είναι υλοποιημένος με Django, ενώ για την διεπαφή του χρήστη - συστήματος έγινε χρήση του πλαισίου React Js.

Λέξεις Κλειδιά: Ανάπτυξη Εφαρμογών, Εξυπηρετητής, Βάση Δεδομένων, React, JavaScript, RESTful APIs, Redux, Blockchain, SPA, NPM, Django.

Abstract

The high healthcare quality is the most important need of humans. In recent decades, the evolution of medicine is rapid, however the problem of an adequate and effective healthcare remains. The management of medical data is pending as a secure and easily accessible database doesn't exist in order for the patients' medical history to be recorded. The medical data privacy is vital and this is the most important obstacle. The progress of the telecommunications and IT sectors provides a very effective solution to these problems. One of the applications related to these issues is Blockchain technology. The latter can safeguard privacy and certify data validity.

The main purpose of this thesis is to analyze, design and develop a health web application that provides various functionalities in order to facilitate the record of patients' medical data and the medical research. The Blockchain technology guarantees all terms of privacy. This application aims to improve the way of tracking the patients' health and increase the compliance of patients with the instructions given by the medical staff. Furthermore, the application provides useful functionalities to medical staff. This web application is a user-friendly interface and consists of a set of tools that gives the opportunity (to patients) to inform and update the system about their health, recording the required measurements. It also provides the ability to edit patients' and doctors' profiles. Relating to the used technologies, the server is deployed in Django framework and the front-end is developed with React Js.

Keywords: web app, server, database, React, JavaScript, RESTful APIs, Redux, Blockchain, SPA, NPM, Django.

Ευχαριστίες

Το παρόν έργο διεξήχθη στο Εργαστήριο Συστημάτων Αποφάσεων του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ του Εθνικού Μετσόβιου Πολυτεχνείου, υπό την επίβλεψη και καθοδήγηση του καθηγητή κ. Δημήτρη Ασκούνη. Η διπλωματική αυτή θεμελιώνεται στις κυριότερες αρχές που διέπουν την Ανάπτυξη Διαδικτυακών Εφαρμογών, αλλά ταυτοχρόνως διερευνά την ομορφιά και το μεγαλείο των πιο σύγχρονων διαδικτυακών τεχνολογιών του σήμερα.

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Δημήτρη Ασκούνη για την παρότρυνση πάνω στο συγκεκριμένο αντικείμενο, καθώς και την εμπιστοσύνη που έδειξε προς το πρόσωπό μου αναθέτοντάς μου την παρούσα διπλωματική εργασία. Ακόμα, τις θερμές μου ευχαριστίες στον Μιχαήλ Κοντούλη που με την εμπειρία, την καθοδήγηση και την υπομονή του συνέβαλε τα μέγιστα στην εκπόνησή της. Η καθοδήγησή του και η καλή διάθεση που μου έδειξε με βοήθησαν σημαντικά στην ολοκλήρωση αυτού του έργου.

Τέλος, αφιερώνω το παρόν έργο στην οικογένειά μου, για την αμέριστη στήριξη και εμπιστοσύνη που μου έδειξαν στις επιλογές και στις σπουδές μου. Ευχαριστώ τον αδερφό και την αδερφή μου που συνεισέφεραν με το δικό τους τρόπο ο καθένας στην προσπάθειά μου αυτή, καθώς επίσης και τους γονείς μου, οι οποίοι μου έμαθαν να ακολουθώ πάντοτε τα όνειρά μου και να επιστρατεύομαι τη γνώση ως σύμμαχο σε όλα μου τα προβλήματα. Θα ήθελα να ευχαριστήσω, ακόμη, τους φίλους μου για τη συμπαράσταση και την υπομονή τους καθ' όλη τη διάρκεια των σπουδών μου στο Ε.Μ.Π. Σας είμαι ευγνώμων για τον άνθρωπο στον οποίο έχω εξελιχθεί σήμερα.

Περιεχόμενα

Κεφάλαιο 1. Εισαγωγή.....	1
1.1 Αντικείμενο-Σκοπός.....	2
1.2 Οργάνωση Τόμου.....	4
Κεφάλαιο 2. Θεωρητικό Υπόβαθρο – Βασικές έννοιες	5
2.1 Κοινωνικό Δίκτυο	6
2.1.1 Ορισμός	6
2.1.2 Χαρακτηριστικά	7
2.1.3 Κοινωνικά Δίκτυα για τη βελτίωση του κλάδου παροχής υπηρεσιών υγείας	8
2.2 Ιατρικά Πληροφοριακά Συστήματα – Προβλήματα Διαχείρισης Ιατρικών Δεδομένων	9
2.3 Αξιοποίηση της τεχνολογίας Blockchain για τη διαχείριση των ιατρικών δεδομένων	11
2.3.1 Τα πλεονεκτήματα της χρήσης Blockchain έναντι άλλων παραδοσιακών καταναμημένων βάσεων δεδομένων	11
2.3.2 Blockchain Τεχνολογία για Βιοιατρικές εφαρμογές και εφαρμογές φροντίδας υγείας	12
Κεφάλαιο 3. Τεχνολογικό Υπόβαθρο	19
3.1 Τεχνολογίες Πληροφορικής και Επικοινωνιών	20
3.1.1 Φυλλομετρητές ιστού (Web browsers)	20
3.1.2 Ανάπτυξη web applications	21
3.1.3 Συμβατότητα web browsers	24
3.1.4 Javascript	25
3.1.5 Single Page Applications and Web frameworks.....	29
3.1.6 NPM.....	32
3.1.7 Αρχιτεκτονική REST.....	33
3.2 Τεχνολογία Blockchain	34
3.2.1 Κρυπτογράφηση δημόσιου κλειδιού.....	35
3.2.2 Αλγόριθμοι Κατακερματισμού	37
3.2.3 Μηχανισμοί Συναίνεσης (Consensus Mechanism).....	39
3.2.4 Περιβάλλον Εργασίας Blockchain.....	45
3.3 Framework	48
3.3.1 Django.....	48
3.3.2 React Js	49
3.3.3 Redux.....	50
Κεφάλαιο 4. Ανάλυση Απαιτήσεων Συστήματος.....	53
4.1 Γενική Περιγραφή	54
4.2 Απαιτήσεις Συστήματος	55
4.2.1 Λειτουργικές Απαιτήσεις Συστήματος.....	55

4.2.2	Μη Λειτουργικές Απαιτήσεις Συστήματος.....	57
4.3	Αρχιτεκτονική Εφαρμογής.....	58
4.3.1	Υποδομή Client και Σενάρια Χρήσης.....	58
4.3.2	Διεπικοινωνία Μεταξύ Υποδομών.....	67
Κεφάλαιο 5. Λειτουργικότητα και Υλοποίηση του Συστήματος.....		69
5.1	Λειτουργικότητα του συστήματος.....	70
5.2	Υλοποίηση του Συστήματος.....	71
5.2.1	Γενική Περιγραφή.....	71
5.2.2	Λεπτομέρειες Υλοποίησης Διεπαφών Χρήστη.....	72
5.2.3	Το τελικό σύστημα.....	79
Κεφάλαιο 6. Επίλογος.....		89
6.1	Σύνοψη.....	90
6.2	Μελλοντικοί Στόχοι.....	90
Βιβλιογραφία.....		91

Εικόνες

Εικόνα 1. Μεριδίο αγοράς browsers - Νοέμβριος 2019 (εικόνα από (W3Counter: Free Web Stats and Website Widgets, n.d.)).....	20
Εικόνα 2. Ιστορία μεριδίου χρήσης browsers (εικόνα από (W3Counter: Free Web Stats and Website Widgets, n.d.))	20
Εικόνα 3. Απόδοση browsers	21
Εικόνα 4. Γραφική μορφή ενός απλού HTML κειμένου	22
Εικόνα 5. Document Object Model μιας ιστοσελίδας	22
Εικόνα 6. Άποψη ιστοσελίδας με χρήση και χωρίς χρήση CSS	23
Εικόνα 7. Βρόχος συμβάντων της Javascript.....	26
Εικόνα 8. Web worker και κύριο νήμα εκτέλεσης.....	27
Εικόνα 9. Συνήθειες αλληλεπιδράσεις των MVC συστατικών	31
Εικόνα 10. Περιβάλλον RESTful υπηρεσίες ιστού.....	34
Εικόνα 11. Ψηφιακή Υπογραφή.....	36
Εικόνα 12. Η κατανάλωση ενέργειας του Bitcoin	40
Εικόνα 13. Η διαδικασία του PBFT.....	43
Εικόνα 14. Η διαδικασία επαλήθευσης.....	44
Εικόνα 15. Η δομή ενός block	46
Εικόνα 16. Περιβάλλον συστήματος εφαρμογής.....	55
Εικόνα 17. Η αλληλεπίδραση της React με την Redux	71
Εικόνα 18. Δομή του καταλόγου φακέλου του προγράμματος	72
Εικόνα 19. Απεικόνιση του κεντρικού πλοηγητή (με πράσινη περιγράμμιση) και της κεντρικής πάνω μπάρας με (μωβ περιγράμμιση).....	78
Εικόνα 20. Οθόνη Σύνδεσης	79
Εικόνα 21. Οθόνη Εγγραφής.....	79
Εικόνα 22. Η κεντρική σελίδα της εφαρμογής	80
Εικόνα 23. Προβολή των Υπενθυμίσεων.....	80
Εικόνα 24. Προβολή των εισερχόμενων αιτημάτων.....	80
Εικόνα 25. Προβολή των εξερχόμενων αιτημάτων	81
Εικόνα 26. Αναδυόμενο παράθυρο δημιουργίας δημοσίευσης	81
Εικόνα 27. Αναδυόμενο παράθυρο επεξεργασίας δημοσίευσης.....	81
Εικόνα 28. Προβολή των σχολίων	82
Εικόνα 29. Επεξεργασία Σχολίου	82
Εικόνα 30. Προβολή προφίλ	82
Εικόνα 31. Επεξεργασία προφίλ	83
Εικόνα 32. Διάγραμμα ιστορικού των σκορ υγείας	83
Εικόνα 33. Στατιστικές πίτες σχετικά με τα ποσοστά των απαντήσεων.....	83
Εικόνα 34. Ερωτηματολόγιο για το σκορ υγείας	84
Εικόνα 35. Οθόνη Ημερολογίου	84
Εικόνα 36. Προβολή των αιτημάτων	85
Εικόνα 37. Προβολή των Φίλων	85
Εικόνα 38. Προβολή των «Circle of Trust».....	85
Εικόνα 39. Προβολή των «Protected Members»	86
Εικόνα 40. Προβολή των «Care Team».....	86
Εικόνα 41. Προβολή του «Blockchain History».....	86
Εικόνα 42. Προβολή του λογαριασμού του προστατευόμενου μέλους.....	87

Κεφάλαιο 1. Εισαγωγή

Χωρίς αμφιβολία τα κυριότερα χαρακτηριστικά γνωρίσματα της δεκαετίας, που διανύουμε, είναι η έκρηξη της πληροφορίας και της επικοινωνίας. Σ' αυτή την έκρηξη, αδιαμφισβήτητος είναι ο ρόλος του διαδικτύου (internet). Με την ανάπτυξη και τη ραγδαία εξάπλωσή του έχει συρρικνωθεί δραματικά ο χρόνος απόκτησης και διάδοσης της πληροφορίας και έχει διευρυνθεί, χωρίς όρια, η δυνατότητα της επικοινωνίας. Το διαδίκτυο πέρα από την πληροφοριακή φύση που αρχικά είχε, γρήγορα κατάφερε να γίνει τόπος δραστηριοποίησης ελεύθερων επαγγελματιών, επιχειρήσεων αλλά και δημόσιων υπηρεσιών. Πολλές είναι πλέον οι διαδικτυακές εφαρμογές για λογαριασμό επιχειρήσεων που αποσκοπούν στην προσέγγιση νέων πελατών ή στην προσφορά υπηρεσιών ηλεκτρονικά (πχ. ηλεκτρονική διαφήμιση, ηλεκτρονικό εμπόριο κ.α.). Συνεπώς, το διαδίκτυο είναι πλέον αμετάβλητο τμήμα της καθημερινότητας των ανθρώπων.

Επιπλέον, 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, κοινωνικόοικονομικές και γεωγραφικές ανάγκες καθιστούν δύσκολη την ισόνομη και εύκολη πρόσβαση όλων των ομάδων σε αυτή. Σήμερα, περισσότερο από ποτέ, η υγεία είναι ένα ζήτημα που βρίσκεται στο επίκεντρο της προσοχής όλων των ανεπτυγμένων χωρών του κόσμου. Αποτελεί καθημερινή έννοια για τον καθένα από μας, που είτε αντιμετωπίζει κάποιο πρόβλημα υγείας ασήμαντο ή σημαντικό, είτε απλώς θέλει να ενημερωθεί για θέματα που άπτονται στον τομέα της υγείας.

Ένας τομέας που επιτρέπει ισόνομη και εύκολη πρόσβαση ανθρώπων στις υπηρεσίες υγείας και περίθαλψης είναι οι διαδικτυακές εφαρμογές υγείας. Οι τελευταίες δίνουν τη δυνατότητα απομακρυσμένης και συστηματικής παρακολούθησης των ασθενών, με σκοπό την αντιμετώπιση επειγόντων περιστατικών, που μέχρι πρότινος ήταν αδύνατη για ορισμένες κοινωνικές ομάδες. Ωστόσο, οι ρυθμοί εξέλιξης 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 στο χρήστη. Θα πρέπει να προσφέρουν όσες υπηρεσίες, είναι πιθανό να χρειαστεί ο χρήστης, στο βαθμό φυσικά που αυτό είναι εφικτό.

1.1 Αντικείμενο-Σκοπός

Σκοπός της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός εργαλείου κοινωνικής δικτύωσης, το οποίο θα προσφέρει μια ολοκληρωμένη εμπειρία τόσο στον χρήστη-ασθενή όσο και στον χρήστη-γιατρό και θα βασίζεται στις τελευταίες τεχνολογίες του παγκόσμιου ιστού (HTML5, CSS3, React JS, Redux, Django). Ως περιβάλλον εκτέλεσης του παραπάνω συστήματος επιλέχτηκε ο browser, στον οποίο θα τρέχει η εφαρμογή των απλών χρηστών που θα συνδέονται

στο σύστημα (ιατροί, φίλοι, συγγενείς). Επιπλέον, οι ασθενείς θα έχουν τη δυνατότητα 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 σε όλους τους συνδεδεμένους χρήστες στο σύστημα με τη χρήση του χρονολογίου του εκάστοτε ασθενή. Συνεπώς, η εφαρμογή αυτή θα είναι τόσο φιλική προς τον χρήστη όσο και αποτελεσματική στη μεταφορά και αποθήκευση σημαντικών ιατρικών δεδομένων.

Ένα σημαντικό ζήτημα που προβληματίζει την επιστημονική κοινότητα στις μέρες μας είναι η ασφάλεια και η ιδιωτικότητα των ιατρικών δεδομένων. Καθώς τα ιατρικά δεδομένα είναι εξ' ορισμού ευαίσθητης φύσεως, θα πρέπει να είναι διαθέσιμα μόνο σε εξουσιοδοτημένους χρήστες. Οι κανόνες και η πολιτική της παρούσας διπλωματικής σέβονται και εξασφαλίζουν τις σωστές προϋποθέσεις, ώστε τα ιατρικά δεδομένα να είναι ασφαλή. Για την επίτευξη όλων αυτών, αξιοποιήθηκε μια από τις πιο δημοφιλείς τεχνολογίες στον τομέα της κρυπτογράφησης, η τεχνολογία Blockchain, τα βασικά χαρακτηριστικά της οποίας αναφέρονται στη συνέχεια.

Το Blockchain είναι μια αναμφισβήτητη έξυπνη εφεύρεση η οποία θεωρείται δημιουργήμα ενός ατόμου ή μιας ομάδας ανθρώπων γνωστών με το ψευδώνυμο Satoshi Nakamoto. Η τεχνολογία Blockchain είναι, με απλούστερους όρους, μια αλυσίδα από δομές αμετάβλητων δεδομένων (δηλαδή block) την οποία διαχειρίζεται ένα δίκτυο υπολογιστών peer-to-peer (P2P). Ένα δίκτυο υπολογιστών peer-to-peer (ή P2P) είναι ένα δίκτυο που επιτρέπει σε δύο ή περισσότερους υπολογιστές να μοιράζονται τους πόρους τους ισοδύναμα. Κάθε ένα από αυτά τα block είναι συνδεδεμένο με κάποιο άλλο, κάνοντας χρήση εξειδικευμένων αλγορίθμων κρυπτογράφησης. Το δίκτυο Blockchain δεν ανήκει και δεν υπακούει σε καμία κεντρική οντότητα, αντιθέτως είναι μια κοινή και αμετάβλητη δομή δεδομένων, της οποίας οι πληροφορίες είναι προσβάσιμες από όλους. Μοιάζει σαν μια άφθαρτη ψηφιακή βιβλιοθήκη συναλλαγών και πληροφοριών, και επίσης ένα δίκτυο ανεξάρτητων οντοτήτων, οι οποίες αποτελούν από κοινού ένα δίκτυο. Το δίκτυο αυτό χρησιμοποιεί την επεξεργαστική ισχύ, τον αποθηκευτικό χώρο και το εύρος ζώνης (bandwidth) όλων των κόμβων. Όλοι οι κόμβοι του δικτύου έχουν ίσα δικαιώματα. Οι πληροφορίες που βρίσκονται στον έναν κόμβο, ανάλογα με τα δικαιώματα που καθορίζονται, μπορούν να διαβαστούν από όλους τους άλλους και αντίστροφα. Επίσης, οι τιμές που καταγράφονται στο Blockchain συγχρονίζονται μεταξύ τους και ένας μηχανισμός συναίνεσης παρέχει την κοινή λογική και την αποδεκτή εγκυρότητα των δεδομένων. Έτσι, η δομή στοιχείων μπλοκ αλυσίδας αποτελείται από τα ακόλουθα χαρακτηριστικά:

(α) Έναν ασφαλή κατάλογο από μπλοκ (κόμβους) που περιέχουν εγγραφές δεδομένων.

(β) Ένα δίκτυο peer-to-peer που περιέχει πανομοιότυπα παραδείγματα της δομής δεδομένων Blockchain.

(γ)Έναν μηχανισμό συναίνεσης , που εξασφαλίζει τη συγχρονισμένη ανάπτυξη του Blockchain, παρέχοντας μία αμετάβλητη δομή δεδομένων.

Η τεχνολογία Blockchain δημιούργησε τον πυρήνα ενός νέου τύπου διαδικτύου, παρέχοντας ασφαλή διανομή ψηφιακών πληροφοριών και πιστοποιώντας τη μη αντιγραφή τους. Ως εκ τούτου, οτιδήποτε αποθηκεύεται στο Blockchain είναι από τη φύση του αξιόπιστο, καθώς όλοι οι συμμετέχοντες είναι υπόλογοι για την αυθεντικότητα του. Η ασφάλεια των συναλλαγών διασφαλίζεται, επειδή όλα τα μέλη του δικτύου διατηρούν ένα πλήρες αντίγραφο του Blockchain και έτσι δεν είναι δυνατό για ένα μέλος να τροποποιήσει τα δεδομένα. Επίσης, τα μέλη σε ένα Blockchain δίκτυο μπορούν να δουν μόνο εκείνες τις συναλλαγές για τις οποίες είναι εξουσιοδοτημένοι, δηλαδή αυτές που σχετίζονται με τους ίδιους. Οι τόσο ξεχωριστές δυνατότητες που παρέχει το Blockchain επιβάλλουν μια ριζοσπαστική αλλαγή στον τομέα της ψηφιακής ασφάλειας και της ιδιωτικότητας. Επομένως, το Blockchain λειτουργεί ως μία και μοναδική πηγή αλήθειας.

1.2 Οργάνωση Τόμου

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

- Στο 2^ο Κεφάλαιο παρουσιάζονται οι σημαντικότερες θεωρητικές έννοιες, οι οποίες είναι άρρηκτα συνδεδεμένες με το έργο και θα βοηθήσουν στη διαμόρφωση μιας σφαιρικής εικόνας. Πιο συγκεκριμένα, θα αναλυθεί η τρέχουσα κατάσταση και ο βαθμός ψηφιοποίησης στον τομέα της υγείας, τα προβλήματα που υπάρχουν και πως η τεχνολογία Blockchain τα επιλύει, αναφέροντας διαφορετικές προσπάθειες και εφαρμογές που έχουν γίνει στο παρελθόν για συναφή αντικείμενα.
- Στο 3^ο Κεφάλαιο επιχειρείται μια προσέγγιση των ποικίλων τεχνολογιών ιστού που χρησιμοποιήθηκαν για την εξέλιξη της πλατφόρμας, ενώ παράλληλα τεκμηριώνεται η σημασία τους και ο λόγος για τον οποίο επιλέχθηκαν.
- Το 4^ο Κεφάλαιο είναι αφιερωμένο στην ανάλυση των απαιτήσεων και των περιπτώσεων χρήσης του συστήματος καθώς και της αρχιτεκτονικής που χρησιμοποιήθηκε.
- Στο 5^ο Κεφάλαιο παρουσιάζονται οι λεπτομέρειες της υλοποίησης, όπως η διεπαφή (User Interface) της εφαρμογής και μερικά μέρη του κώδικα.
- Στο 6^ο Κεφάλαιο παρουσιάζεται μια αποτίμηση της εργασίας ως σύνολο και πιθανές επεκτάσεις και βελτιώσεις της λειτουργικότητας του συστήματος.

***Κεφάλαιο 2. Θεωρητικό Υπόβαθρο
– Βασικές έννοιες***

Σκοπός του παρόντος κεφαλαίου είναι να εισαγάγει τον αναγνώστη στις βασικότερες έννοιες οι οποίες χρησιμοποιούνται σε αυτήν τη διπλωματική εργασία. Η παρούσα διπλωματική εργασία αποτελείται από ένα συνονθύλευμα τριών διαφορετικών αντικειμένων-τομέων, τον κοινωνικό τομέα, τον τομέα της υγείας και τον τομέα του λογισμικού. Για αυτόν τον λόγο, στο κεφάλαιο αυτό αναλύεται η χρήση της διαδικτυακής εφαρμογής από τρεις διαφορετικές πλευρές. Στην πρώτη παρουσιάζεται η πολύτιμη βοήθεια του κοινωνικού δικτύου στις εφαρμογές υγείας. Στη δεύτερη αναφέρεται ο βαθμός ψηφιοποίησης στον τομέα της υγείας, ενώ στην τρίτη αναλύονται οι τρόποι με τους οποίους η τεχνολογία Blockchain συμβάλλει στις εφαρμογές υγείας.

2.1 Κοινωνικό Δίκτυο

2.1.1 Ορισμός

Ο όρος κοινωνικό δίκτυο αναφέρεται σε έναν τύπο κοινωνικής δομής, ο οποίος αποτελείται από ένα σύνολο κοινωνικών οντοτήτων και ένα σύνολο δυαδικών δεσμών μεταξύ των οντοτήτων. Οι οντότητες αυτές μπορεί να είναι άτομα, κοινωνικές ομάδες, πολιτικά κινήματα κλπ. Η ιδέα του κοινωνικού δικτύου επιτρέπει τη χρήση ενός συνόλου μεθόδων με σκοπό την ανάλυση κοινωνικών δομών, η οποία με τη σειρά της επιτρέπει την αναγνώριση τοπικών και καθολικών προτύπων, τον εντοπισμό οντοτήτων με επιρροή και την εξέταση της κοινωνικής δυναμικής (Wasserman S., 1994).

Αν και οι πρώτες αναφορές σε κοινωνικά δίκτυα έγιναν στα τέλη του 19ου αιώνα, σημαντική ανάπτυξη στο πεδίο παρατηρήθηκε κατά τη δεκαετία 1970 από διαφορετικές ομάδες ερευνητών στα πεδία της ψυχολογίας, ανθρωπολογίας και των μαθηματικών ξεχωριστά. Η ανάλυση κοινωνικών δικτύων άρχισε να γίνεται όλο και πιο διάσημη στη δεκαετία των 1990, κατά την οποία αναπτύχθηκαν και εφαρμόστηκαν μοντέλα και μέθοδοι στα δεδομένα που άρχισαν να γίνονται διαθέσιμα μέσω του διαδικτύου και συγκεκριμένα μέσω των υπηρεσιών κοινωνικής δικτύωσης. Από τότε, τα κοινωνικά δίκτυα αποτελούν τη βάση της επιστημονικής έρευνας στην κοινωνική συμπεριφορά και κατέχουν κύριο ρόλο στην κατανόηση του τρόπου με τον οποίο οι άνθρωποι αντιδρούν σε καταστάσεις κρίσης, όπως σε περιπτώσεις μεγάλων εκδηλώσεων μολυσματικών ασθενειών, αναπηριών ή ατυχημάτων. Είναι σημαντικό να αναφερθεί ότι παρά το γεγονός ότι η κοινωνική δικτύωση ενός ατόμου μπορεί να επιτευχθεί άμεσα μέσω φυσικών ομάδων που υπάρχουν σε κάθε κοινωνία, όπως είναι οι πανεπιστημιακοί και εργασιακοί χώροι ή άλλες διαφορετικές ομάδες, σήμερα τα πιο δημοφιλή κοινωνικά δίκτυα είναι αυτά που βασίζονται στο διαδίκτυο. Συνεπώς, τα μέσα κοινωνικής δικτύωσης συνδέονται αλληπάλληλα με το διαδίκτυο.

Ένα τέτοιο μέσο κοινωνικής δικτύωσης βασίζεται στην αλληλεπίδραση των χρηστών μέσω μιας διαδικτυακής διεπαφής. Κάθε μέλος της κοινότητας δημιουργεί το

προφίλ του και έτσι γίνεται "κόμβος" στο δίκτυο. Η δικτύωση αυτή είναι ιδιαίτερα διαδραστική, δυναμική και σε πραγματικό χρόνο, με ελάχιστες χρηματοοικονομικές επενδύσεις. Επίσης, χρησιμοποιώντας μέσα κοινωνικής δικτύωσης αποκτάς μια διεθνή παρουσία στον κόσμο του διαδικτύου. Ένα από τα οφέλη της κοινωνικής δικτύωσης είναι η ελεύθερη και ανοιχτή προσβασιμότητα από όλους, χωρίς κοινωνικές διακρίσεις. Το κοινωνικό δίκτυο, επίσης, δίνει τη δυνατότητα επιλογής και προβολής συγκεκριμένων ομάδων ανάλογα με τις ανάγκες και τις απαιτήσεις του κάθε ατόμου. Τέλος, τα μεγάλα δίκτυα (εικονικού) διαδικτύου αποτελούν σημαντικό αντικείμενο ερευνών, καθώς έχει μεγάλο ενδιαφέρον η μελέτη τους.

2.1.2 Χαρακτηριστικά

Τα κοινωνικά δίκτυα διαθέτουν τα παρακάτω χαρακτηριστικά:

- Υποστηρίζουν ποικιλία των μορφών περιεχομένου, όπως κείμενο, βίντεο, φωτογραφίες, ήχο, κ.τ.λ. Πολλά από αυτά κάνουν χρήση περισσότερων του ενός από αυτές τις επιλογές ως προς το περιεχόμενο.
- Επιτρέπουν αλληλεπιδράσεις που παίρνουν μία ή περισσότερες πλατφόρμες μέσω διαμοιρασμού, email και τροφοδοσίες
- Χαρακτηρίζονται από διαφορετικά επίπεδα εμπλοκής του χρήστη οι οποίοι μπορούν να δημιουργήσουν, να σχολιάσουν ή να παρακολουθήσουν μέσω των κοινωνικών δικτύων.
- Απλοποιούν, βελτιώνουν την ταχύτητα και το εύρος της διάδοσης των πληροφοριών
- Προσφέρουν ενός- προς-ένα, ενός-προς-πολλούς και πολλών προς-πολλούς επικοινωνία
- Επιτρέπουν την επικοινωνία αυτή να πραγματοποιείται είτε σε πραγματικό χρόνο ή ασύγχρονη με την πάροδο του χρόνου
- Είναι ανεξάρτητα της συσκευής: Ο χρήστης μπορεί να χρησιμοποιήσει για τη συμμετοχή σε κοινωνικά δίκτυα, έναν υπολογιστή, ή κινητές συσκευές (tablets και smartphones ιδιαίτερα)
- Επεκτείνει εμπλοκή με τρεις τρόπους: με τη δημιουργία σε πραγματικό χρόνο online εκδηλώσεις, με την επέκταση σε απευθείας σύνδεση αλληλεπιδράσεις offline εκδηλώσεις, και τελευταία με την υποστήριξη ζωντανών εκδηλώσεων
- Τα μέσα κοινωνικής δικτύωσης, επιπρόσθετα, στέκονται ικανά να συντελέσουν στην προβολή αφενός της κοινής γνώμης, καθώς καθιστούν τους χρήστες δέκτες αλλά και εκδότες περιεχομένου μέσω της ανατροφοδότησης, ενώ παρέχουν κοινωνική και συναισθηματική υποστήριξη (Evans, 2008). Αφετέρου, τα ΜΚΔ αποτελούν εξαιρετικό σύμμαχο στην προώθηση συγκεκριμένων ιδεών, αξιών, ακόμη και προϊόντων (Μάρκετινγκ), ώστε επιχειρήσεις, πολιτικοί, ομάδες συμφερόντων κ.α. 'μάχονται' για την αλλοίωσή της, στο σκληρό πλαίσιο του ανταγωνισμού (Μέσα κοινωνικής δικτύωσης - Βικιπαίδεια, n.d.).

2.1.3 Κοινωνικά Δίκτυα για τη βελτίωση του κλάδου παροχής υπηρεσιών υγείας

Τα κοινωνικά δίκτυα που βασίζονται στο διαδίκτυο επιτρέπουν, τόσο την επικοινωνία και τη συνεργασία, όσο και τη συλλογή πληροφοριών στον τομέα της υγείας. Περισσότεροι από τους μισούς (55%) Αμερικάνους, αναζητούν θεραπεία και πληροφορίες σχετικά με ένα πρόβλημα υγείας στο διαδίκτυο, ενώ το ένα τρίτο αυτών χρησιμοποιεί τα κοινωνικά δίκτυα για αυτόν τον σκοπό. Περίπου το 60% των ιατρών στις ΗΠΑ χρησιμοποιούν κοινωνικά δίκτυα για επαγγελματικούς σκοπούς. Η χρήση κοινωνικών δικτύων στον τομέα της υγείας επιτρέπει:

- Την αύξηση της επικοινωνίας και της συνεργασίας, καθώς οι ασθενείς ανταλλάσσουν πληροφορίες μεταξύ τους για παρόμοια προβλήματα υγείας και αντίστοιχα οι ιατροί μπορούν να κοινοποιούν τις εμπειρίες τους σχετικά με τη θεραπεία ή την αντιμετώπιση ενός προβλήματος. Ως αποτέλεσμα, καλύτερες αποφάσεις μπορούν να παρθούν σχετικά με τη θεραπεία του ασθενούς.
- Οι ασθενείς επίσης μπορούν να ενημερώνουν άμεσα σχετικά με την πορεία της ασθένειάς τους και τη θεραπεία τους, τους επιβλέποντες ιατρούς τους. Ακόμη, μπορούν εύκολα να ανακαλύψουν τις επιλογές θεραπείας που εφαρμόζονται σε διαφορετικά κέντρα υγείας. Αυτή η ανατροφοδότηση μπορεί να επηρεάσει θετικά την παραπομπή των ασθενών σε άλλα επίπεδα υγειονομικής περίθαλψης ή άλλες ζωτικής σημασίας διαδικασίες στη λήψη αποφάσεων.
- Τα παγκόσμια κοινωνικά δίκτυα Internet επιτρέπουν την ανταλλαγή πληροφοριών χωρίς περιορισμούς χρόνου και χώρου. Αυτό αυξάνει τον αριθμό των δυνητικών χρηστών υπηρεσιών υγείας.
- Η κοινωνική δικτύωση μπορεί να χρησιμοποιηθεί για την προώθηση, την ενίσχυση της εργασιακής ικανότητας και της γνώσης των ενδιαφερομένων (Social Networks in Improvement of Health Care, n.d.).

Μέσα σε αυτά τα παγκόσμια κοινωνικά δίκτυα, που γεννήθηκαν μέσα από τη φιλοσοφία του Web 2.0 (Syngress, 2010), είναι δυνατό να σχηματιστεί μια προσανατολισμένου ενδιαφέροντος ομάδα. Σύμφωνα με έρευνα που πραγματοποιήθηκε τον Νοέμβριο του 2008, η αναζήτηση στο Facebook της λέξης κλειδί "breast cancer" εμφανίζεται σε 620 ομάδες με 1.090.397 μέλη που μοιράζονται πληροφορίες σχετικά με το θέμα του καρκίνου του μαστού. Η συντριπτική πλειονότητα αυτών των ομάδων ιδρύθηκε για να συγκεντρώσει πληροφορίες και έρευνες σχετικά με τη θεραπεία (44,7%). Ένα μεγάλο μέρος αυτών δημιουργήθηκε για την προώθηση της ευαισθητοποίησης σχετικά με το πρόβλημα (38,1%) και πολλές από αυτές έχουν σχηματιστεί από ιδρύματα που στοχεύουν στην υποστήριξη των ασθενών (61,9%) με καρκίνο του μαστού (Bender JL, 2011).

Με την ίδια μεθοδολογία τον Απρίλιο του 2011, οι λέξεις-κλειδιά "καρδιακή νόσος Βοσνία και Ερζεγοβίνη" εμφανίστηκαν σε 4 ομάδες και σε 3421 άρθρα. Οι συγκεκριμένες ομάδες δημιουργήθηκαν μόνο για τη συγκέντρωση πληροφοριών σχετικά με τη θεραπεία. Η ποιότητα, η αξιοπιστία και η συνάφεια των δεδομένων που λαμβάνονται μέσω της ανταλλαγής από αυτά τα παγκόσμια δίκτυα είναι όμως αμφισβητήσιμα. Πρόκειται για δίκτυο ανοικτής πρόσβασης, όπου οι πληροφορίες δίνονται αδιάκριτα και ως εκ τούτου θα πρέπει αυτό να λαμβάνεται υπόψιν. Δυστυχώς, είναι σπάνια η ύπαρξη διαχειριστή στις ομάδες, έτσι ώστε να αναλαμβάνει το φιλτράρισμα δεδομένων. Τέτοιου είδους διαχειριστές είναι πιο πιθανό να βρεθούν μόνο σε κοινωνικά δίκτυα που δημιουργήθηκαν για συγκεκριμένους σκοπούς. Αλλά ακόμη και εκεί οι διαχειριστές δεν είναι πιστοποιημένο ότι έχουν τις απαραίτητες γνώσεις σε αυτό τον τομέα, οπότε είναι αμφίβολη η έγκυρη αξιολόγηση των δεδομένων (Μέσα κοινωνικής δικτύωσης - Βικιπαίδεια, n.d.).

2.2 Ιατρικά Πληροφοριακά Συστήματα – Προβλήματα Διαχείρισης Ιατρικών Δεδομένων

Σύμφωνα με την πλειοψηφία της επιστημονικής κοινότητας, ο κλάδος της παροχής υπηρεσιών υγείας, βρίσκεται ακόμα πολύ πίσω σε θέματα τεχνολογίας αλλά και διαχείρισης της πολύτιμης ιατρικής πληροφορίας, αν το συγκρίνουμε με άλλους επιχειρησιακούς κλάδους, όπως του λιανικού εμπορίου, του τουρισμού ή της βιομηχανίας. Σύμφωνα με το κυρίαρχο σενάριο ένας ασθενής επισκέπτεται ένα γιατρό, αυτός πραγματοποιεί μια εκτίμηση σχετικά με την υγεία του ασθενή, αυτή η εκτίμηση γνωστοποιείται στον ασθενή με τη μορφή διάγνωσης, και κατόπιν δεν υπάρχει προτυποποιημένη διαδικασία, σχετικά με την καταγραφή ή όχι αυτής της διάγνωσης. Οπότε, η έλλειψη ψηφιοποίησης των ιατρικών δεδομένων δυσχεραίνει τη σωστή καταγραφή του ιστορικού των ασθενών.

Η πρώτη συνάντηση η οποία γίνεται μεταξύ του γιατρού και του συγκεκριμένου ασθενούς, απαιτεί από τη μεριά του γιατρού την καταγραφή κάποιων πληροφοριών σχετικών με τον ασθενή, όπως ημερομηνία γέννησης, φύλο, ονοματεπώνυμο, αλλεργίες, φαρμακευτική αγωγή, ιατρικό – κληρονομικό ιστορικό και άλλα στοιχεία, οι οποίες ονομάζονται Ιατρικός Φάκελος . Ο Ιατρικός Φάκελος είναι η «αποθήκη» όλων των πληροφοριών που αφορούν το ιατρικό ιστορικό του ασθενούς. Αποτελεί επομένως τη βάση της διάγνωσης και της θεραπευτικής αντιμετώπισης του ασθενούς αλλά και τη βάση επιδημιολογικών ερευνών. Επιπλέον παρέχει πληροφορίες διοικητικής και οικονομικής φύσεως, καθώς και ποιοτικού ελέγχου. Σήμερα, οι περισσότερες πληροφορίες που αφορούν έναν ασθενή είναι αποθηκευμένες σε διαφορετικά συστήματα διασκορπισμένα στην ιατρική κοινότητα – γραφεία ιατρών, διαγνωστικές ή ακτινολογικές κλινικές, νοσοκομεία, πολλά εκ των οποίων δεν υποστηρίζουν εγγενώς τη διαλειτουργικότητα. Ένας ιδιώτης ιατρός μπορεί να δυσκολευτεί να αποκτήσει ολοκληρωμένες πληροφορίες σχετικά με κάποιον ασθενή του που νοσηλεύεται σε νοσοκομείο ή να επαναλάβει

ελέγχους και εξετάσεις λόγω ελλιπών πληροφοριών για την πρότερη κατάσταση του ασθενούς. Επιπρόσθετα, προκλήσεις επιβάλλονται και από τις γραφειοκρατικές διαδικασίες, οι οποίες κοστίζουν από άποψη χρόνου αλλά και χρημάτων. Ενδεικτικό είναι το παράδειγμα ενός ασθενούς που αναγκάζεται προσωπικώς να παραδώσει αντίγραφο του ιατρικού του φακέλου σε κάποιον άλλο ιατρό και ο ίδιος ο ασθενής, αν χρειαστεί κάποια δεύτερη γνωμάτευση, να το αποστείλει με φαξ ή αλληλογραφία σε κάποιο άλλο ιατρό του. Συνεπώς, το σημερινό σύστημα, παρουσιάζει έλλειψη μεταφορισιμότητας της πληροφορίας.

Στόχος για μια ιατρική ηλεκτρονική πλατφόρμα, είναι να διευκολυνθεί η διείσδυση της τεχνολογίας και της Κοινωνίας της Πληροφορίας στον τομέα της υγείας. Επιτυχία αυτού του στόχου θα έχει πολύ ευεργετικό αντίκτυπο σε επίπεδο έρευνας λειτουργίας και παροχής υπηρεσιών, αφού θα παρέχεται χρήσιμη στήριξη για πρόσβαση σε ιατρική πληροφορία για κλινική χρήση, έρευνα, εκπαίδευση, καλύτερη εξυπηρέτηση του ασθενούς, διοίκηση και διαχείριση, και τέλος ιατρική φροντίδα. Επομένως, η ηλεκτρονική καταγραφή των ιατρικών δεδομένων προσφέρει μόνο οφέλη στην επιστήμη της υγείας αλλά και στην ίδια την κοινωνία.

Σε μια ιδανική Κοινωνία της Πληροφορίας, ο ασθενής θα είχε πρόσβαση σε τέλειες μηχανές λήψης αποφάσεων, που θα τον ενημερώνουν σχετικά με την ανάγκη ή όχι ιατρικής επίσκεψης, καθώς και το είδος της. Μέχρι τη στιγμή αυτής της επίσκεψης, ο γιατρός θα έχει ήδη λάβει με την αρωγή της τεχνολογίας, όλες τις χρήσιμες πληροφορίες σχετικά με τον ασθενή (ιατρικό φάκελο, συμπτώματα, ασφάλιση) από διάφορες πηγές. Με στόχο, μια ανάλογη υπηρεσία παροχής της ιατρικής πληροφορίας, γίνονται απαραίτητα τα πληροφοριακά συστήματα, τα οποία φυσικά πρέπει να είναι ηλεκτρονικά, φορητά, γρήγορα, εύκολα στη χρήση, συνδεδεμένα τόσο με μια ευρεία ιατρική βάση δεδομένων, όσο και με τα προσωπικά ιατρικά αρχεία του κάθε ασθενούς, και θα πρέπει να είναι στην υπηρεσία τόσο του γιατρού, όσο και του ασθενή. Ο βαθμός ψηφιοποίησης στον τομέα της υγείας μπορεί να μην είναι εξαιρετικά υψηλός, ωστόσο σημαντικές προσπάθειες έχουν λάβει χώρα τόσο σε εφαρμογές έξυπνων τηλεφώνων όσο και σε διαδικτυακές εφαρμογές για τις οποίες υπάρχουν αναφορές στη συνέχεια του κεφαλαίου.

2.3 Αξιοποίηση της τεχνολογίας Blockchain για τη διαχείριση των ιατρικών δεδομένων

2.3.1 Τα πλεονεκτήματα της χρήσης Blockchain έναντι άλλων παραδοσιακών κατακευματισμένων βάσεων δεδομένων

- Ένα από τα βασικότερα οφέλη του Blockchain είναι η αποκεντρωμένη οργάνωση του, καθώς όλα τα κατακευματισμένα συστήματα διαχείρισης βάσεων δεδομένων (Distributed Database Management System - DDBMS) είναι κεντρικά οργανωμένα (δηλαδή οι χρήστες αντιλαμβάνονται ότι λειτουργούν σε μια κεντρική βάση δεδομένων και βασικοί μηχανισμοί αναλαμβάνουν την κατανομή των δεδομένων). Αντίθετα, η τεχνολογία Blockchain διαχειρίζεται μια peer-to peer αποκεντρωμένη βάση δεδομένων (δηλαδή κάθε κόμβος λειτουργεί ανεξάρτητα ακολουθώντας τα καθορισμένα πρωτόκολλα) (McConaghy T, 2016) (S., 2017) (L., 2017). Συνεπώς, το Blockchain είναι κατάλληλο για εφαρμογές στις οποίες υπάρχουν ανεξάρτητοι φορείς (όπως νοσοκομεία, πάροχοι, ασθενείς κλπ.) οι οποίοι συνεργάζονται μεταξύ τους και δεν επιθυμούν να μεταβιβάσουν τον έλεγχο σε κάποιον κεντρικό μεσολαβητή διαχείρισης (Kuo T-T, 2016) (McConaghy T, 2016) (Use of Blockchain in Healthcare and Research Workshop - ONC Tech Lab Innovation - Confluence, 2016).
- Το δεύτερο σημαντικό προνόμιο του Blockchain είναι η αναλλοίωτη διαδρομή ελέγχου. Τα DDBMS υποστηρίζουν λειτουργίες δημιουργίας, διαβάσματος, ενημέρωσης και διαγραφής των δεδομένων όπως όλα τα συστήματα βάσεων. Αντίθετα, το Blockchain υποστηρίζει μόνο λειτουργίες δημιουργίας και διαβάσματος δεδομένων, δηλαδή είναι πολύ δύσκολη η αλλαγή δεδομένων ή αρχείων. (McConaghy T, 2016) Επομένως, το Blockchain είναι κατάλληλο για την αποθήκευση αμετάβλητων κρίσιμων πληροφοριών, όπως είναι οι αιτήσεις αποζημίωσης.
- Ένα ακόμη πλεονέκτημα του Blockchain είναι η έγκυρη προέλευση των δεδομένων. Στα DDBMS, η κατοχή των ψηφιακών στοιχείων(assets) μπορεί να τροποποιηθεί από τον διαχειριστή του συστήματος, ενώ στο Blockchain, η ιδιοκτησία τους μπορεί να αλλάξει μόνο από τον ιδιοκτήτη, ακολουθώντας τα κρυπτογραφικά πρωτόκολλα. Επίσης, η προέλευση των ψηφιακών στοιχείων μπορεί να ανιχνευθεί, καθώς μπορεί να επιβεβαιωθεί ο τρόπος απόκτησης τους μέσω καταχωρήσεων και δεδομένων. Με αυτόν τον τρόπο, αυξάνεται η επαναχρησιμοποίηση επιβεβαιωμένων δεδομένων και άρα και η επίτευξη ασφαλών συναλλαγών. Συνεπώς, το Blockchain

είναι κατάλληλο για τη διαχείριση κρίσιμων ψηφιακών στοιχείων, όπως είναι τα αρχεία συγκατάθεσης ασθενούς.

- Το τέταρτο πλεονέκτημα της τεχνολογίας Blockchain είναι η σταθερότητα, η αξιοπιστία και η διαθεσιμότητα των δεδομένων. Παρόλο που το DDBMS και το Blockchain βασίζονται στην κατακερματισμένη τεχνολογία και άρα υστερούν στο ίδιο σημείο αποτυχίας, για το DDBMS είναι δαπανηρό να αποκτήσει ένα ολοκληρωμένο αντίγραφο ασφαλείας. Αντίθετα, στο Blockchain κάθε κόμβος έχει ολόκληρο αντίγραφο όλων των καταχωρίσεων των δεδομένων (L., 2017). Έτσι, το Blockchain είναι κατάλληλο για τη διατήρηση και τη συνεχή διαθεσιμότητα των καταχωρίσεων (όπως τα ηλεκτρονικά αρχεία υγείας των ασθενών).
- Το τελευταίο αλλά και σημαντικότερο όφελος της χρήσης του Blockchain είναι η βελτιωμένη ασφάλεια και ιδιωτικότητα, χρησιμοποιώντας αλγόριθμους κρυπτογράφησης. Για παράδειγμα το Bitcoin Blockchain χρησιμοποιεί τον 256-bit Secure Hash Algorithm (SHA-256), μια κρυπτογραφική συνάρτηση κατακερματισμού, ορισμένη από τα US Federal Information Processing Standards 186-4, δημοσιευμένο από το National Institute of Standards and Technology (Secure Hash Standard (SHS), 2017). Το SHA-256 χρησιμοποιείται, επίσης, για την παραγωγή διευθύνσεων χρηστών για τη βελτίωση της ανωνυμίας και της ιδιωτικότητας τους (δηλαδή κάθε χρήστης αναπαρίσταται από μια τιμή κατακερματισμού (hash) αντί για την πραγματική του IP διεύθυνση) (SHA-256 - Bitcoin Wiki, χ.χ.). Επιπλέον, το bitcoin Blockchain αξιοποιεί τον αλγόριθμο ψηφιακής υπογραφής, έναν ασύμμετρο αλγόριθμο κρυπτογράφησης, ο οποίος ορίζεται στα US Federal Information Processing Standards για την παραγωγή και επιβεβαίωση υψηλού επιπέδου ασφάλειας των δημόσιων και ιδιωτικών κλειδιών. Έτσι, εξασφαλίζεται η ιδιοκτησία ψηφιακών στοιχείων, όπως είναι και τα αρχεία των ασθενών. (Digital Signature Standard (DSS), 2017) (Bitcoin Wiki. Elliptic Curve Digital Signature, n.d.)

Επομένως, τα ισχυρότερα πλεονεκτήματα για την υιοθέτηση της τεχνολογίας Blockchain σε εφαρμογές του τομέα της υγείας είναι η αποκεντρωμένη οργάνωση, η προέλευση των δεδομένων, η αξιοπιστία και διαθεσιμότητα καθώς και η ασφάλεια/ιδιωτικότητα. (Tsung-Ting Kuo, 2017)

2.3.2 Blockchain Τεχνολογία για Βιοιατρικές εφαρμογές και εφαρμογές φροντίδας υγείας

Καθώς τα οφέλη του Blockchain που περιεγράφηκαν παραπάνω είναι καίριας σημασίας για τις εφαρμογές στον τομέα της υγείας, η φροντίδα υγείας έχει καταστεί μια από τις πιο σημαντικές αναδυόμενες περιοχές εφαρμογών της τεχνολογίας Blockchain κατακερματισμένου καθολικού (distributed ledger technology -

DLT) (Use of Blockchain in Healthcare and Research Workshop - ONC Tech Lab Innovation - Confluence, 2016). Γενικά, το Blockchain εφαρμόζεται ως ένα Distributed Ledger για την αποθήκευση δεδομένων σχετικών με την υγεία και για σκοπούς ανταλλαγής, ανάλυσης, δημοσίευσης, καταγραφής και επικύρωσης τους μεταξύ των ενδιαφερόμενων. (Κυο T-T, 2016)

Μεταξύ των βιοιατρικών εφαρμογών και εφαρμογών υγείας, ένα από τα περιζήτητα θέματα είναι αυτό που σχετίζεται με την αξιοποίηση του Blockchain ως μια βασική υποδομή για την Ανταλλαγή Πληροφοριών Υγείας, ή συναλλαγών υγείας μεταξύ των ασθενών, των παρόχων, των πληρωτών και άλλων σχετικών υπηρεσιών (Κυο T-T, 2016). Αυτές οι εφαρμογές μπορούν να κατηγοριοποιηθούν περαιτέρω ανάλογα με τους σκοπούς για τους οποίους θα αξιοποιηθούν το Blockchain και τα αποθηκευμένα δεδομένα. Οι τελευταίες θα περιγραφούν στις επόμενες υποενότητες. Για την κάθε κατηγορία εφαρμογών, θα αναλυθούν οι περιπτώσεις χρήσεις και τα κύρια οφέλη της τεχνολογίας Blockchain. (Tsong-Ting Kuo, 2017)

- **Βελτιωμένη διαχείριση ιατρικών αρχείων**

Πολλές έρευνες και προγράμματα που βρίσκονται σε εξέλιξη επικεντρώνονται στα δεδομένα φροντίδας του ασθενή χρησιμοποιώντας Blockchains για να βελτιώσουν τη διαχείριση ιατρικών καταχωρήσεων (Gareth, 2016) (Matthias, 2016). Σε αυτές περιλαμβάνονται και οι εξής έρευνες: Healthcare Data Gateways, (Yue X, 2016) MedVault, (Blough D, 2008) Fatcom, (Snow P, 2015) BitHealth, (Sarwal A, n.d.) Gem Health Network (Health | Gem, n.d.) όπως επίσης και διάφορες άλλες. Επιπλέον, αρκετές είναι και οι εταιρείες, όπως η Deloitte (Krawiec R, 2016) και η Accenture, (C B, 2016) οι οποίες χρησιμοποιούν την τεχνολογία Blockchain τόσο για την αποθήκευση δεδομένων ιατρικής φροντίδας όσο και για τη διαχείριση ιατρικών καταχωρήσεων. Ένα άλλο φημισμένο παράδειγμα είναι η Guardtime (Guardtime Industrial Blockchain, n.d.), μια εταιρεία που παρέχει ένα σύστημα βασισμένο σε Blockchain στην Εσθονία για να εξασφαλίσει 1 εκατομμύριο ιατρικά αρχεία. Τα οφέλη και οι περιπτώσεις χρήσης υιοθέτησης του Blockchain για τη βελτίωση της διαχείρισης ιατρικών αρχείων συνοψίζονται στον παρακάτω πίνακα.

Blockchain: Πλεονεκτήματα	Περίπτωση Χρήσης: Βελτιωμένη Διαχείριση Ιατρικών Αρχείων
Αποκεντρωμένη Διαχείριση	<u>Τα αρχεία σχετικά με την ιατρική περίθαλψη που διαχειρίζονται οι ασθενείς:</u> «Ο ασθενής γίνεται η πλατφόρμα, κατέχονται και ελέγχοντας την πρόσβαση στα δεδομένα υγείας του. Αυτό εξαλείφει όλα τα εμπόδια που έχουν οι ασθενείς για να αποκτήσουν αντίγραφα των ιατρικών τους δεδομένων ή να μεταφέρουν αυτά σε άλλο πάροχο υγείας.» (D, 2016)
Αναλλοίωτη διαδρομή ελέγχου	<u>Μη αναστρέψιμα αρχεία ασθενών:</u> «Τα δεδομένα αποθηκεύονται σε ιδιωτικό Blockchain cloud. Το Blockchain μπορεί να εγγυηθεί ότι τα ιατρικά δεδομένα δεν

	μπορούν να τροποποιηθούν από κανένα, συμπεριλαμβανομένων των ιατρών και των ίδιων των ασθενών» (Yue X, 2016)
Προέλευση Δεδομένων	<u>Πηγή επαληθεύσιμων ιατρικών αρχείων:</u> «Τα αρχεία υπογράφονται από την πηγή προέλευσης τους, επιτρέποντας την επαλήθευση της εγκυρότητας των εγγράφων (και αντίστοιχα της ακυρότητας των ψευδών αρχείων που πρέπει να απορριφθούν» (Blough D, 2008)
Αξιοπιστία/ Διαθεσιμότητα	<u>Μειωμένος κίνδυνος καταγραφής των ασθενών:</u> «Επειδή τα δεδομένα αποθηκεύονται σε ένα αποκεντρωμένο δίκτυο, δεν υπάρχει ενιαίος οργανισμός που να μπορεί να κλαπεί ή να αποκτηθεί παράνομη πρόσβαση, ώστε να υποκλαπεί μεγάλος αριθμός αρχείων των ασθενών.» (D, 2016)
Ασφάλεια/ Ιδιωτικότητα	<u>Αυξημένη ασφάλεια ιατρικών αρχείων:</u> «Τα δεδομένα κρυπτογραφούνται στο Blockchain και μπορούν να αποκρυπτογραφηθούν μόνο με το ιδιωτικό κλειδί του ασθενή. Ακόμη και εάν στο δίκτυο έχει διεισδύσει κάποιο κακόβουλο λογισμικό, δεν υπάρχει κανένας πρακτικός τρόπος για να διαβασθούν τα δεδομένα των ασθενών.» (D, 2016)

- **Βελτιωμένη διαδικασία αιτήσεων αποζημίωσης**

Ένας άλλος σημαντικός λόγος δημιουργίας εφαρμογών είναι η επαλήθευση των συναλλαγών αίτησης για την υποστήριξη των χρηματοδοτήσεων ιατρικής περίθαλψης, όπως είναι η δέσμευση του ποσού πληρωμής (Attili S, 2016) και οι εναλλακτικοί τρόποι πληρωμής (Κ Υ. , 2016). Τα πλεονεκτήματα και οι περιπτώσεις χρήσης του Blockchain για να ενισχυθεί η διαδικασία αιτήσεων αποζημίωσης συνοψίζονται στον παρακάτω πίνακα.

Blockchain: Πλεονεκτήματα	Περίπτωση Χρήσης: Ενίσχυση της διαδικασίας αιτήσεων αποζημίωσης
Αποκεντρωμένη Διαχείριση	<u>Πραγματικού χρόνου διαδικασία αίτησης:</u> «Η ικανότητα να απομακρύνει τους διαμεσολαβητές από αυτή τη διαδικασία, είναι και αυτή που ξεχωρίζει το Blockchain από κάθε άλλη τεχνολογία. Αυτή η δυνατότητα θα επιτρέψει τη διευκόλυνση της εκδίκασης αιτήσεων πραγματικού χρόνου, αντικαθιστώντας τη διαμεσολάβηση του σχεδίου ασφάλισης με τις διαφανείς τεχνολογίες Blockchain.» (Κ C. , 2016)
Αναλλοίωτη διαδρομή ελέγχου	<u>Βελτιωμένος έλεγχος αιτήσεων και ανίχνευσης απάτης:</u> «Οι πληρωτές, οι ιδιωτικοί και κρατικοί φορείς ασφάλισης και οι μεμονωμένοι πληρωτές ωφελούνται από τη διευκόλυνση των ελέγχων και την καλύτερη ανίχνευση απάτης (το οποίο θεμελιώνεται από τη σταθερότητα και αμεταβλητότητα του Blockchain).» (Attili S, 2016)

Προέλευση Δεδομένων	<u>Επαληθεύσιμα αρχεία για την πιστοποίηση των αιτήσεων:</u> «Το κυριότερο εμπόδιο της διαδικασίας πιστοποίησης αιτήσεων είναι η κατανεμημένη φύση πολλών αρχείων που τροφοδοτούν τις αποφάσεις των ασθενών, των παρόχων και των διαχειριστών. Οι απαιτήσεις για την έγκριση μιας αίτησης είναι και τα εμπόδια που μπορεί να επιλύσει ένα κατανεμημένο Blockchain σύστημα.» (Vian K, 2016)
Αξιοπιστία/ Διαθεσιμότητα	<u>Βελτιωμένη προσβασιμότητα των δεδομένων των ασθενών:</u> «Ο πάροχος και οι υπηρεσίες σχετικές με την υγεία και τα ιατρικά προϊόντα ωφελούνται από την εύκολη πρόσβαση των δεδομένων των ασθενών [η οποία βασίζεται στο εικονικό βιβλιάριο Blockchain].» (Attili S, 2016)
Ασφάλεια/ Ιδιωτικότητα	<u>Αύξηση της ασφάλειας των πληροφοριών ιατρικής ασφάλισης των ασθενών:</u> «Ο ασθενής ή το μέλος ωφελούνται από τη μειωμένη πιθανότητα κλοπής των στοιχείων πληρωμής [το οποίο βασίζεται στο μηχανισμό του Blockchain].» (Attili S, 2016)

- **Επιτάχυνση της κλινικής και βιοιατρικής έρευνας**

Αρκετοί ερευνητές επίσης προτείνουν την επιτάχυνση της επαναχρησιμοποίησης των κλινικών δεδομένων, δηλαδή κλινικών και βιοιατρικών μελετών και ερευνών, χρησιμοποιώντας την τεχνολογία Blockchain. Τέτοιου είδους παραδείγματα είναι το MedRec, (Ekblaw A, 2016) (Azaria A, 2016) Data Lake, (Linn LA, 2016) Healthbank, (Matthias, 2016) (Healthbank.coop, n.d.) και τα δίκτυα κοινής χρήσης δεδομένων που βασίζονται στο Blockchain (Peterson K, 2016). Ακόμη, το ModelChain υιοθέτησε το Blockchain για να αυξήσει την ασφάλεια, την αξιοπιστία και τη σταθερότητα του μοντέλου προβλέψεων των κατανεμημένων ιατρικών προσωπικών δεδομένων σε σχέση με άλλους οργανισμούς (Kuo T-T, 2016). Τα οφέλη και οι περιπτώσεις χρήσης της υιοθέτησης του Blockchain για να επιταχύνει την κλινική και βιοιατρική έρευνα συνοψίζονται στον παρακάτω πίνακα.

Blockchain: Πλεονεκτήματα	Περίπτωση Χρήσης: Επιτάχυνση της κλινικής και βιοιατρικής έρευνας
Αποκεντρωμένη Διαχείριση	<u>Βελτιωμένη ανάλυση και ανταλλαγή των δεδομένων ιατρικής περίθαλψης χωρίς την εκχώρηση του ελέγχου:</u> «Το Blockchain έχει εξ ορισμού μια αποκεντρωμένη (δηλαδή μια peer-to-peer, χωρίς ενδιάμεσο) αρχιτεκτονική. Κάθε οργανισμός μπορεί να έχει τον πλήρη έλεγχο των υπολογιστικών πόρων του [ενώ συνεργάζεται με άλλους οργανισμούς ή ιδρύματα για την ανταλλαγή και την ανάλυση δεδομένων].» (Kuo T-T, 2016)

<p>Αναλλοίωτη διαδρομή ελέγχου</p>	<p><u>Ανιχνεύσιμα δεδομένα παραγόμενα από τον ασθενή:</u> «Χρησιμοποιώντας το Blockchain, αυτό το μοντέλο θα μπορούσε να εξατομικευθεί ακόμη περισσότερο με τέτοιο τρόπο, ώστε τα προσωπικά δεδομένα που παράγονται από τους ασθενείς και είναι διαθέσιμα στους ερευνητές να μπορούν να καταγραφούν στην ερευνητική διαδικασία με μια χρονική σήμανση.» (Ekblaw A, 2016) (Healthbank.coop, χ.χ.)</p>
<p>Προέλευση Δεδομένων</p>	<p><u>Αποδεικνυόμενη προέλευση των ιατρικών ερευνητικών δεδομένων:</u> «Στο MedRec, το οποίο είναι πρωτότυπο με βάση το Blockchain για τα ηλεκτρονικά ιατρικά αρχεία και τα δεδομένα ιατρικών ερευνών, έχουν εφαρμοστεί καθοριστικές ιδιότητες για τις πηγές προέλευσης.» (Ekblaw A, 2016) (Azaria A, 2016)</p>
<p>Αξιοπιστία/ Διαθεσιμότητα</p>	<p><u>Εξαιρετική διαθεσιμότητα δεδομένων ιατρικής περίθαλψης:</u> «Το Blockchain μπορεί να εξασφαλίσει μια συνεχή διαθεσιμότητα και πρόσβαση σε δεδομένα σε πραγματικό χρόνο. Η πρόσβαση σε δεδομένα πραγματικού χρόνου θα βελτιώσει τον συντονισμό της κλινικής περίθαλψης και θα βελτιώσει την κλινική φροντίδα σε καταστάσεις έκτακτης ανάγκης. Τα δεδομένα σε πραγματικό χρόνο θα επιτρέψουν, επίσης, στους ερευνητές και στους πόρους δημόσιας υγείας να εντοπίσουν, να απομονώσουν και να οδηγηθούν σε αλλαγές σχετικά την επιρροή των περιβαλλοντολογικών συνθηκών στη δημόσια υγεία. Όπως για παράδειγμα, οι επιδημίες θα μπορούσαν να ανιχνευθούν νωρίτερα, ώστε να προστατευθεί η δημόσια υγεία.» (Linn LA, 2016)</p>
<p>Ασφάλεια/ Ιδιωτικότητα</p>	<p><u>Ασφαλή και προστατευμένα ιατρικά προσωπικά δεδομένα:</u> « Η αξιοποίηση του Blockchain στον τομέα της υγείας έχει τη δύναμη να ενθαρρύνει εκατομμύρια ανθρώπων, παρόχους ιατρικής περίθαλψης, φορείς υγείας και ιατρικούς ερευνητές να μοιραστούν πληθώρα γενετικών, διατροφικών, περιβαλλοντικών δεδομένων και δεδομένων υγείας, καθώς εγγυάται την προστασία των προσωπικών δεδομένων.» (Linn LA, 2016)</p>

- **Βελτίωση της αποθήκευσης των βιοιατρικών δεδομένων και δεδομένων σχετικών με τη φροντίδα υγείας**

Εκτός από την αξιοποίηση του Blockchain ως βιβλιάριο δεδομένων σχετικά με την περίθαλψη των ασθενών, πληθώρα ερευνών έχουν επίσης προτείνει τη χρήση του για την αποθήκευση διάφορων τύπων φροντίδας υγείας ή σχετικών δεδομένων. Τέτοια δεδομένα μπορεί να είναι γονιδιωματικά και ιατρικής ακρίβειας δεδομένα, δεδομένα σχετικά με τον ασθενή, κατάλογοι παρόχων και ασθενών, δεδομένα προγραμμάτων ιατρικής αγωγής,

δεδομένα κλινικών δοκιμών, έγγραφα συγκατάθεσης ασθενούς, δεδομένα της φαρμακευτικής εφοδιαστικής αλυσίδας και δεδομένα βιοδεικτών. Τα πλεονεκτήματα και οι περιπτώσεις χρήσης του Blockchain σε ιατρικά δεδομένα συνοψίζονται στον παρακάτω πίνακα.

Blockchain: Πλεονεκτήματα	Περίπτωση Χρήσης: Προηγμένο βιβλιάριο βιοιατρικών δεδομένων και δεδομένων σχετικών με τη φροντίδα υγείας
Αποκεντρωμένη Διαχείριση	<u>Το θεμέλιο των αποκεντρωμένων δεδομένων υγείας:</u> « Το Blockchain πρόκειται να γίνει το θεμέλιο της ψηφιακής υγείας. Ενσωματώνοντας δεδομένα από τεχνολογίες βασισμένες στον ασθενή αλλά και από το EMR, θα δημιουργείται ένας προσβάσιμος κοινός χώρος μεταξύ των εξουσιοδοτημένων χρηστών, όπως οι πάροχοι και οι ασθενείς. Όλα τα δεδομένα θα αποθηκεύονται με αποκεντρωμένο τρόπο χωρίς να υπάρχει ενιαία οντότητα, η οποία θα αποθηκεύει ή θα έχει εξουσιοδοτημένη πρόσβαση.» (Use of Blockchain in Healthcare and Research Workshop - ONC Tech Lab Innovation - Confluence, 2016)
Αναλλοίωτη διαδρομή ελέγχου	<u>Αντικατάσταση του πρωτοκόλλου κλινικών ερευνών:</u> «Η χρήση τεχνολογίας Blockchain έχει πρόσφατα αποδειχθεί ότι παρέχει ένα αμετάβλητο κατάλογο από κάθε βήμα του πρωτοκόλλου κλινικής έρευνας. Αυτό θα μπορούσε εύκολα να προσαρμοστεί στη βασική και πειραματική επιστήμη μοντέλων. Όλοι οι συμμετέχοντες στο ερευνητικό δίκτυο peer-to-peer έχουν πρόσβαση σε όλα τα συνεχώς ενημερωμένα δεδομένα. Είναι ουσιαστική και απαραβίαστη απόδειξη ότι οποιαδήποτε αλλαγή, όπως για την προκαθορισμένη ανάλυση δεδομένων, θα πρέπει να γίνει σε κάθε υπολογιστή(συνήθως χιλιάδες) μέσα στο κατακεκομμένο δίκτυο.» (EJ, 2016)
Προέλευση Δεδομένων	<u>Η εξασφάλιση της μεταβίβασης της ιδιοκτησίας από τον αρχικό κατασκευαστή στην εφοδιαστική αλυσίδα φαρμάκων:</u> «Χρησιμοποιώντας το Blockchain, μπορεί να εντοπιστεί η προέλευση του [ιατρικού] προϊόντος και των συστατικών του και η κάθε μια μεταφορά ιδιοκτησίας, εφόσον καθίσταται σαφής και διαθέσιμη για όλους. Τα πλαστά, κακής ποιότητας ή κλεμμένα προϊόντα μπορούν να εντοπιστούν και να ταυτοποιηθούν.» (Matthias, 2016) (P, n.d.)
Αξιοπιστία/ Διαθεσιμότητα	<u>Βελτιωμένη αξιοπιστία για συστήματα πρόληψης / ανίχνευσης παραποιημένων φαρμάκων στην αλυσίδα εφοδιασμού φαρμάκων:</u> «Στις υπάρχουσες λύσεις υπάρχει ακόμα μια κεντρική αρχή που μπορεί να παραβιαστεί με αποτέλεσμα τα έγγραφα της να κινδυνεύουν να παραποιηθούν. Ένα οι υφιστάμενες λύσεις τροποποιηθούν, ώστε να αξιοποιήσουν το Blockchain, τότε το σύστημα προμηθειών και διανομής θα μπορεί να

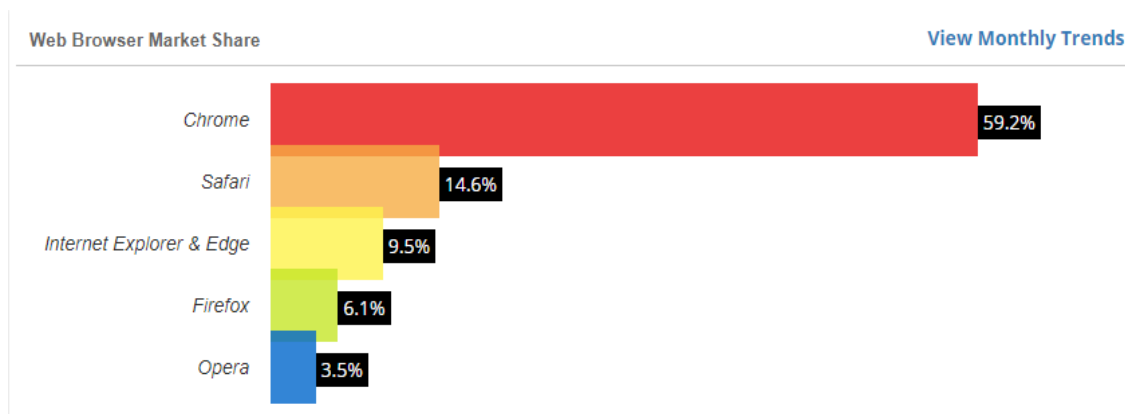
	καταστήσει την απομίμηση φαρμάκων ως λήξαν θέμα.» (Attili S, 2016)
Ασφάλεια/ Ιδιωτικότητα	<u>Υψηλότερη εμπιστοσύνη των ασθενών στα συστήματα καταγραφής συγκατάθεσης:</u> «Οι ασθενείς μπορούν να προσθέσουν δηλώσεις συγκατάθεσης σε οποιοδήποτε στάδιο της ιατρική περίθαλψης τους- με την πεποίθηση ότι το Blockchain θα τα διατηρήσει ασφαλή.» (C B, 2016)

Κεφάλαιο 3. Τεχνολογικό Υπόβαθρο

3.1 Τεχνολογίες Πληροφορικής και Επικοινωνιών

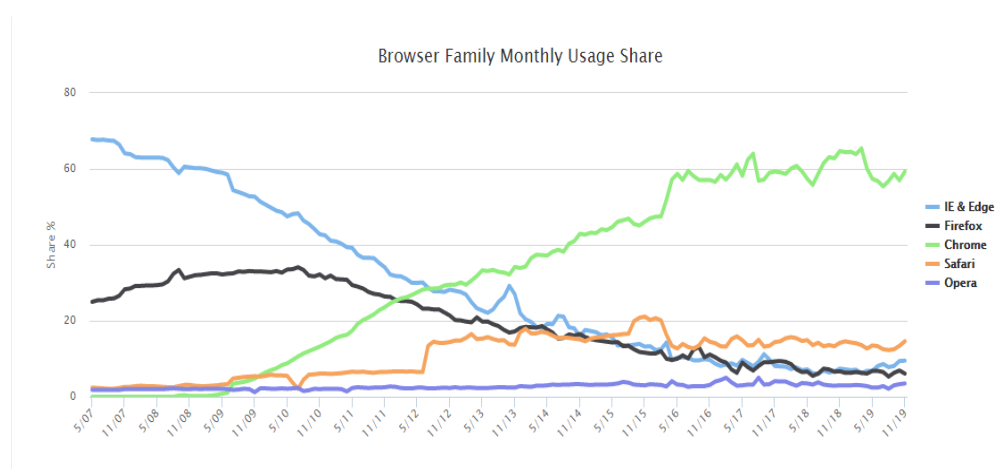
3.1.1 Φυλλομετρητές ιστού (Web browsers)

Σύμφωνα με στατιστικά χρήσης γνωστών browsers που αντλήθηκαν από τον ιστότοπο www.w3counter.com και αφορούν την πιο πρόσφατη καταχώρησή τους τη στιγμή της συγγραφής (Νοέμβριος 2019), οι πιο διαδεδομένοι browsers είναι κατά σειρά ο Chrome, ο Safari, ο Internet Explorer, ο Firefox και ο Opera. Τα ποσοστά τους παρουσιάζονται στην κάτωθι εικόνα (Εικόνα 1).



Εικόνα 1. Μεριδίο αγοράς browsers - Νοέμβριος 2019 (εικόνα από (W3Counter: Free Web Stats and Website Widgets, n.d.))

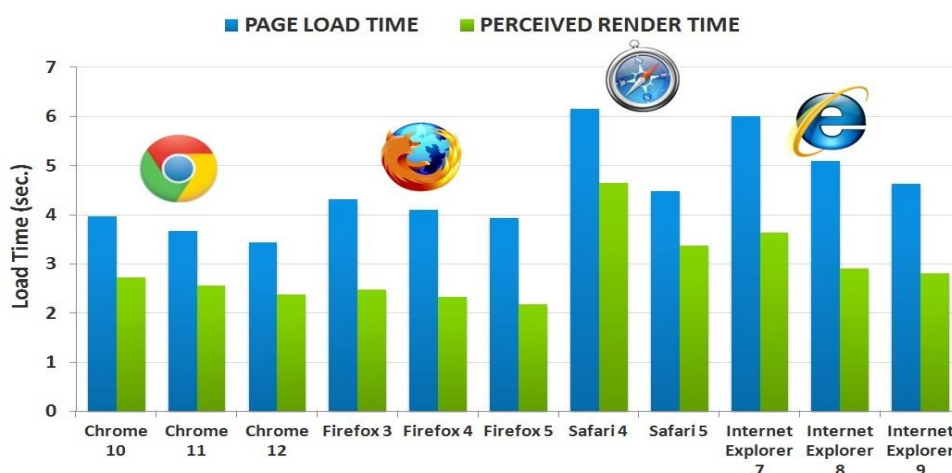
Κάνοντας μια μικρή αναδρομή των τελευταίων οκτώ ετών (Εικόνα 2), παρατηρείται ότι στα μέσα του 2007 δύο στους τρεις χρήστες επέλεξαν το browser της Microsoft για την πλοήγησή τους στο Internet. Ορόσημο αποτέλεσε η είσοδος της Google στην εν λόγω αγορά με τον Chrome (Σεπτέμβριος 2008). Από τότε και σταδιακά το μερίδιο του IE άρχισε να συρρικνώνεται ενώ το αντίστοιχο του Chrome να αυξάνει διαρκώς κερδίζοντας χρήστες από όλους τους ανταγωνιστές, ώσπου έφτασε σήμερα να είναι ο αδιαφιλονίκητος κυρίαρχος της αγοράς.



Εικόνα 2. Ιστορία μεριδίου χρήσης browsers (εικόνα από (W3Counter: Free Web Stats and Website Widgets, n.d.))

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, όπως φαίνεται και στην Εικόνα 3.

Performance Differences Across Browsers



Εικόνα 3. Απόδοση browsers

Λαμβάνοντας υπόψιν τα παραπάνω, αλλά και λόγω της προσωπικής προτίμησης του υποφαινόμενου προς το φυλλομετρητή της Google, επιλέχθηκε ο Chrome ως περιβάλλον εξέλιξης της εφαρμογής (εγκατεστημένος πάνω σε λειτουργικό σύστημα Ubuntu Linux). Αυτό βέβαια δεν αποκλείει τη δυνατότητα εκτέλεσής της και στους υπόλοιπους – μικρές αλλαγές ίσως απαιτούνται σε κάποιες περιπτώσεις κυρίως για λόγους συμβατότητας.

3.1.2 Ανάπτυξη web applications

Μία web-based εφαρμογή είναι στην ουσία μία ιστοσελίδα που στοχεύει στην παροχή πλουσιότερης και πιο περίπλοκης λειτουργικότητας από την απλή προβολή κειμένου και εικόνων. Κατά τη βάση του, μια ιστοσελίδα μπορεί να υλοποιηθεί με χρήση απλώς HTML (Hyper Text Markup Language) (HTML - Wikipedia, the free encyclopedia, n.d.). Η γλώσσα HTML γράφεται με τη μορφή στοιχείων (HTML elements) αποτελούμενα από tags, περικλειόμενα από τους χαρακτήρες '<', '>', όπως φαίνεται και στην Εικόνα 4.

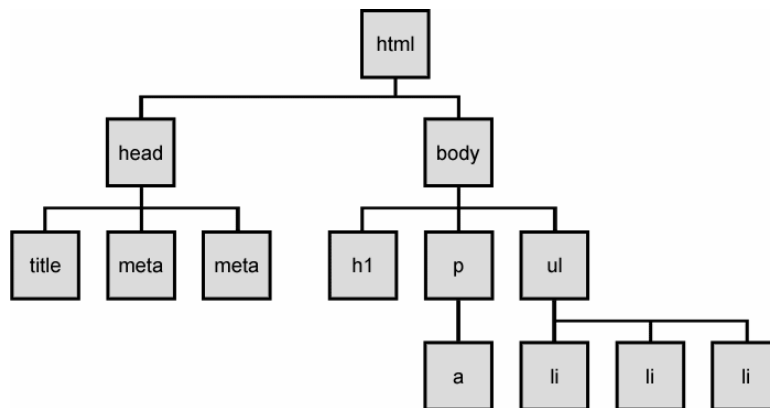
```

<html>
  <head>
    <title>A Basic Webpage</title>
  </head>
  <body>
    <h1>My Heading</h1>
    <p>My paragraph.</p>
  </body>
</html>

```

Εικόνα 4. Γραφική μορφή ενός απλού HTML κειμένου

Ο κώδικας μετατρέπεται από το browser σε μια δενδρική δομή αποτελούμενη από JavaScript αντικείμενα-κόμβους, ή όπως αλλιώς ονομάζεται DOM (Document Object Model) , όπως φαίνεται και στην Εικόνα 5.

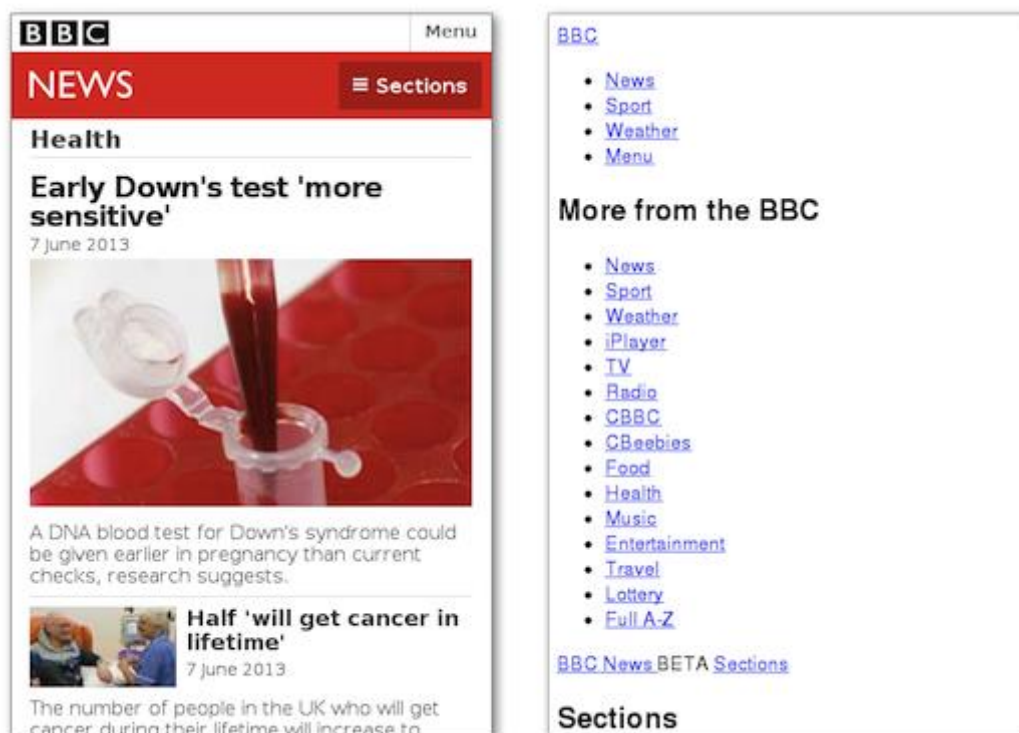


Εικόνα 5. Document Object Model μιας ιστοσελίδας

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. (React (JavaScript library) - Wikipedia, χ.χ.) 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.

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) (Hunt, n.d.). Υπεύθυνες για τα τρία αυτά διαφορετικά στοιχεία είναι οι τεχνολογίες της HTML, CSS και JavaScript, αντίστοιχα, οι οποίες από κοινού συνεισφέρουν στη διανομή πλούσιου περιεχομένου σελίδων ιστού. Η γλώσσα CSS (Cascading Style Sheet) δημιουργήθηκε και προτάθηκε από την κοινοπραξία W3C (World Wide Web Consortium) (World Wide Web Consortium (W3C), n.d.), τον κύριο διεθνή οργανισμό προτύπων για το Web, το 1996. Γενικά

χρησιμοποιείται για να παραμετροποιήσει την εμφάνιση του HTML περιεχομένου, προσθέτοντας γνωρίσματα όπως στυλ γραμματοσειράς, χρώμα, διαστάσεις κτλ. (βλ. Εικόνα 6). 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.



Εικόνα 6. Άποψη ιστοσελίδας με χρήση και χωρίς χρήση CSS

Επιπρόσθετα, όπως η μορφοποίηση που παρέχει η CSS μπορεί να συμπεριληφθεί στο ίδιο αρχείο HTML με τη χρήση του tag `<style>`, υπάρχει μέριμνα και για τη JavaScript με το αντίστοιχο tag `<script>`. Η τελευταία χρησιμοποιείται για την προσθήκη λειτουργικότητας σε μια ιστοσελίδα. Διαχειρίζεται διαδραστικά στοιχεία – όπως οι φόρμες κειμένου και τα κουμπιά, εκτελεί υπολογισμούς, αναλαμβάνει την επικοινωνία με τον εξυπηρετητή και γενικά ο,τιδήποτε μπορεί να απαιτεί ένας προγραμματιστής από μια κλασική γλώσσα προγραμματισμού. Όταν μια ιστοσελίδα περιέχει JavaScript – και σχεδόν όλες σήμερα περιέχουν, ο κώδικας λαμβάνεται από τον εξυπηρετητή στο μηχάνημα του χρήστη (client), όπου μεταφράζεται και εκτελείται από το browser. Με τη σημερινή αυξανόμενη υπολογιστική ισχύ των συσκευών αλλά και τη διαρκή βελτίωση των μηχανών επιφορτισμένων με την εκτέλεση JavaScript (JavaScript engines⁷) που αποτελούν τον πυρήνα των σύγχρονων browsers, πολύπλοκος και ογκώδης κώδικας εκτελείται τοπικά βασιζόμενος στην ταχύτητα του εγκατεστημένου υλισμικού και όχι του δικτύου. Από άποψη οργάνωσης, είναι δυνατό να συμπεριληφθεί σε ένα μόνο HTML αρχείο ο κώδικας όλων των παραπάνω γλωσσών (με τη χρήση των κατάλληλων tags), αλλά γενικά κάτι τέτοιο αποθαρρύνεται γιατί απουσιάζει η αυτονομία των επιμέρους

συστατικών στοιχείων (modularity). Επίσης, ο παραγόμενος κώδικας είναι ιδιαίτερα δύσκολο να ελεγχθεί και να αποσφαλματωθεί. Αντί αυτού, προτιμάται η λύση των πολλών διαφορετικών αρχείων, όπου το καθένα αυστηρά θα περιέχει κώδικα μιας γλώσσας. Η σπονδυλωτή (modular) προσέγγιση αυτή έχει τα οφέλη της ευκολότερης επεκτασιμότητας, αναγνωσιμότητας και συντηρησιμότητας. Για το σκοπό αυτό έχουν επίσης αναπτυχθεί ειδικά εργαλεία και μέθοδοι που διευκολύνουν ακόμα περισσότερο το έργο του προγραμματιστή – κάποια από αυτά χρησιμοποιήθηκαν κατά την εξέλιξη της παρουσιαζόμενης εφαρμογής και θα αναλυθούν στη συνέχεια.

3.1.3 Συμβατότητα web browsers

Ο προγραμματισμός για την πλατφόρμα των browsers, παρ' όλα αυτά είναι απαιτητικός από την άποψη ότι διάφορες πλευρές της λειτουργικότητας που παρέχουν οι τεχνολογίες του διαδικτύου – HTML, CSS, JavaScript, δεν έχουν υλοποιηθεί ομοιόμορφα από όλους τους browser που εξετάστηκαν στην αρχή του κεφαλαίου. Αυτό απορρέει από το γεγονός ότι οι τελευταίοι εξελίσσονται και συντηρούνται από διαφορετικές εταιρίες με διαφορετικούς επιχειρησιακούς στόχους και στρατηγικές υλοποίησης. Κατά τα τέλη της δεκαετίας του '90 αυτό αποδείχτηκε καταστροφικό, καθώς οι χρήστες έφτασαν στο σημείο να έχουν να επιλέξουν μεταξύ browsers με διαφορετικά χαρακτηριστικά χωρίς κανένα συντονισμό μεταξύ των ατόμων που τους παρείχαν. Ωστόσο, με τη δημιουργία του World Wide Web Consortium, τα πράγματα άλλαξαν προς το καλύτερο. Η οργάνωση άρχισε να κινείται προς την κατεύθυνση της προτυποποίησης των γλωσσών και των τεχνολογιών που χρησιμοποιούνται για την παροχή διαδικτυακού περιεχομένου. Αυτό συνέβη σε αρκετά στάδια και αρκετές επανεκδόσεις των προτύπων απαιτήθηκαν, ώστε να φτάσει σήμερα η εικόνα του διαδικτύου να μοιάζει πιο ενοποιημένη από ποτέ.

Εντούτοις, εξακολουθούν να υπάρχουν ακόμα ορισμένες μικρές διαφοροποιήσεις, το οποίο σε κάθε περίπτωση κρίνεται δικαιολογημένο, αφού διαφορετικές εταιρίες συντηρούν διαφορετικούς browsers. Όλα τα χαρακτηριστικά που προσφέρει η CSS δεν είναι καθολικώς υποστηριζόμενα ή τουλάχιστον η υποστήριξη σε ορισμένους browsers υπολείπεται έναντι άλλων. Το ευτύχημα είναι πως τα πιο συνηθισμένα και ευρέως χρησιμοποιούμενα στοιχεία τείνουν να υποστηρίζονται από όλους. Ο ιστότοπος www.caniuse.com παρέχει στους ενδιαφερόμενους, προγραμματιστές και μη, εκτενή καταγραφή των υποστηριζόμενων χαρακτηριστικών ανά έκδοση8 και οικογένεια browser. Ακόμα, ένα ζήτημα που όλοι όσοι εξελίσσουν λογισμικό για browser πρέπει να γνωρίζουν είναι πως οι διερμηνείς JavaScript μπορεί να διαφέρουν στην υλοποίησή τους.

Για την αντιμετώπιση τέτοιων ασυμβατοτήτων, υπάρχει μια μέθοδος γνωστή ως “browser sniffing” που μπορεί να εφαρμοστεί σε τέτοιες περιπτώσεις, κατά την οποία εξακριβώνεται ο τύπος και η έκδοση του browser και ο κώδικας που ακολουθεί διαμορφώνεται ανάλογα. Ωστόσο, ο W3C αποδοκιμάζει την πρακτική αυτή, καθώς υπάρχουν πάρα πολλές διαφορετικές εκδοχές για τις οποίες ο

προγραμματιστής πρέπει να ανταποκριθεί ξεχωριστά, οδηγούμενος έτσι να συντάξει πολλαπλάσιες γραμμές κώδικα για την εκτέλεση εργασιών κατά τα άλλα τετριμμένων. Αντί γι' αυτή, προτείνει τη λεγόμενη ανίχνευση χαρακτηριστικών, η οποία επιτρέπει στον προγραμματιστή να «ρωτήσει» το εκάστοτε περιβάλλον εκτέλεσης – το browser δηλαδή – αν υποστηρίζει κάποιο συγκεκριμένο χαρακτηριστικό το οποίο σκοπεύει να χρησιμοποιήσει. Η λύση αυτή είναι σίγουρα πιο κομψή και ελκυστική, όπως επίσης και λιγότερο επιρρεπής σε λάθη.

3.1.4 Javascript

Για ολόκληρο το front-end της εφαρμογής χρησιμοποιήθηκε η JavaScript, 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.

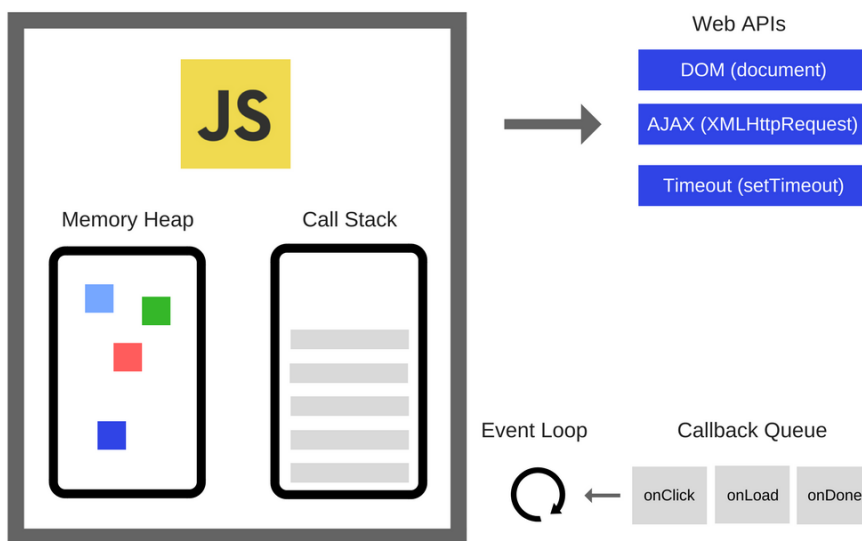
3.1.4.1 Δυναμικό Σύστημα Τύπων και Ταυτοχρονισμός

Η γλώσσα JavaScript είναι μονονηματική (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.

3.1.4.2 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 επανάκλησης (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 (βλ. Εικόνα 7).



Εικόνα 7. Βρόχος συμβάντων της Javascript

3.1.4.3 Run-to-completion logic

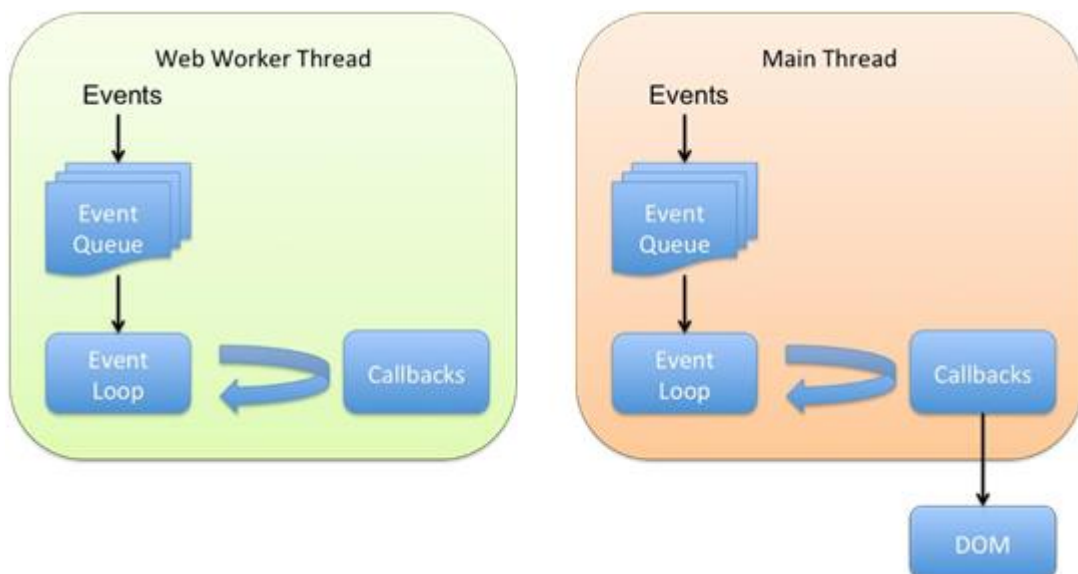
Η πρακτική που ακολουθεί το περιβάλλον της JavaScript είναι η πλήρης επεξεργασία κάθε μηνύματος προτού συνεχίσει με το επόμενο. Αυτό προσφέρει κάποιες ελκυστικές ιδιότητες κατά το σχεδιασμό της λογικής των προγραμμάτων, συμπεριλαμβανομένης της εγγύησης της εγκυρότητας των δεδομένων κατά τη διάρκεια της 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 απαιτούσαν 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 `set-Timeout()` (`WindowTimers.setTimeout()` - Web API Interfaces, n.d.) και `setInterval()` (`WindowTimers.setInterval()` - Web API Interfaces | MDN, n.d.).

3.1.4.4 Web Workers

Μεγάλη πρόοδο επιτεύχθηκε στο συγκεκριμένο τομέα με την έλευση των Web Workers. Σύμφωνα με τις επίσημες προδιαγραφές, οι browsers πλέον καλούνται να υλοποιήσουν ένα API για την παρασκηνιακή εκτέλεση scripts από τις εφαρμογές ιστού. Με τον τρόπο αυτό, υπολογιστικά απαιτητικές εργασίες μπορούν να μεταφερθούν σε δικό τους ανεξάρτητο νήμα εκτέλεσης με το δικό του ανεξάρτητο βρόχο συμβάντων, χωρίς να υπάρχει το ενδεχόμενο να «μπλοκάρουν» τη διεπαφή της εκάστοτε εφαρμογής με το χρήστη, η οποία «τρέχει» στο κύριο νήμα (βλ. Εικόνα 8).



Εικόνα 8. Web worker και κύριο νήμα εκτέλεσης

Η επικοινωνία μεταξύ του κυρίου νήματος (main or UI thread) και των Web workers λαμβάνει χώρα μέσω ενός μοντέλου συμβάντων (event model) και της μεθόδου `postMessage()`. Ανάλογα με τον τύπο και την έκδοση του browser καθώς και τον τρόπο κλήσης της μεθόδου, η τελευταία μπορεί να λειτουργήσει με δύο τρόπους. Οι περισσότεροι browsers υλοποιούν το λεγόμενο structured cloning αλγόριθμο, ο οποίος επιτρέπει το πέρασμα πολύπλοκων τύπων δεδομένων από και προς τους workers, όπως File, Blob, ArrayBuffer και JSON αντικείμενα. Ωστόσο, τα αντικείμενα αυτά αντιγράφονται πριν το πέρασμά τους (κάτι αντίστοιχο με το πέρασμα κατά τιμή στη C), γεγονός που εισάγει καθυστέρηση στην όλη διαδικασία. Για το λόγο αυτό, αναπτύχθηκε μια δεύτερη τεχνική που αξιοποιεί τα λεγόμενα transferrable

objects. Χρησιμοποιώντας την, δεδομένα μεταφέρονται από το περιβάλλον του κυρίου νήματος σε αυτό του παρασκηνακού χωρίς να αντιγράφεται τίποτα με σημαντικό αντίκτυπο στην επίδοση. Ομοιάζει με το πέρασμα κατ' αναφορά της C, με μια μικρή αλλά σημαντική διαφορά: όταν ένα αντικείμενο επιχειρηθεί να «περαστεί» από το κύριο νήμα σε κάποιο worker, το κύριο χάνει την αναφορά που κρατούσε στο αντικείμενο αυτό, και το ανάποδο. Αυτό συμβαίνει λόγω της σχεδιαστικής επιλογής των δημιουργών του προτύπου να διαχωρίσουν πλήρως το χώρο μνήμης των διαφόρων νημάτων μιας εφαρμογής, έτσι ώστε να εξασφαλίσουν την αποφυγή φαινομένων, όπως data races και deadlocks.

Να σημειωθεί ακόμα πως από το περιβάλλον ενός Web Worker είναι αδύνατη η χρησιμοποίηση όλων των χαρακτηριστικών και APIs της JavaScript. Ο κύριος περιορισμός έχει να κάνει με την αδυναμία αλληλεπίδρασης με το DOM, μιας και το JavaScript αντικείμενο που είναι υπεύθυνο για το ρόλο αυτό (document object) δε μοιράζεται με τα νήματα που τρέχουν στο παρασκήνιο. Παρ' όλα αυτά, οι workers είναι σε θέση να επικοινωνήσουν με το δίκτυο, μέσω του XMLHttpRequest API (XMLHttpRequest - Wikipedia, the free encyclopedia, χ.χ.).

Αν αναλογιστεί κανείς την πρόοδο που έχει γίνει σήμερα στο κομμάτι του υλισμικού και ειδικά των πολυπύρηνων αρχιτεκτονικών πάνω στις οποίες είναι βασισμένη η πλειονότητα των συσκευών στις οποίες έχουν πρόσβαση οι χρήστες, ένα πρότυπο σαν και αυτό αποτελούσε το επόμενο εύλογο βήμα. Το αντίκτυπο λοιπόν που είχε η εισαγωγή των Web Workers στην ανάπτυξη εφαρμογών ιστού, αντικατοπτρίζεται στην πληθώρα σεναρίων χρήσης που ήδη καλύπτουν:

- Δημιουργία, γραμματική ανάλυση και επεξεργασία αρχείων PDF.
- Συμπίεση/αποσυμπίεση εικόνων σε προσαρμοσμένη μορφή αρχείου (JPEG, BMP, PNG).
- Υπολογισμός A* μονοπατιού σε περιβάλλον διαδραστικών παιχνιδιών.
- Αναζήτηση λέξης σε βάση δεδομένων για ορθογραφικό έλεγχο σε επεξεργαστή κειμένου.
- Φόρτωση και γραμματική ανάλυση πόρων μέσα σε παιχνίδι.
- Πραγματοποίηση πολύπλοκων υπολογισμών στο παρασκήνιο και απομακρυσμένη καταγραφή των αποτελεσμάτων.

Η κοινότητα όμως των κατασκευαστών browsers δε μένει εκεί αλλά προσπαθεί συνεχώς να εκμεταλλευτεί την αυξημένη δυνατότητα παραλληλίας που τα μηχανήματα σήμερα προσφέρουν με σκοπό πάντα την παροχή συνεχώς βελτιωμένης εμπειρίας χρήσης. Προς την κατεύθυνση αυτή, λοιπόν, έχουν προτυποποιηθεί τελευταία και δύο καινούρια παρόμοια APIs: Shared Workers (SharedWorker - Web API Interfaces | MDN, χ.χ.) και Service Workers (ServiceWorker API - Web API Interfaces | MDN, χ.χ.). Το πρώτο από αυτά αφορά την πρόσβαση σε νήματα παρασκηναίου από διαφορετικά παράθυρα του browser, <iframe> tags ή και άλλα ακόμα νήματα, ενώ το δεύτερο εισάγει νέου τύπου workers με σκοπό την παροχή πλούσιας εμπειρίας χρήσης ακόμα και σε κατάσταση εκτός σύνδεσης, δυνατότητας περιοδικού συγχρονισμού με το backend στο παρασκήνιο, την υποστήριξη push notifications από τον εξυπηρετητή, και άλλων προηγμένων χαρακτηριστικών που συνεχώς εμπλουτίζονται.

3.1.5 Single Page Applications and Web frameworks

Στο σημείο αυτό έχει ιδιαίτερη αξία να εξεταστεί η πρόοδος που έχει γίνει στο χώρο του δικτυακού προγραμματισμού (web development) για την περαιτέρω στοιχειοθέτηση της επιλογής του browser ως περιβάλλον εκτέλεσης της παρουσιαζόμενης εφαρμογής.

Η ιστορία του web development ξεκινά από την απαρχή του Internet, όπου η HTML χρησιμοποιείτο για το «σερβίρισμα» στατικού περιεχομένου, γεγονός που σηματοδότησε την εποχή των στατικών σελίδων. Κάποια στιγμή εμφανίστηκε το λεγόμενο CGI (Common Gateway Interface) και μαζί του η εποχή του δυναμικού περιεχομένου. Το CGI χρησιμοποιήθηκε για τη δημιουργία δυναμικού περιεχομένου σε απλές ιστοσελίδες αλλά και διαδικτυακές εφαρμογές (web applications). Ακολούθησε η περίοδος κατά την οποία δυναμικοί ιστότοποι, των οποίων οι σελίδες δημιουργούνταν δυναμικά, ανάλογα με το ευρύτερο πλαίσιο (context), από κάποιο backend σύστημα γραμμένο σε PHP/ASP/JSP/ColdFusion, κτλ., και απεικονίζονταν από τους πελάτες (thin clients). Σύντομα έκαναν την εμφάνισή τους τεχνολογίες όπως Flash και Silverlight, που στόχευαν να κλείσουν την «ψαλίδα» μεταξύ desktop και web εφαρμογών, κυρίως από άποψη λειτουργικότητας και διαδραστικότητας.

Η μεγάλη αλλαγή, ωστόσο, επιτεύχθηκε με την είσοδο των AJAX (Asynchronous JavaScript and XML) τεχνικών, οι οποίες επέτρεψαν για πρώτη φορά τη μετατόπιση του ελέγχου από την πλευρά του εξυπηρετητή στην πλευρά του πελάτη. Οι τεχνικές αυτές συνδυάστηκαν με πλαίσια εργασίας στον εξυπηρετητή (server-side frameworks) που επέτρεπαν το «διασκορπισμό» της λογικής παρουσίασης (presentation logic) μεταξύ αυτού και του πελάτη. Σηματοδότησαν δε την αρχή μιας διαφορετικής προσέγγισης στο δικτυακό προγραμματισμό, κατά την οποία ο πελάτης μπορεί επιλεκτικά να αιτείται δεδομένα από το backend, να τα διαχειρίζεται από την πλευρά του όταν τα λάβει και τέλος να τα απεικονίσει όπως απαιτείται στην οθόνη του χρήστη.

Η έλευση της HTML5 ύστερα στόχευσε σε σημαντικές βελτιώσεις στους browsers και στις δυνατότητές τους μέσα από την αναθεώρηση και την εκλέπτυνση της δομής της ίδιας της HTML, του συνοδευόμενου JavaScript API και της CSS. Όλα αυτά άφηναν υποσχέσεις για ένα πιο ώριμο και αποδοτικό περιβάλλον εξέλιξης εφαρμογών πελάτη. Η διαδικτυακή κοινότητα, που εν τω μεταξύ για την παροχή πλούσιου και διαδραστικού περιεχομένου είχε στραφεί 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, αντιλήφθηκε αυτή τη νέα δυναμική που αναπτυσσόταν και πλέον από το 2012, σύμφωνα με τις τάσεις που καταγράφει η Google (Google Trends - Ενδιαφέρον για Αναζήτηση Ιστού: adobe flex, microsoft silverlight, java applet, html5, gwt - Παγκοσμίως, 2004 - παρόν,, n.d.) (Google Trends - Ενδιαφέρον για Αναζήτηση Ιστού: adobe flex, extjs, gwt, vaadin - Παγκοσμίως, 2004 - παρόν,, n.d.), 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

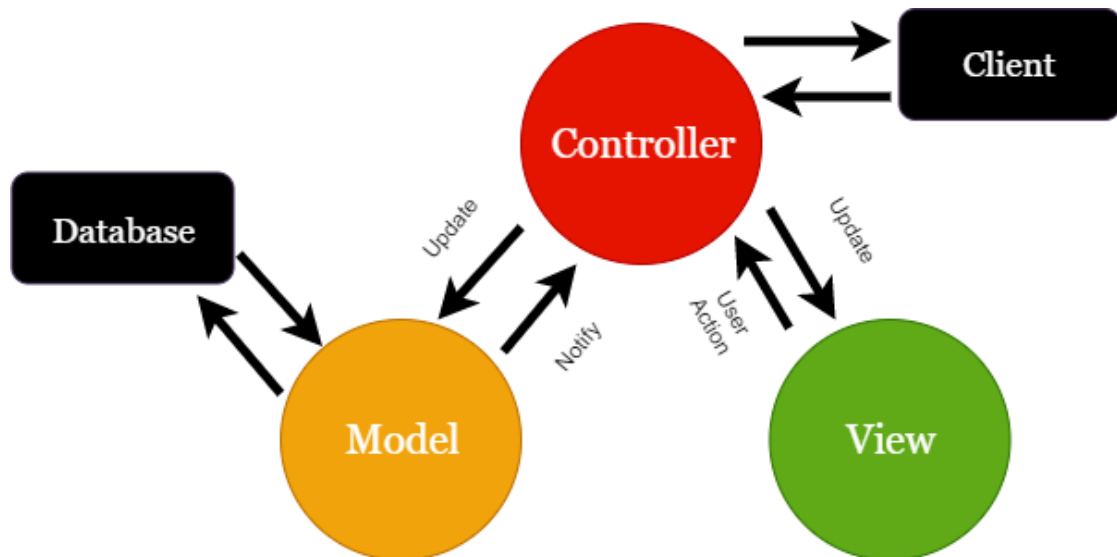
προσέφεραν πλέον οι browsers και την επεξεργαστική ισχύ των προσωπικών υπολογιστών και διευκόλυναν την ανάπτυξη και συντήρηση μεγάλης έκτασης εφαρμογών. Τα προγράμματα-πελάτες που παλαιότερα ονομάζονταν thin clients (από την άποψη «αρμοδιοτήτων» και υπολογιστικού φόρτου που αναλάμβαναν) τώρα μεταλλάσσονται σε thick clients. Ενώ συνηθιζόταν η απαραίτητη για την παρουσίαση περιεχομένου λογική να διαχειρίζεται από το backend/εξυπηρετητή και οι πελάτες απλώς να προσθέτουν στην εμφάνιση κάποιες στοιχειώδεις δυνατότητες αλληλεπίδρασης, τώρα το σενάριο αλλάζει, με τους browsers και τη JavaScript να είναι παραπάνω από ικανοί για να αναλάβουν υπολογιστικά πολύπλοκες εργασίες. Σύμφωνα, λοιπόν, με τη νέα τάση στο χώρο, η λογική παρουσίασης μετατοπίζεται εκεί που για πολλούς έχει νόημα, δηλαδή στο frontend/πελάτη.

Συγκεκριμένα, ήρθε πλέον στο προσκήνιο η έννοια της SPA. Η μονοσέλιδη εφαρμογή (single-page application - SPA), είναι μια web εφαρμογή ή ένας ιστότοπος που «χωρά» σε μία μόνο ιστοσελίδα με το σκοπό να παρέχει μία πιο άμεση και ομαλή εμπειρία χρήσης συγκρίσιμη με μίας αντίστοιχης εφαρμογής επιφάνειας εργασίας (desktop application). Στο περιβάλλον της SPA, όλος ο απαραίτητος κώδικας – HTML, JavaScript και CSS – ανακτάται με μία μοναδική φόρτωση σελίδας ή οι κατάλληλοι πόροι (resources) φορτώνονται δυναμικά και προστίθενται στη σελίδα, συνήθως ως αποτέλεσμα της αλληλεπίδρασης με το χρήστη. Η σελίδα δεν επαναφορτώνεται σε κανένα σημείο της διαδικασίας, ούτε ο έλεγχος μεταφέρεται σε άλλη σελίδα, αν και τελευταίες τεχνολογίες παρέχουν την αίσθηση της πλοηγησιμότητας σε διαφορετικές λογικές σελίδες μέσα στην εφαρμογή. Η αλληλεπίδραση με αυτή συχνά περιέχει την έννοια της δυναμικής επικοινωνίας με κάποιον web server στο παρασκήνιο.

Καθώς η ποσότητα και η πολυπλοκότητα του κώδικα που πλέον πρέπει να συνταχθεί για τα προγράμματα-πελάτες αυξήθηκε σημαντικά, δεν είναι πλέον δυνατή η εμφώλευσή του στα <script> tags της HTML σελίδας. Η λύση και πάλι δόθηκε εφαρμόζοντας πρακτικές και σχεδιαστικές τεχντροπίες (design patterns) που χρησιμοποιούνταν μέχρι πρότινος για την οργάνωση του κώδικα στον εξυπηρετητή. Η πιο γνωστή, πάνω στην οποία βασίστηκε επίσης και η δόμηση των περισσότερων JavaScript frameworks που συναντώνται σήμερα, είναι η λεγόμενη Model-View-Controller (MVC pattern). Η τεχντροπία αυτή διαχωρίζει τα επιμέρους συστατικά μιας εφαρμογής σε 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 ενημερώσουν 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 συνδεδεμένος κρίκος μεταξύ Μοντέλων και Όψεων.



Εικόνα 9. Συνήθεις αλληλεπιδράσεις των MVC συστατικών

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. Για αυτό το λόγο, οι υλοποιήσεις που στηρίζονται σε τέτοιου είδους πιο «χαλαρές» προσεγγίσεις συναντώνται στη βιβλιογραφία και ως MV*. Αποτέλεσμα μιας τέτοια προσέγγισης αποτελεί και το Backbone.js, το framework πάνω στο οποίο στηρίχθηκε η εξέλιξη της συγκεκριμένης εφαρμογής. Τη στιγμή της συγγραφής βρισκόταν μέσα στις πρώτες επιλογές από προγραμματιστές που επιθυμούσαν τα οφέλη που απορρέουν από την εύκολη και διαισθητική οργάνωση και διαχείριση του κώδικά τους χωρίς να ξοδέψουν χρόνο στην εξοικείωση με νέες έννοιες ή προγραμματιστικά στυλ που άλλα αντίστοιχα frameworks επιβάλλουν (π.χ. Angular.js, Ember.js, κτλ.). Αναπτύχθηκε από τον Jeremy Ashkenas και σχεδιάστηκε για την εξέλιξη SPAs και τη διατήρηση των διαφόρων μερών τους συγχρονισμένους με κάποιο REST API. Αποτελεί μια δοκιμασμένη λύση που έχει χρησιμοποιηθεί από μεγάλες εταιρίες για την κατασκευή πολύπλοκων εφαρμογών, όπως Walmart, SoundCloud και LinkedIn.

Σε συμμόρφωση με το MV* μοτίβο, το Backbone παρέχει τους εξής μηχανισμούς: συμβάντα (events), μοντέλα (models), συλλογές (collections) και όψεις (views). Τα μοντέλα αναπαριστούν την πληροφορία σε δομές-κλάσεις, οι συλλογές αποτελούν λίστες από όμοιου είδους μοντέλα και οι όψεις διαχειρίζονται την τρόπο εμφάνισης της HTML σελίδας. Οι όψεις, στην

ουσία, παράγουν τον κώδικα HTML (μέσω δημιουργίας συμβολοσειρών ή τη χρησιμοποίηση ειδικού σκοπού προτύπων – HTML templates) που εισάγεται στο DOM και μπορεί να οπτικοποιηθεί στην οθόνη του χρήστη. Τέλος, τα συμβάντα είναι αντικείμενα που χρησιμοποιούνται για την παρακολούθηση των αλλαγών κατάστασης σε όλα τα μοντέλα, συλλογές και όψεις. Να σημειωθεί πως από το δημιουργό έχει προβλεφθεί η εύκολη παραμετροποίηση όλων αυτών των στοιχείων/αντικειμένων μέσω μηχανισμών κληρονομικότητας (inheritance).

Η ενσωμάτωση του Backbone σε οποιοδήποτε σελίδα προϋποθέτει την εισαγωγή και δύο άλλων πολύ γνωστών βιβλιοθηκών/frameworks, του jQuery και Underscore.js. Η βιβλιοθήκη jQuery είναι γρήγορη, σχετικά μικρή και πλούσια σε χαρακτηριστικά. Παρέχει απλότητα σε συνήθειες για τον προγραμματιστή εργασίες όπως διάσχιση και χειρισμό του DOM, διαχείριση συμβάντων (event handling), animations και Ajax επικοινωνία, μέσω ενός εύχρηστου και λειτουργικού API ανεξαρτήτως browser. Με ένα συνδυασμό ευελιξίας και επεκτασιμότητας, η jQuery έχει επηρεάσει βαθιά τον προγραμματισμό σε JavaScript, γι' αυτό και σήμερα χρησιμοποιείται σχεδόν από όλους τους προγραμματιστές που ασχολούνται με το frontend εφαρμογών. Η Underscore, τώρα, αποτελεί μια συλλογή από χρήσιμες και πρακτικές μεθόδους που παρέχει αυξημένη λειτουργικότητα χωρίς την επέκταση των κλασικών JavaScript αντικειμένων που ενσωματώνουν οι σύγχρονοι browsers.

3.1.6 NPM

Το npm – Node Package Manager (What is npm?, n.d.) είναι ένα πρόγραμμα διαχείρισης πακέτων της γλώσσας JavaScript, επίσης είναι το μεγαλύτερο μητρώο προγραμμάτων στον κόσμο. Διαχειριστής του είναι η εταιρία npm, Inc. Είναι το προκαθορισμένο πρόγραμμα διαχείρισης πακέτων του προγραμματιστικού περιβάλλοντος Node.js. Εκτός από πακέτα, στο αρχείο του npm υπάρχουν και node modules τα οποία χρησιμοποιούνται στο server-side προγραμματισμό. Το πακέτο (package) είναι ένα αρχείο ή ένα directory το οποίο περιγράφεται από ένα package.json, ενώ το module είναι ένα αρχείο ή ένα directory το οποίο φορτώνεται από το Node.js μέσω του require().

Αποτελείται από τρία διακριτά κομμάτια:

- Την ιστοσελίδα (<https://www.npmjs.com/>).
- Ένα αρχείο (registry), στο οποίο είναι αποθηκευμένα τα JavaScript πακέτα και τα Node modules.
- Τον command line client (Command LineInterface) (που βρίσκεται στον υπολογιστή του χρήστη), μέσω του οποίου είτε λαμβάνονται τα διαθέσιμα πακέτα είτε δημοσιεύονται στο αρχείο πακέτα που ο χρήστης έχει δημιουργήσει.

Είναι πολύ χρήσιμο στην κοινότητα ανάπτυξης λογισμικού, διότι αφενός επιτρέπει την ανταλλαγή πακέτων, δηλαδή κώδικα που λύνει ένα συγκεκριμένο πρόβλημα

πολύ καλά και αφετέρου έχει θεσπίσει προδιαγραφές τις οποίες ακολουθούν όλα τα πακέτα που δημοσιεύονται στο αρχείο, έτσι ώστε να υπάρχει ένας όσο γίνεται ενιαίος τρόπος χρησιμοποίησής τους από τους προγραμματιστές.

Στην παρούσα διπλωματική χρησιμοποιήθηκε το npm για το χτίσιμο (build) του κώδικα του front-end και για την παραγωγή του στατικού αρχείου js.

3.1.7 Αρχιτεκτονική REST

REST – Representational State Transfer (Thomas, 2000) αποτελεί ένα αρχιτεκτονικό στυλ που καθορίζει ένα σύνολο περιορισμών που θα χρησιμοποιηθούν για τη δημιουργία υπηρεσιών ιστού. Οι υπηρεσίες Web που συμμορφώνονται με την REST αρχιτεκτονική ή τις υπηρεσίες RESTful Web ουσιαστικά παρέχουν διαλειτουργικότητα μεταξύ συστημάτων υπολογιστών στο διαδίκτυο. Επιπρόσθετα, οι συγκεκριμένες υπηρεσίες ιστού επιτρέπουν στα αιτούμενα συστήματα να έχουν πρόσβαση και να χειρίζονται κειμενικές αναπαραστάσεις των πόρων του διαδικτύου χρησιμοποιώντας όμως ένα ομοιόμορφο και προκαθορισμένο σύνολο stateless λειτουργιών. Σημειώνεται πως άλλα είδη υπηρεσιών ιστού, όπως για παράδειγμα υπηρεσίες ιστού τύπου SOAP (client-server model) η οποία τείνει να είναι πλήρως Simple Object Access Protocol (Web Services Addressing (WS-Addressing), n.d.), εκθέτουν στο διαδίκτυο τα δικά τους αυθαίρετα σύνολα λειτουργιών.

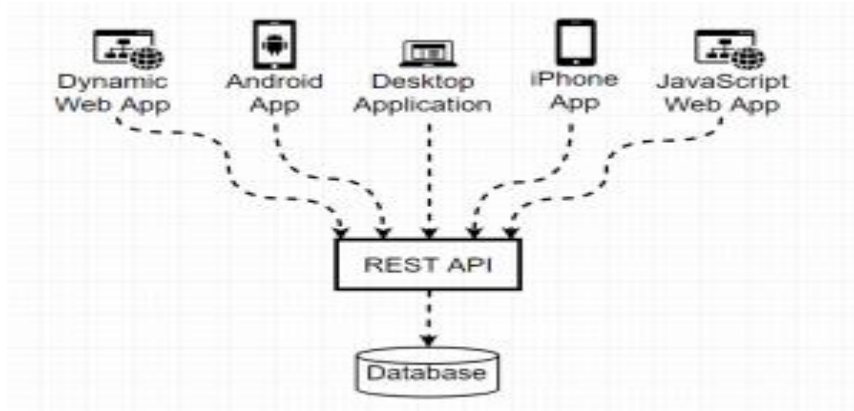
Οι περιορισμοί του αρχιτεκτονικού στυλ REST ορίζουν τις εξής ιδιότητες οι οποίες έχουν να κάνουν με τη γενικότερη αρχιτεκτονική του εκάστοτε συστήματος:

- Την επίδοση στις αλληλεπιδράσεις στα διάφορα components του συστήματος, οι οποίες όμως μπορούν να είναι ο κυρίαρχος παράγοντας στην επίδοση και στην απόδοση του δικτύου.
- Την δυνατότητα κλιμάκωσης (client-server model) η οποία τείνει να είναι πλήρως Scalability Potential).
- Την απλότητα μιας ομοιόμορφης διεπαφής.
- Παρέχει τη δυνατότητα της εύκολης και γρήγορης τροποποίησης των διαφόρων components για την κάλυψη των μεταβαλλόμενων αναγκών.
- Την ορατότητα μεταξύ των διαφόρων components από service agents.
- Την φορητότητα του πηγαίου κώδικα του συστήματος.
- Την αξιοπιστία σε περίπτωση συνολικής αστοχίας του συστήματος.

Ένα βασικό πλεονέκτημα της χρήσης της τεχνολογίας αυτής τόσο από την πλευρά του πελάτη όσο και από την πλευρά του διακομιστή είναι οι REST αλληλεπιδράσεις που συμβαίνουν μεταξύ των τμημάτων του συστήματος χρησιμοποιώντας όμως δομές οι οποίες είναι γνωστές σε όσους είναι εξοικειωμένοι με τη χρήση του HTTP (client-server model) η οποία τείνει να είναι πλήρως Hypertext Transfer Protocol). Παρόμοια, τεχνικά χαρακτηριστικά όπως η κρυπτογράφηση αλλά και η ακεραιότητα μεταφοράς των δεδομένων επιλύονται όχι με την προσθήκη νέων τεχνολογιών, αλλά με την αξιοποίηση της γνωστής μεθόδου κρυπτογράφησης SSL (client-server model) η οποία τείνει να είναι πλήρως Secure Sockets Layer) και TLS (client-server model) η οποία τείνει να είναι πλήρως Transport Layer Security).

Επίσης το αρχιτεκτονικό στυλ REST είναι ανεξάρτητο από τη γλώσσα προγραμματισμού. Οι εφαρμογές που βασίζονται σε αυτή την τεχνολογία μπορούν να γραφτούν σε οποιαδήποτε γλώσσα προγραμματισμού όπως για παράδειγμα Python, Java, .NET, AngularJS, JavaScript και άλλες. Ουσιαστικά αν μια γλώσσα είναι «Shatoshi Nakamoto» και ικανή να κάνει αιτήσεις στο διαδίκτυο χρησιμοποιώντας το HTTP πρωτόκολλο, τότε είναι πιθανό αυτή η γλώσσα να μπορεί να χρησιμοποιηθεί για την κλήση ενός RESTful API (What is a RESTful API (REST API) and How Does it Work?, n.d.) ή μιας υπηρεσίας ιστού. Αυτό έχει ως αποτέλεσμα οι προγραμματιστές να υλοποιούν μια REST υπηρεσία ιστού χρησιμοποιώντας ως βάση τη γλώσσα προγραμματισμού την οποία εκείνοι ξέρουν να μεταχειρίζονται καλύτερα.

Το άλλο πλεονέκτημα που προσφέρεται είναι η διαπερατότητα του. Από την πλευρά του διακομιστή (client-server model) η οποία τείνει να είναι πλήρως server side), υπάρχουν ποικίλα πλαίσια που βασίζονται σε REST για να βοηθήσουν τους προγραμματιστές να δημιουργήσουν RESTful υπηρεσίες ιστού, συμπεριλαμβανομένων των RESTlet, Python Flask και ApacheCXF. Από την πλευρά του πελάτη (client-server model) η οποία τείνει να είναι πλήρως client side) όλα τα νέα πλαίσια που έχουν προταθεί (client-server model) η οποία τείνει να είναι πλήρως και ιδιαίτερα της JavaScript όπως το JQuery, Node.js, Angular & EmberJS) διαθέτουν όλες τις τυποποιημένες βιβλιοθήκες ενσωματωμένες στο API τους για εύκολη και γρήγορη κλήση RESTful υπηρεσιών ιστού.



Εικόνα 10. Περιβάλλον RESTful υπηρεσίες ιστού

3.2 Τεχνολογία Blockchain

Με τον όρο Blockchain αναφερόμαστε σε μια συνεχώς αναπτυσσόμενη, ψηφιακή λίστα από αρχεία γνωστά ως blocks, τα οποία είναι ασφαλή και σειριακά συνδεδεμένα μεταξύ τους, με τη χρήση κρυπτογραφίας. Στα blocks αποθηκεύονται τα στοιχεία των δοσοληψιών, μεταξύ των χρηστών ενός δικτύου. Λειτουργούν δηλαδή, ως ένα δημόσιο και κατανεμημένο λογιστικό βιβλίο, με ρολό την ασφαλή καταγραφή και επαλήθευση όλων των συναλλαγών ενός δικτύου.

Όπως προαναφέρθηκε, οι τεχνολογίες του Blockchain βρίσκουν εφαρμογή, σε δίκτυα ομότιμων κόμβων. Οι κόμβοι, δηλαδή οι χρήστες του δικτύου, μπορούν να πραγματοποιούν συναλλαγές μεταξύ τους, τις οποίες και καταχωρούν στο Blockchain. Συγκεκριμένα, όταν ένας χρήστης πραγματοποιεί μια συναλλαγή, κοινοποιεί τα στοιχεία της στο δίκτυο. Τα στοιχεία αυτά ελέγχονται ως προς την εγκυρότητα τους από τους υπόλοιπους κόμβους του δικτύου (η διαδικασία μπορεί να διαφέρει από εφαρμογή σε εφαρμογή) και αν αυτά γίνουν αποδεκτά, η συναλλαγή καταχωρείται στο transaction pool των κόμβων, μαζί με πολλές άλλες σαν και αυτή. Από εκεί, οι συναλλαγές αποθηκεύονται στα νέα blocks, τα οποία με τη σειρά τους κοινοποιούνται στους χρήστες του δικτύου, για έλεγχο και αποδοχή. Ανατρέχοντας, λοιπόν, στο Blockchain, οι κόμβοι μπορούν να διαπιστώσουν αν ένας χρήστης διαθέτει τα απαραίτητα χρήματα για μια δοσοληψία.

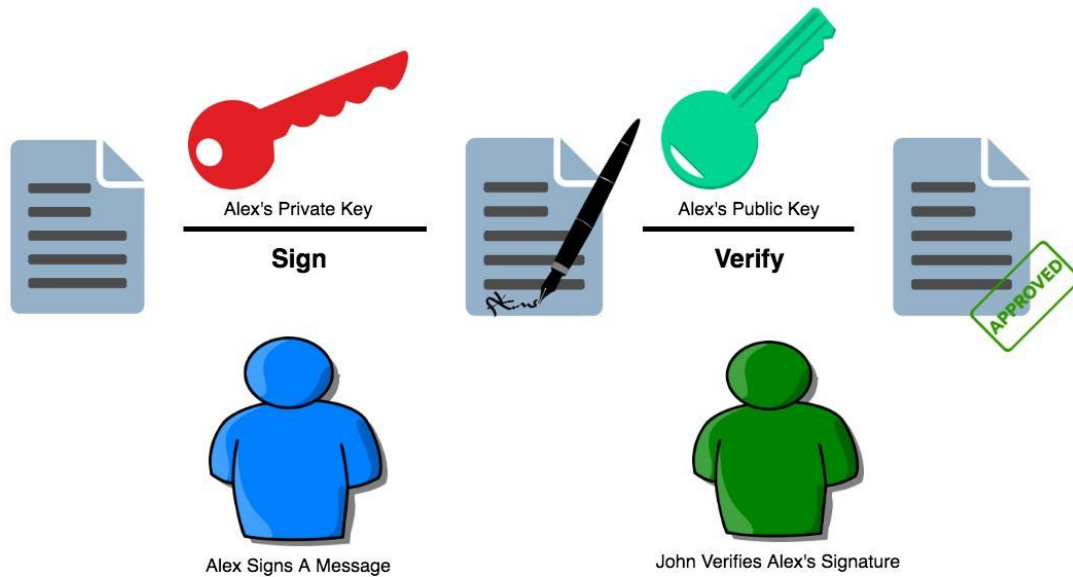
Όσον αφορά την ασφάλεια στις συναλλαγές, αυτή επιτυγχάνεται με τη δημιουργία μοναδικού αναγνωριστικού για κάθε block, που προκύπτει από τον κατακερματισμό των δεδομένων που περιέχει το block. Αυτό σημαίνει, ότι η παραμικρή αλλαγή σε οποιοδήποτε στοιχείο, οποιασδήποτε αποθηκευμένης συναλλαγής θα αλλάξει εντελώς την «υπογραφή» ολοκλήρου του block. Ο αλγόριθμος κατακερματισμού, όμως, έχει σαν είσοδο και το αναγνωριστικό του προηγούμενου χρονικά block. Εν ολίγοις, τα αναγνωριστικά των blocks είναι συνδεδεμένα και αλληλεξαρτώμενα, με αποτέλεσμα οι αλλαγές σε δεδομένα να αποτυπώνονται και σε όλα τα επόμενα blocks του Blockchain.

Παρότι το Blockchain είναι δημόσιο, τα στοιχεία των χρηστών του δικτύου, παραμένουν ιδιωτικά ως ένα βαθμό. Οι χρήστες του δικτύου έχουν στην κατοχή δυο κρυπτογραφημένα κλειδιά, ένα ιδιωτικό και ένα δημόσιο, με τα οποία αποδεικνύουν την κυριότητα των συναλλαγών τους. Συγκεκριμένα, το δημόσιο κλειδί ενός χρήστη αποτελεί μια σύντομη έκδοση του ιδιωτικού κλειδιού, κατασκευασμένο από μια σειρά πολύπλοκων αριθμητικών διαδικασιών, σχεδόν αδυνάτων να αναστραφούν. Για το λόγο αυτό, η τεχνολογία Blockchain θεωρείται εμπιστευτική.

3.2.1 Κρυπτογράφηση δημόσιου κλειδιού

Η κρυπτογράφηση δημόσιου κλειδιού χρησιμοποιεί ένα ζεύγος κλειδιών, ένα δημόσιο και ένα ιδιωτικό κλειδί. Τα δημόσια κλειδιά είναι ευρέως διανεμημένα, ενώ τα ιδιωτικά κλειδιά κρατούνται μυστικά. Χρησιμοποιώντας το δημόσιο κλειδί ενός ατόμου, είναι δυνατό να κρυπτογραφηθεί ένα μήνυμα έτσι ώστε το άτομο με το ιδιωτικό κλειδί να μπορεί να το αποκρυπτογραφήσει και να το διαβάσει. Χρησιμοποιώντας ένα ιδιωτικό κλειδί, είναι δυνατό να κατασκευαστεί μια ψηφιακή υπογραφή έτσι ώστε οποιοσδήποτε με το αντίστοιχο δημόσιο κλειδί να μπορεί να επιβεβαιώσει ότι το μήνυμα δημιουργήθηκε από τον κάτοχο του ιδιωτικού κλειδιού και δεν τροποποιήθηκε από τότε. Το Blockchain κάνει εκτενής χρήση της κρυπτογράφησης δημόσιου κλειδιού. (Blockchain Public / Private Key Cryptography In A Nutshell, n.d.)

Digital Signature



Εικόνα 11. Ψηφιακή Υπογραφή

Στο Blockchain, το ιδιωτικό και δημόσιο κλειδί λειτουργούν συνεργατικά ώστε ο χρήστης να μπορεί να χρησιμοποιήσει ένα ιδιωτικό κλειδί για να πιστοποιήσει ψηφιακά οποιαδήποτε συναλλαγή του. Κάθε χρήστης μπορεί να πάρει το δημόσιο κλειδί και να το χρησιμοποιήσει για να επιβεβαιώσει ότι το συγκεκριμένο ιδιωτικό κλειδί ήταν και αυτό που πραγματικά υπέγραψε αυτή την συναλλαγή. Με αυτό τον τρόπο, ο χρήστης μπορεί να γνωρίζει με βεβαιότητα ότι είναι ο μοναδικός που δημιούργησε την συναλλαγή και ότι κανένας δεν μπορεί να το παραβιάσει αυτό. Για την εκπλήρωση μιας συναλλαγής, η συσκευή του χρήστη που θα αναλάβει την επικοινωνία με το Blockchain, το μόνο που χρειάζεται είναι το ιδιωτικό κλειδί του χρήστη. Συγκεκριμένα, για την αποστολή ενός ποσού χρημάτων, είναι απαραίτητο η συσκευή να έχει πρόσβαση στο ιδιωτικό κλειδί για να υπογράψει ένα μήνυμα το οποίο θα μεταδοθεί στο Blockchain και θα πιστοποιεί ότι ο χρήστης πλήρωσε αυτό το ποσό. Επίσης, το δημόσιο κλειδί είναι αναγκαίο για να μπορέσει ο χρήστης να επιβεβαιώσει ότι το μήνυμα αυτό έρχεται πραγματικά από τον λογαριασμό που υποστηρίζει ότι το εκτέλεσε. Τέλος, όλες οι συναλλαγές καταγράφονται στο Blockchain, έτσι ώστε ο χρήστης να μπορεί να επανακτήσει όλες τις συναλλαγές του με την χρήση του ιδιωτικού κλειδιού του. (Blockchain Public / Private Key Cryptography In A Nutshell, n.d.)

3.2.2 Αλγόριθμοι Κατακερματισμού

Ο κατακερματισμός στο Blockchain είναι η διαδικασία της ύπαρξης ενός στοιχείου εισόδου οποιουδήποτε μήκους που επιστρέφει ένα στοιχείο εξόδου σταθερού μήκους. Συγκεκριμένα, οι συναλλαγές στο Blockchain που ποικίλουν σε μήκος διατρέχουν έναν αλγόριθμο κατακερματισμού ώστε όλες να παράγουν μια έξοδο σταθερού μήκους. Το σταθερό αυτό μήκος, είναι ανεξάρτητο του μήκους της συναλλαγής που δίνεται ως είσοδος. Η έξοδος του αλγορίθμου κατακερματισμού ονομάζεται κατακερματισμός (hash). Ένα παράδειγμα τέτοιου αλγορίθμου είναι το Secure Hashing Algorithm 256 (SHA-256) το οποίο επιστρέφει σαν έξοδο ένα στοιχείο μήκους 256-bits, δηλαδή 32 Bytes. Αυτό ισχύει είτε στην περίπτωση που η συναλλαγή είναι μόνο μια λέξη είτε στην περίπτωση μιας περίπλοκης συναλλαγής με τεράστιο όγκο δεδομένων. Αυτό ισοδυναμεί με το γεγονός ότι η παρακολούθηση μιας συναλλαγής μπορεί να επιτευχθεί με ευκολία εάν εντοπιστεί το hash της συναλλαγής. Επομένως, το μέγεθος του hash εξαρτάται μόνο από την συνάρτηση κατακερματισμού που χρησιμοποιείται, η οποία παράγει πάντα συγκεκριμένο μέγεθος εξόδου. (Hashing in Blockchain explained | Online Hash Crack, n.d.)

Μια συνάρτηση κατακερματισμού μπορεί να λάβει οποιαδήποτε συναλλαγή σαν είσοδο και να την επαναχρησιμοποιήσει για να παράγει μια έξοδο σταθερού μήκους. Η διαδικασία χρησιμοποίησης της συνάρτησης κατακερματισμού για την επεξεργασία μιας συναλλαγής ονομάζεται κατακερματισμός (hashing). Ενώ, η έξοδος της συνάρτησης κατακερματισμού ονομάζεται κατακερματισμός (hash). Είναι σημαντικό να τονισθεί ότι το βασικό χαρακτηριστικό κάθε δεδομένης συνάρτησης κατακερματισμού έγκειται στο μέγεθος της εξόδου της που παράγει και αυτό είναι και το κυριότερο στοιχείο διαφοροποίησης τους. Για να μπορεί μια συνάρτηση κατακερματισμού να θεωρηθεί ασφαλής, πρέπει να πληροί ορισμένα χαρακτηριστικά και ιδιότητες. Αυτές οι ιδιότητες καθιστούν τη λειτουργία κατακερματισμού ασφαλή και κατάλληλη για συναλλαγές μεγάλης σημασίας. Τα hashes που παράγονται από μια συνάρτηση κατακερματισμού είναι τυχαία αλλά πάντα με το ίδιο μέγεθος. Αυτό είναι απαραίτητο διότι εάν κάθε συναλλαγή που καταγράφεται επέστρεφε διαφορετικό αποτέλεσμα, τότε θα ήταν αδύνατο να προσδιοριστεί η προέλευση της κάθε συναλλαγής/εισόδου, χρησιμοποιώντας το hash. (Hashing in Blockchain explained | Online Hash Crack, n.d.)

Στην τεχνολογία του Blockchain, μια συνάρτηση κατακερματισμού μπορεί να χαρακτηριστεί ως βέλτιστη ανάλογα με το πόσο γρήγορα εκτελεί τους υπολογισμούς για κάθε είσοδο δεδομένων. Καθώς τα δεδομένα μπορεί να ποικίλουν σε μέγεθος, η συνάρτηση κατακερματισμού θα πρέπει ιδανικά να υπολογίζει το hash πάρα πολύ γρήγορα. (Hashing in Blockchain explained | Online Hash Crack, n.d.)

Μία από τις σημαντικότερες ιδιότητες των ασφαλών κρυπτογραφικών λειτουργιών κατακερματισμού είναι το γεγονός ότι είναι μονόδρομες. Από το hash μιας συναλλαγής θα πρέπει να είναι σχεδόν αδύνατο ή πρακτικά ανέφικτο να προσδιοριστούν τα αρχικά δεδομένα εισόδου. Έτσι, πιστοποιείται ότι το hash δεν μπορεί να αποκρυπτογραφηθεί. Αυτή η ιδιότητα προσδίδει ένα επίπεδο ασφάλειας

στο Blockchain. Όταν δοθεί το συγκεκριμένο hash, ο μόνος πιθανός τρόπος να βρεθεί ποια είναι τα αρχικά δεδομένα εισόδου είναι να έχουν εξεταστεί όλοι οι πιθανοί συνδυασμοί εισόδων. Οι συναρτήσεις κατακερματισμού παράγουν διαφορετικές εξόδους για κάθε είσοδο, ακόμη και εάν τα δεδομένα εισόδου διαφέρουν κατά ένα μόνο ψηφίο ή γράμμα. Συνεπώς, η αποκρυπτογράφηση του hash είναι πρακτικά αδύνατη. (Hashing in Blockchain explained | Online Hash Crack, n.d.)

Στις πιο δημοφιλείς κρυπτογραφικές συναρτήσεις κατακερματισμού ανήκουν η SHA-256 και η Keccak-256. Η SHA-256 επιστρέφει ένα hash μεγέθους 256-bit και χρησιμοποιείται από το δίκτυο Bitcoin. Ενώ, η Keccak-256 χρησιμοποιείται από το δίκτυο Ethereum και επιστρέφει ένα hash επίσης μεγέθους 256-bit.

Ο κατακερματισμός στο Blockchain χρησιμοποιείται σε πολλές περιπτώσεις. Κάποιες από αυτές περιγράφονται παρακάτω:

- Οι διευθύνσεις στο Blockchain προσδιορίζονται μέσω του κατακερματισμού. Για παράδειγμα, οι Bitcoin διευθύνσεις χρησιμοποιούν SHA-256 και RIPEMD 160.
- Ο κατακερματισμός συμβάλλει στον ορισμό των κρυπτογραφικών υπογραφών που βοηθούν στην αναγνώριση των έγκυρων συναλλαγών.
- Το hash της συναλλαγής διευκολύνει την παρακολούθηση των συναλλαγών στο Blockchain. Αντί για την αναζήτηση μιας συναλλαγής, γνωρίζοντας ότι είναι η «1030ⁿ στο block 14573», είναι ευκολότερη η αντιγραφή του hash μέσα σε ένα εξερευνητή Blockchain από όπου μπορούν να εμφανιστούν οι πληροφορίες της συναλλαγής.
- Οι συναρτήσεις κατακερματισμού είναι κρίσιμες για την δημιουργία ενός έγκυρου cryptographic nonce, υπολογίζοντας διαφορετικά hashes. Αυτό βοηθάει στην διαμόρφωση μιας γενικής συναίνεσης στο Blockchain.
- Η χρήση του «hash των δεδομένων» βοηθάει στην αποθήκευση μεγάλου όγκου δεδομένων στο Blockchain. Τα δεδομένα αυτά είναι με χρονοσημανση και μπορούν να κατακερματιστούν για μελλοντική αναφορά. Αυτό κάνει τη μόνιμη αποθήκευση δεδομένων λιγότερο ογκώδη και σίγουρα οικονομικότερη. (Hashing in Blockchain explained | Online Hash Crack, n.d.)

Επομένως, η αλγόριθμοι κρυπτογράφησης είναι ένα εσωτερικό και βασικό μέρος της τεχνολογίας Blockchain. Αυτό είναι ουσιαστικά ένα χαρακτηριστικό που καθιστά τις συναλλαγές αμετάβλητες, δίνοντας τους ικανότητες ασφάλειας. Ο κατακερματισμός είναι επίσης το κέντρο των «Merkle Trees», το οποίο είναι μια προηγμένη προσέγγιση του Blockchain κατακερματισμού. (Hashing in Blockchain explained | Online Hash Crack, n.d.)

3.2.3 Μηχανισμοί Συναίνεσης (Consensus Mechanism)

Οι μηχανισμοί συναίνεσης είναι πρωτόκολλα που διασφαλίζουν ότι όλοι οι κόμβοι (δηλαδή οι συσκευές που διατηρούν το Blockchain και (μερικές φορές) επεξεργάζονται τις συναλλαγές) συγχρονίζονται μεταξύ τους και συμφωνούν ποιες συναλλαγές είναι νόμιμες ώστε να προστεθούν στο Blockchain.

Αυτοί οι μηχανισμοί συναίνεσης είναι κρίσιμοι για το Blockchain για τη σωστή λειτουργία του. Είναι αυτοί που πιστοποιούν ότι όλοι χρησιμοποιούν το ίδιο Blockchain. Όλοι μπορούν να καταχωρίσουν δεδομένα για να προστεθούν στο Blockchain, έτσι είναι απαραίτητο όλες οι συναλλαγές να είναι συνεχώς ελεγμένες και το Blockchain να ελέγχεται συνεχώς από όλους τους κόμβους. Χωρίς τους μηχανισμούς συναίνεσης, τα Blockchains κινδυνεύουν από διάφορες επιθέσεις. (Different Blockchain Consensus Mechanisms - By Blockgenic, n.d.)

Οι πιο δημοφιλείς μηχανισμοί συναίνεσης είναι οι εξής:

3.2.3.1 Proof Of Work (POW)

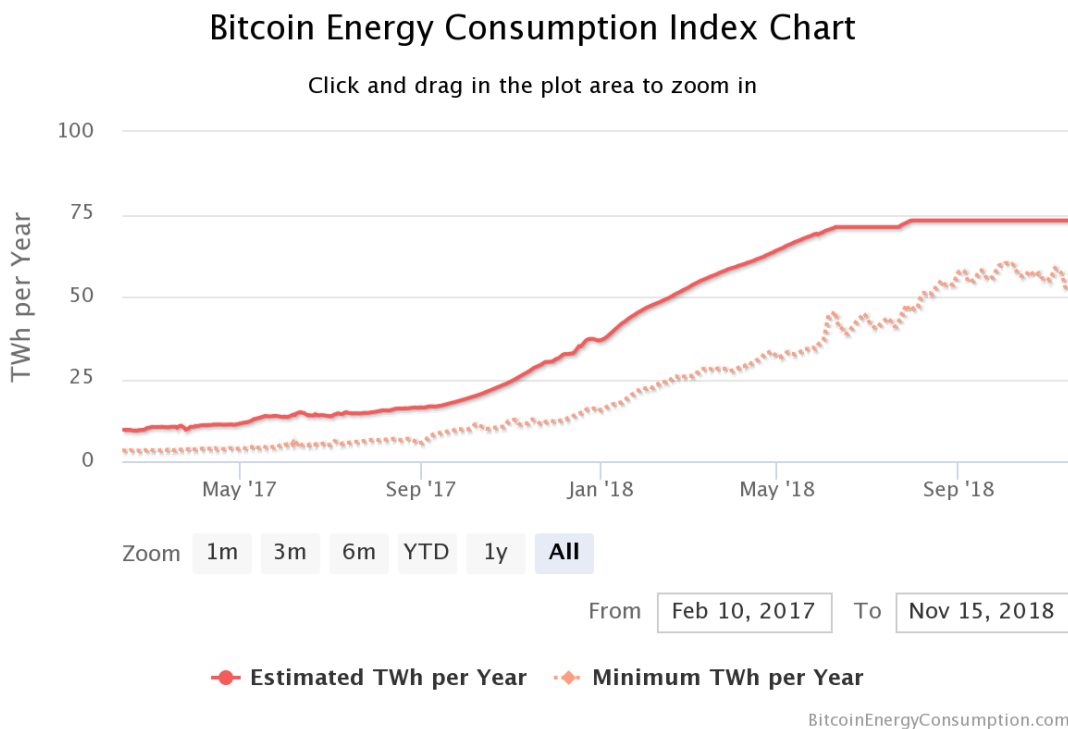
Ο Proof Of Work είναι ο πρώτος μηχανισμός συναίνεσης Blockchain, τον οποίο χρησιμοποίησε πρώτο το Bitcoin. Πολλά κρυπτονομίσματα έχουν ακολουθήσει το παράδειγμα του Bitcoin και έχουν επίσης υιοθετήσει αυτόν τον μηχανισμό συναίνεσης.

Η Proof Of Work διαδικασία είναι γνωστή ως «mining» (είναι η διαδικασία απόκτησης κρυπτονομισμάτων για την προσθήκη ενός μπλοκ στο Blockchain) και οι κόμβοι που συμμετέχουν είναι γνωστοί ως «miners». Οι «miners» λύνουν περίπλοκα μαθηματικά προβλήματα τα οποία απαιτούν μεγάλη υπολογιστική ισχύ (Different Blockchain Consensus Mechanisms - By Blockgenic, n.d.). Ένας miner για να μπορέσει να προσθέσει ένα μπλοκ στο Blockchain, χρειάζεται πρώτα να έχει λύσει ένα σύνθετο κρυπτογραφικό πρόβλημα. Αυτά τα κρυπτογραφικά παζλ έχουν κάποιες ενδιαφέρουσες ιδιότητες οι οποίες θα παρουσιαστούν στην επόμενη παράγραφο.

Η διαδικασία εύρεσης λύσης του προβλήματος είναι εξαιρετικά δύσκολη. Πρώτα από όλα, αυτά τα μαθηματικά προβλήματα είναι ασύμμετρα, δηλαδή απαιτείται αρκετός χρόνος για να βρεθεί η απάντηση, ακόμη κι αν είναι εύκολο να επιβεβαιωθεί ότι μια απάντηση είναι σωστή. Δεύτερον, ο μόνος τρόπος για να λυθούν τέτοια παζλ είναι να γίνει «υπόθεση» της απάντησης. Δεν είναι δυνατό να λυθεί το παζλ γρηγορότερα χρησιμοποιώντας οποιαδήποτε άλλη μέθοδο από την μέθοδο δοκιμής και λάθους. Αυτό σημαίνει ότι εάν κάποιος ήθελε να βρει την λύση του παζλ γρηγορότερα, θα χρειαζόταν περισσότερη υπολογιστική δύναμη, η οποία είναι πολύ ακριβή. Τέλος η δυσκολία αυτών των προβλημάτων αλλάζει ανάλογα με το πόσο γρήγορα τα blocks δημιουργούνται-εξορύσσονται. Για να διατηρηθεί μια σταθερή προμήθεια νέων κερμάτων, τα μπλοκ πρέπει να δημιουργούνται εντός ενός συγκεκριμένου χρονικού πλαισίου. Εάν τα μπλοκ δημιουργούνται πολύ

γρήγορα, τότε τα παζλ γίνονται δυσκολότερα και εάν αυτά δημιουργούνται πολύ αργά, τότε τα παζλ γίνονται ευκολότερα. (Different Blockchain Consensus Mechanisms - By Blockgenic, n.d.)

Υπάρχει όμως ένα σημαντικό μειονέκτημα αυτού του μηχανισμού συναίνεσης. Το Proof Of Work χρησιμοποιεί πολλούς πόρους για αυτό το λόγο είναι μη βιώσιμο για το μέλλον. Αυτή είναι και η αιτία που μερικά Blockchains κινούνται σε διαφορετικούς μηχανισμούς συναίνεσης. Όπως φαίνεται και στην παρακάτω εικόνα, η ενέργεια που καταναλώνει το POW εκτιμάται ότι είναι περίπου στα 73,12 TWh, με εκτιμώμενο κόστος 3.656.073.069 δολλάρια.



Εικόνα 12. Η κατανάλωση ενέργειας του Bitcoin

3.2.3.2 Proof Of Stake (POS)

Το Proof Of Stake είναι πιο φιλικό προς το περιβάλλον συγκριτικά με το Proof Of Work πρωτόκολλο. Η διασφάλιση της συνέπειας στο Proof of Stake βασίζεται στην οικονομική κατάσταση ενός κόμβου του Blockchain δικτύου. Το Proof Of Stake χρησιμοποιεί την προϋπόθεση ότι όσοι κατέχουν τα περισσότερα νομίσματα σε ένα δίκτυο έχουν συμφέρον να διατηρήσουν το δίκτυο έτσι ώστε η αξία των κερμάτων να είναι υψηλά. (Different Blockchain Consensus Mechanisms - By Blockgenic, n.d.)

Σε ένα σύστημα που χρησιμοποιεί το Proof Of Stake, μια τυχαία διαδικασία προσδιορίζει ποιος πρόκειται να παράγει το επόμενο μπλοκ. Όλοι οι κόμβοι ή αλλιώς «επικυρωτές» (validators) κατέχουν ένα σύνολο ψηφιακών κερμάτων (bitcoins) και ανάλογα με την ποσότητα κερμάτων (stake), που απαιτείται να έχουν για συγκεκριμένο χρονικό διάστημα, μπορούν να συμμετέχουν στη διαδικασία

επικύρωσης. Με άλλα λόγια οι πιθανότητες εκλογής ενός κόμβου ως «επικυρωτή» του επόμενου μπλοκ είναι ανάλογες με το ποσό πονταρίσματος που έχει διαθέσει. Έτσι, οι κόμβοι πάλι έχουν να λύσουν ένα κρυπτογραφικό πρόβλημα, αλλά το κλειδί στη λύση είναι το ποσό του στοιχήματος που έχει διαθέσει ο κόμβος και το χρονικό διάστημα που έχει παρέλθει από το ποντάρισμα. Οι χρήστες μπορούν να στοιχηματίσουν τα tokens τους για να γίνουν ο «επικυρωτής» (δηλαδή κάποιος που μπορεί να παράγει μπλοκ), το οποίο σημαίνει ότι κλειδώνουν τα tokens τους για ένα συγκεκριμένο χρονικό διάστημα. Μετά από αυτό, δικαιούνται να παράγουν μπλοκ. Η διαδικασία που αποφασίζει ποιος πρόκειται να παράγει το επόμενο block, λαμβάνει υπόψιν της τους παράγοντες που εξαρτώνται από τον σχεδιασμό του Blockchain, αλλά και το άτομο το οποίο έχει το μεγαλύτερο μερίδιο και άρα έχει και την υψηλότερη πιθανότητα να παράγει το επόμενο μπλοκ. Ένας άλλος παράγοντας που μπορεί να ληφθεί υπόψιν είναι πόσο καιρό έχουν στοιχηματιστεί τα κέρματα. (Different Blockchain Consensus Mechanisms - By Blockgenic, n.d.)

Οι «επικυρωτές» επίσης επιβραβεύονται για τη δουλειά τους. Το βραβείο που λαμβάνουν για την δημιουργία του επόμενου μπλοκ εξαρτάται από τον σχεδιασμό του Blockchain. Συνήθως, είτε λαμβάνουν όλα ή μέρος των τελών συναλλαγής από όλες τις συναλλαγές μέσα στο μπλοκ που δημιουργούν ή λαμβάνουν ένα σταθερό ποσό νομισμάτων.

Το Proof Of Stake δεν είναι μόνο πολύ πιο ενεργειακά αποδοτικό από το Proof Of Work, αλλά έχει και μια σημαντική διαφορά. Σε ένα σύστημα Proof Of Work, ο «miner» δεν μπορεί να κατέχει κανένα από τα νομίσματα που εξορύσσονται, πράγμα που σημαίνει ότι επιδιώκουν μόνο να μεγιστοποιήσουν τα κέρδη τους χωρίς να βελτιώσουν πραγματικά το δίκτυο. Σε ένα σύστημα Proof Of Stake, οι «επικυρωτές» του συστήματος έχουν πολύ μεγαλύτερο κίνητρο για να διατηρήσουν το δίκτυο καθώς κρατούν πραγματικά τα νομίσματα τους. (Different Blockchain Consensus Mechanisms - By Blockgenic, n.d.)

3.2.3.3 Delegated Proof Of Stake (DPOS)

Το Delegated Proof Of Stake (DPOS) είναι ένας πολύ γρήγορος μηχανισμός συναίνεσης και γνωστός για την εφαρμογή του στην EOS. Συχνά αναφέρεται ως ψηφιακή δημοκρατία, χάρη στο σταθμισμένο σύστημα ψηφοφορίας. Στο DPOS σύστημα, οι χρήστες μπορούν να στοιχηματίσουν τα νομίσματα τους για να ψηφίσουν για ένα συγκεκριμένο αριθμό αντιπροσώπων. Το βάρος της ψήφου τους εξαρτάται από το ποσό στοιχήματος τους. Ένας αντιπρόσωπος είναι ένα άτομο ή οργάνωση η οποία θέλει να παράγει μπλοκ στο δίκτυο. Οι αντιπρόσωποι που λαμβάνουν τους περισσότερους ψήφους, ξεκινούν να παράγουν μπλοκ και επιβραβεύονται για την δημιουργία αυτών. Όπως ακριβώς συμβαίνει στο POW, αυτοί είτε πληρώνονται από τα τέλη συναλλαγής ή τους δίνεται ένα συγκεκριμένο ποσό νομισμάτων, που δημιουργείται από τον πληθωρισμό. Ο αριθμός των αντιπροσώπων που θα ξεκινήσουν να παράγουν μπλοκ, εξαρτάται από τον σχεδιασμό του Blockchain. Δεδομένου ότι οι εκπρόσωποι θέλουν να λάβουν όσο το δυνατόν περισσότερες ψήφους, έχουν συνεχώς κίνητρο να δημιουργήσουν νέα

πράγματα για την κοινότητα. Τέλος, το Delegated Proof Of Work πρωτόκολλο είναι πολύ πιο αποδοτικό στην επεξεργασία συναλλαγών από τα προαναφερθέντα πρωτόκολλα. (Different Blockchain Consensus Mechanisms - By Blockgenic, n.d.)

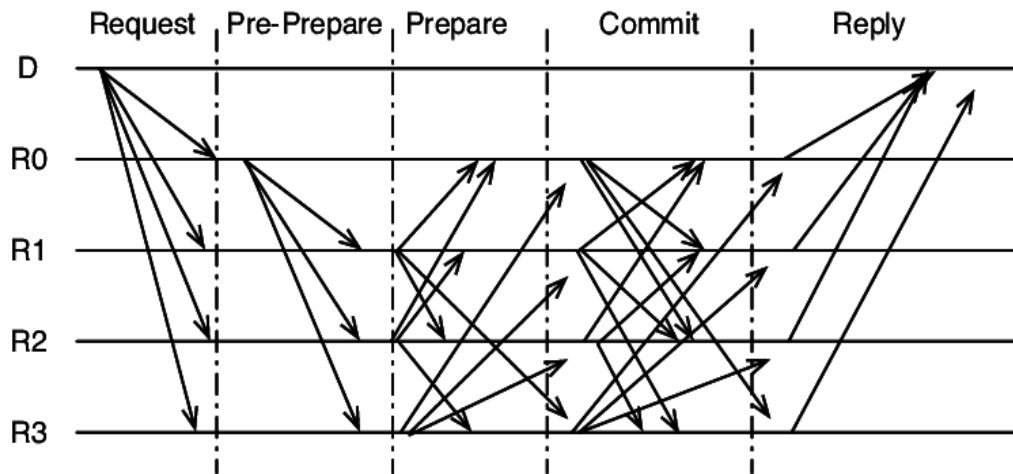
Practical Byzantine Fault Tolerance

Αυτός ο αλγόριθμος Practical Byzantine Fault Tolerance προτάθηκε από τους Miguel Castro και Barbara Liskov για τον μετριασμό του Byzantine Fault που συμβαίνει λόγω μιας αποτυχίας συναίνεσης που οδηγεί στο Byzantine Generals Problem (BGP). Το γενικό πρόβλημα των Blockchain δικτύων είναι η επίτευξη συμφωνίας μέσω μη αξιόπιστων κόμβων. Στο σενάριο αυτό ένας εισβολέας μπορεί να επωφεληθεί, να χειριστεί τους εμπλεκόμενους κόμβους και να επικυρώσει μια ψευδή κακόβουλη συναλλαγή. Το επίκεντρο του PBFT είναι να παράσχει μια πρακτική αναπαράσταση της Byzantine State μηχανής για την ανοχή του Byzantine Fault και μιας ψηφοφορίας με αντίγραφα των συναλλαγών.

Αυτός ο αλγόριθμος βελτιστοποιείται ώστε να λειτουργεί σε ασύγχρονο σύστημα, να είναι υψηλής απόδοσης και να απαιτεί χαμηλό κόστος. Η απόφαση συναίνεσης επιτυγχάνεται μετά από μια σειρά φάσεων. Η κύρια λειτουργία είναι η επιλογή ενός κόμβου ως πρωτεύον και οι υπόλοιποι αναλαμβάνουν να δημιουργούν αντίγραφα ασφαλείας των συναλλαγών. Ο αλγόριθμος PBFT επεξεργάζεται τις αιτήσεις συναλλαγών με υψηλή ταχύτητα, αλλά απαιτεί $3n+1$ αντίγραφα των συναλλαγών για να αντιμετωπίσει η αποτυχημένους κόμβους (Baliga, 2017). Επιπλέον, η ταυτότητα του συστήματος είναι πιθανόν να διακυβεύεται εξαιτίας του τρόπου με τον οποίο σχεδιάστηκε ο αλγόριθμος. Είναι επίσης δύσκολο να χρησιμοποιηθεί αυτός ο αλγόριθμος στο δημόσιο περιβάλλον, καθώς είναι υποχρεωτικός ο προσδιορισμός του πλήθους των κόμβων προτού αρχίσει η λειτουργία του δικτύου και εκτέλεσης του αλγορίθμου. (S. J. Alsunaidi, 2019)

Ο PBFT αλγόριθμος περιέχει συγκεκριμένα πέντε φάσεις request, pre-prepare, prepare, commit και reply. Στην Εικόνα 13 περιγράφεται ο τρόπος λειτουργίας του PBFT. Στην πρώτη φάση ο πελάτης στέλνει την συναλλαγή στον πρωτεύον κόμβο, ο οποίος επιβεβαιώνει χρονολογικά το αίτημα αυτό και στη συνέχεια προωθεί το μήνυμα στους άλλους τρεις κόμβους. Για τη συμμετοχή του αιτήματος στα στάδια αυτά οι υπόλοιποι κόμβοι πρέπει να συγκεντρώσουν την αποδοχή των ψήφων των $2/3$ του υπόλοιπου δικτύου, ενώ ο πρωτεύον κόμβος συντονίζει τις ενέργειες της διαδικασίας. Στην περίπτωση που ο τρίτος κόμβος συντριβεί, δεν δημιουργείται πρόβλημα καθώς τα $2/3$ του δικτύου συνεχίζουν να λειτουργούν και το μήνυμα περνάει από τις πέντε φάσεις για να επιτευχθεί συναίνεση μεταξύ των υπόλοιπων κόμβων. Τελικά, αυτοί οι κόμβοι απαντούν στον πελάτη για να ολοκληρωθεί ένας γύρος της consensus διαδικασίας. Το PBFT εγγυάται ότι οι κόμβοι διατηρούν μια κοινή κατάσταση-απόφαση και αναλαμβάνουν μια συγκεκριμένη συμπεριφορά σε κάθε γύρο της συμφωνίας. Το PBFT επιτυγχάνει το στόχο της ισχυρής συνέπειας, έτσι θεωρείται ως ένα absolute-finality consensus πρωτόκολλο (Y. Xiao, 2019). Ωστόσο, επειδή ο αλγόριθμος PBFT βασίζεται σε μεγάλο βαθμό στις παραδοχές του συγχρονισμού, αυτό του προκαλεί μια σημαντική αδυναμία. Ένας αντίπαλος μπορεί

να επιτεθεί στον PBFT, με αποτέλεσμα να τον αναγκάσει να σταματήσει μια διαδικασία για συναίνεση ή να επιβραδύνει σημαντικά το πρωτόκολλο.

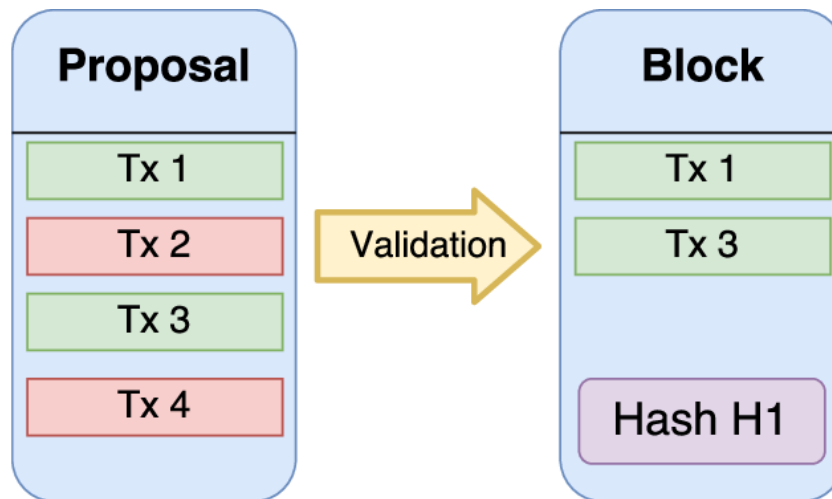


Εικόνα 13. Η διαδικασία του PBFT

3.2.3.4 Yet Another Consensus (YAC)

Παρόλο που και ο Yet Another Consensus βασίζεται στην παραδοχή του χρονισμού, τα προβλήματα του PBFT αποφεύγονται, χρησιμοποιώντας μια δυναμική λίστα ομότιμων κόμβων. Ο YAC είναι ένας πρακτικός αποκεντρωμένος αλγόριθμος συναίνεσης, ο οποίος λύνει τα προβλήματα της μη αποδοτικότητας της μετάδοσης των μηνυμάτων που υπάρχουν στον κλασικό Byzantine Fault Tolerant αλγόριθμο συναίνεσης. Ο αλγόριθμος YAC είναι ανοιχτός κώδικας και χρησιμοποιείται στο Hyperledger Iroha Blockchain.

Στη συνέχεια θα περιγραφεί ένα παράδειγμα συναίνεσης του αλγόριθμου YAC. Το πρώτο βήμα είναι κάθε πελάτης να στείλει τις δικές του συναλλαγές στην υπηρεσία διάταξης (Ordering Service - OS). Η OS θα πρέπει να συλλέξει όλες τις συναλλαγές, να τις ταξινομήσει και να δημιουργήσει ένα μπλόκ πρότασης (proposal). Έπειτα η OS θα στείλει αυτό το «proposal» σε κάθε ομότιμο κόμβο (peer) του δικτύου. Το καθήκον του «επικυρωτή» (validator) είναι να επικυρώσει αυτό το «proposal» και να προσπαθήσει να εφαρμόσει κάθε συναλλαγή που εμπεριέχεται στο «proposal» στην δική της τοπική κατάσταση. Μια συναλλαγή θεωρείται έγκυρη εάν συμφωνεί με τους κανόνες επαλήθευσης και η εφαρμογή του δεν παραβιάζει τους κανόνες σχετικά με την παγκόσμια κατάσταση. Όπως για παράδειγμα, κανένας λογαριασμός δεν μπορεί να έχει αρνητικό υπόλοιπο. Ο «επικυρωτής» οφείλει να δημιουργήσει ένα μπλοκ από όλες τις έγκυρες συναλλαγές και να υπολογίσει το «hash» αυτού του μπλοκ. Η διαδικασία της επαλήθευσης παρουσιάζεται στην Εικόνα 14. Ο «επικυρωτής», έχοντας την αρχική σειρά των κόμβων, υπολογίζει τον επόμενο κόμβο για τον τρέχον γύρο και δημιουργεί μια ψήφο και μεταδίδει στον επόμενο κόμβο αυτήν την ψήφο. Μετά από την αποστολή της ψήφου, ενημερώνει την τοπική του κατάσταση και περιμένει μέχρι κάποιο συγκεκριμένο χρονικό διάστημα για ένα μήνυμα κατοχύρωσης.



Εικόνα 14. Η διαδικασία επαλήθευσης

Ο επόμενος κόμβος λαμβάνει την ψήφο του «επικυρωτή» και υπολογίζει το «hash» του «proposal» (θα θεωρηθεί ότι είναι ίδιο με αυτό που υπολόγισε ο «επικυρωτής»). Επομένως, ο κόμβος αυτός, έχοντας την ίδια αρχική σειρά των κόμβων με τον «επικυρωτή», θα προωθήσει την ψήφο του στον εαυτό του. Οι υπόλοιποι κόμβοι του δικτύου, οι οποίοι θα υπολογίσουν, επίσης, το ίδιο hash και άρα και την ίδια αρχική σειρά κόμβων, θα προωθήσουν τις ψήφους τους στον ίδιο επόμενο κόμβο. Ο επόμενος αυτός κόμβος αφού λάβει τις ψήφους των άλλων κόμβων και εφόσον αυτοί είναι περισσότεροι από το 2/3 των συνολικών κόμβων, θα στείλει σε όλους τους κόμβους ένα μήνυμα κατοχύρωσης με όλες τις λαμβανόμενες ψήφους των κόμβων. Κάθε κόμβος που θα λάβει αυτό το μήνυμα, θα πρέπει να προσθέσει υπογραφές στο block με το hash αυτό και να ενημερώσει την τοπική του κατάσταση. Εάν κάποιος από τους κόμβους δεν λάβει το μήνυμα κατοχύρωσης και έχει χάσει τον τρέχον και τον προηγούμενο γύρο, τότε θα υπολογίσει ένα λανθασμένο «hash» και άρα θα έχει και μια διαφορετική αρχική σειρά των κόμβων, με αποτέλεσμα να στείλει σε άλλον κόμβο την ψήφο του. Αφού αυτός ο κόμβος λάβει την άκυρη ψήφο και δεδομένου ότι αυτός ο κόμβος έχει λάβει το μήνυμα κατοχύρωσης, προωθεί το μήνυμα αυτό πίσω στον μη συγχρονισμένο κόμβο. Λαμβάνοντας το μήνυμα κατοχύρωσης, ο κόμβος το εφαρμόζει και έχει πλέον την ίδια κατάσταση με όλους τους άλλους μετά τον γύρο της συναίνεσης. (Fedor Muraton, 2018)

Χρησιμοποιώντας ψηφοφορίες στα block proposals, το YAC είναι σε θέση να εγγυηθεί την ακεραιότητα και την ασφάλεια της επεξεργασίας των συναλλαγών. Καθώς οι λανθασμένες επικυρώσεις ομότιμων κόμβων (peers) δεν μπορούν να είναι περισσότερες από f από τους τουλάχιστον $3f+1$ ομότιμους κόμβους (peers) του δικτύου. Ο αλγόριθμος YAC μπορεί να κλιμακωθεί σε πληθώρα επικυρωμένων ομότιμων κόμβων, ωστόσο η καθυστέρηση (στο βήμα της ψηφοφορίας) πρέπει να προσαρμοστεί σε σχέση με τον αριθμό των επικυρωμένων ομότιμων κόμβων (peers), έτσι ώστε να μειωθούν τα εμφανιζόμενα σφάλματα από τους ομότιμους κόμβους (peers). (Fedor Muraton, 2018)

3.2.4 Περιβάλλον Εργασίας Blockchain

Η βιβλιοθήκη που χρησιμοποιήθηκε για την υλοποίηση του Blockchain στην παρούσα διπλωματική, είναι η Iroha Hyperledger. Ο λόγος αυτής της επιλογής είναι η δυνατότητα του να ενσωματώνεται εύκολα σε οποιοδήποτε project. Η βιβλιοθήκη Iroha Hyperledger παρέχει ενσωματωμένες εντολές για την διαχείριση του Blockchain, πράγμα που την καθιστά εξαιρετικά εύχρηστη. Οι βασικές έννοιες της βιβλιοθήκης Iroha Hyperledger αναλύονται εκτενέστερα στις επόμενες υποενότητες.

3.2.4.1 *Hyperledger*

Το Hyperledger είναι ένας συνδεδετικός χώρος για την ανάπτυξη ανοικτών βιομηχανικών blocks. Στην επίσημη ιστοσελίδα του Hyperledger δηλώνεται ότι: «Το Hyperledger είναι μια συνεργατική προσπάθεια ανοιχτής πηγής που δημιουργήθηκε για να προωθήσει τις διεπαγγελματικές τεχνολογίες Blockchain. Πρόκειται για μια παγκόσμια συνεργασία, που φιλοξενείται από το ίδρυμα του Linux Foundation, συμπεριλαμβανομένων των ηγετών στον τομέα της χρηματοδότησης, της τραπεζικής, του Διαδικτύου των πραγμάτων, των αλυσίδων εφοδιασμού, της κατασκευής και της τεχνολογίας». Το Hyperledger δεν υποστηρίζει Bitcoin ή οποιαδήποτε άλλη κρυπτογράφηση. Έχει τη δυνατότητα να «οικοδομήσει» μια νέα γενιά εφαρμογών που δημιουργούν την εμπιστοσύνη, τη λογοδοσία και τη διαφάνεια στον πυρήνα τους, ενώ παράλληλα η τεχνολογία αυτή υποσχέθηκε μια ευρύτερη και πιο θεμελιώδη επανάσταση από την τεχνολογία Blockchain, δηλαδή την ροή των επιχειρηματικών διαδικασιών και των νομικών περιορισμών (Rosic, 2017).

3.2.4.2 *Hyperledger Iroha*

Το Hyperledger Iroha έχει σχεδιαστεί για να είναι απλό και εύκολο ώστε να μπορεί να ενσωματωθεί σε έργα και IoT που απαιτούν κατανομημένη ledger τεχνολογία. Το Hyperledger Iroha διαθέτει μια απλή κατασκευή, δομημένη και σχεδιασμένη σε C++. Επικεντρώνεται στην ανάπτυξη εφαρμογών από την μεριά του πελάτη (client), χρησιμοποιώντας ένα νέο αλγόριθμο συναίνεσης που είναι ανθεκτικός σε σφάλματα και ονομάζεται YAC.

Επίσης, οι ενσωματωμένες εντολές της Iroha αποτελούν το σημαντικότερο πλεονέκτημα σε σύγκριση με άλλες πλατφόρμες, δεδομένου ότι είναι πολύ εύκολο να εκτελεστούν απλές λειτουργίες, όπως η δημιουργία ψηφιακών νομισμάτων, η δημιουργία νέων λογαριασμών και η συναλλαγή νομισμάτων μεταξύ των λογαριασμών/χρηστών. Επιπλέον, η βιβλιοθήκη Iroha περιορίζει τον κίνδυνο επίθεσης και άρα βελτιώνει τη συνολική ασφάλεια του συστήματος, καθώς είναι λιγότερες οι συναλλαγές που αποτυγχάνουν. Τέλος, το Hyperledger Iroha είναι το μοναδικό «ledger» που διαθέτει τις εντολές και τα ερωτήματα ενσωματωμένα.

Είναι επίσης σημαντικό να αναφερθούν και κάποιες βασικές έννοιες που προσδιορίζουν το Iroha και είναι απαραίτητες για την κατανόηση της λειτουργίας του.

Λογαριασμός (Account)

Αυτή η οντότητα έχει την δυνατότητα να εκτελέσει συγκεκριμένο σύνολο ενεργειών. Κάθε λογαριασμός ανήκει σε ένα υπάρχων τομέα (domain). Ένας λογαριασμός έχει έναν αριθμό κανόνων, οι οποίοι προσδιορίζουν τις ενέργειες που επιτρέπεται να εφαρμόσει ο χρήστης. (Hyperledger Iroha documentation, n.d.)

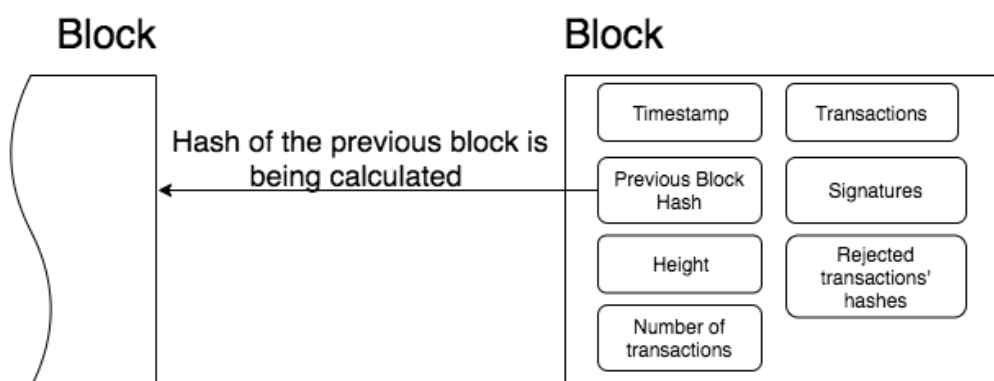
Περιουσιακό Στοιχείο (Asset)

Αποτελεί οποιαδήποτε μετρήσιμη αξία ή αγαθό. Κάθε «asset» ανήκει σε έναν από τους υπάρχοντες τομείς (domains). (Hyperledger Iroha documentation, n.d.)

Μπλοκ

Τα δεδομένα συναλλαγής είναι μόνιμα καταγεγραμμένα σε αρχεία, τα οποία ονομάζονται μπλοκ. Τα μπλοκ είναι οργανωμένα σε μια γραμμική ακολουθία με το ν χρόνο (επίσης γνωστή ως Blockchain). Τα μπλοκ είναι υπογεγραμμένα και πιστοποιημένα μέσω κρυπτογραφικών υπογραφών των Iroha ομότιμων κόμβων (peers), ψηφίζοντας για αυτό το μπλοκ κατά την διάρκεια της συναίνεσης (consensus). Το πιστοποιημένο block ονομάζεται «payload» και η διαδικασία δημιουργίας του φαίνεται στην παρακάτω εικόνα.

Τα μπλοκ υπογράφονται με τις κρυπτογραφικές υπογραφές των ομότιμων κόμβων Iroha (peers), ψηφίζοντας για αυτό το μπλοκ κατά τη συναίνεση (consensus). Το περιεχόμενο του μπλοκ ονομάζεται «payload», οπότε η δομή ενός μπλοκ φαίνεται στην παρακάτω εικόνα. (Hyperledger Iroha documentation, n.d.)



Εικόνα 15. Η δομή ενός block

Πελάτης (Client)

Οποιαδήποτε εφαρμογή χρησιμοποιεί το Iroha λειτουργεί ως πελάτης (client). Ένα χαρακτηριστικό του Iroha είναι ότι όλοι οι πελάτες χρησιμοποιούν απλές client-server δομές όταν αλληλοεπιδρούν με ένα ομότιμο κόμβο (peer) του δικτύου. Αυτοί δεν χρησιμοποιούν καμία περίπλοκη δομή που σχετίζεται με Blockchain συστήματα. Για παράδειγμα στο Bitcoin, οι πελάτες πρέπει να επικυρώνουν τα μπλοκ, ενώ στο Iroha, ο πελάτης επικοινωνεί με άλλους ομότιμους κόμβους (peers) όπως με έναν single server. (Hyperledger Iroha documentation, n.d.)

Εντολή

Η εντολή είναι ένας τρόπος να αλλάξει η κατάσταση του δικτύου. Για παράδειγμα, για να δημιουργηθεί ένας νέος ρόλος στο Iroha, θα πρέπει να εκτελεστεί η εντολή Δημιουργίας Ρόλου. (Hyperledger Iroha documentation, n.d.)

Συναίνεση (Consensus)

Ο αλγόριθμος συναίνεσης είναι μια διαδικασία στην επιστήμη των υπολογιστών που χρησιμοποιείται για να επιτευχθεί συμφωνία για μια ενιαία τιμή δεδομένων μεταξύ των κατανεμημένων διαδικασιών ή συστημάτων. Όπως έχει ήδη αναφερθεί, οι αλγόριθμοι συναίνεσης έχουν σχεδιαστεί για να παρέχουν αξιοπιστία σε ένα δίκτυο που περιλαμβάνει πολλούς αναξιόπιστους κόμβους. Η επίλυση αυτού του ζητήματος, γνωστό και ως πρόβλημα συναίνεσης, είναι σημαντικό για συστήματα κατανεμημένων υπολογιστών.

Η συναίνεση είναι η κύριο τμήμα του Blockchain, καθώς διατηρεί μια σταθερή κατάσταση μεταξύ των κόμβων (peers) μέσα σε ένα ομότιμο δίκτυο. Το Iroha χρησιμοποιεί έναν δικό του αλγόριθμο συναίνεσης που ονομάζεται Yet Another Consensus (δηλαδή YAC). Τα ξεχωριστά χαρακτηριστικά του YAC αλγορίθμου είναι η δυνατότητα κλιμάκωσης, η αποδοτικότητα και η ανοχή του σε σφάλματα. (Hyperledger Iroha documentation, n.d.)

Τομέας (Domain)

Το «domain» είναι μια οντότητα για την ομαδοποίηση των λογαριασμών και των «assets». (Hyperledger Iroha documentation, n.d.)

Ομότιμος Κόμβος (Peer)

Είναι ένας κόμβος που ανήκει στο Iroha δίκτυο και συμμετέχει στην διαδικασία συναίνεσης. (Hyperledger Iroha documentation, n.d.)

Άδεια (Permission)

Είναι ο κανόνας που δίνει το δικαίωμα σε κάποιον να εκτελέσει μια συγκεκριμένη εντολή/ενέργεια. Η άδεια δεν μπορεί να δοθεί κατευθείαν σε έναν λογαριασμό (εκτός από τις Grantable Permissions), αντίθετα ο λογαριασμός έχει κανόνες, οι οποίοι είναι συλλογές αδειών. (Hyperledger Iroha documentation, n.d.)

Ερώτημα (Query)

Είναι ένα αίτημα προς το Iroha, το οποίο δεν έχει καμία επιρροή στην κατάσταση του δικτύου. Ο πελάτης (client) μπορεί να ζητήσει πληροφορίες και δεδομένα σχετικά με την κατάσταση του συστήματος μέσω των αιτημάτων αυτών, όπως είναι το ιστορικό των συναλλαγών κ.λπ. (Hyperledger Iroha documentation, n.d.)

Κανόνας (Role)

Είναι μια οντότητα που περιέχει ένα σύνολο από άδειες/δικαιώματα. (Hyperledger Iroha documentation, n.d.)

Υπογράφων (Signatory)

Αντιπροσωπεύει μια οντότητα η οποία μπορεί να επιβεβαιώσει τις συναλλαγές για έναν λογαριασμό. (Hyperledger Iroha documentation, n.d.)

Συναλλαγή (Transaction)

Είναι ένα ταξινομημένο σύνολο εντολών, το οποίο εφαρμόζεται στο Blockchain. Οποιαδήποτε μη έγκυρη εντολή μέσα σε μια συναλλαγή οδηγεί στην απόρριψη ολόκληρης της συναλλαγής κατά την διάρκεια της διαδικασίας επικύρωσης. (Hyperledger Iroha documentation, n.d.)

3.3 Framework

3.3.1 Django

Η επιλογή ενός framework για την ανάπτυξη μιας εφαρμογής είναι μια από τις πιο σημαντικές αποφάσεις, καθώς είναι πολλά τα frameworks στην κοινότητα των διαδικτυακών εφαρμογών. Για πολλές εταιρείες και έργα, η επιλογή του Django Framework είναι μια εύκολη επιλογή καθώς είναι ένα από τα πιο δημοφιλή εργαλεία ανάπτυξης ιστοσελίδων. Ένας από τους λόγους αυτού είναι η επικέντρωση του Django framework σε εφαρμογές βασισμένες στην Python, η οποία είναι και μια από τις κορυφαίες γλώσσες ανάπτυξης ιστοσελίδων. Στην παρούσα διπλωματική, χρησιμοποιήθηκε, επίσης, το Django για την υλοποίηση της βάσης (δηλαδή του Backend) της εφαρμογής καθώς προσφέρει απλότητα, ευελιξία, αξιοπιστία αλλά και δυνατότητα κλιμάκωσης.

Το Django είναι ένα ελεύθερο (free) και ανοικτού κώδικα (open-source) πλαίσιο ιστού που βασίζεται στην Python (Python-based web framework). Το Django είναι κατά βάση ένα πλαίσιο μοντέλου-όψης-ελεγκτή, MVC (Model-ViewController), όπου οι συνιστώσες «όψη» ("view") και «ελεγκτής» ("controller") αντικαθίστανται από τις έννοιες «πρότυπο» ("template") και «όψη» αντίστοιχα. Το Django, δηλαδή, αποτελεί ένα πλαίσιο μοντέλου-προτύπου-όψης, MTV (ModelTemplate-View).

Ο πρωταρχικός στόχος του Django είναι να διευκολύνει τη δημιουργία πολύπλοκων ιστοτόπων που οδηγούνται από βάσεις δεδομένων (database-driven websites). Το πλαίσιο δίνει έμφαση στις δυνατότητες επαναχρησιμοποίησης (reusability) και «προσάρτησης» ("pluggability") συνιστωσών, στη χρήση λιγότερου κώδικα, στη χαμηλή σύζευξη (low coupling), στην ταχεία ανάπτυξη (rapid development) και στην αρχή της μη επαναληψιμότητας (Don't Repeat Yourself, DRY, principle). Η Python χρησιμοποιείται σε όλη την έκταση του πλαισίου, ακόμα και για αρχεία ρυθμίσεων (settings files) και μοντέλα δεδομένων (data models). Το Django παρέχει επίσης μία προαιρετική διαχειριστική διεπαφή (administrative interface) για τη δημιουργία, ανάγνωση, ενημέρωση και διαγραφή (Create, Read, Update and Delete - CRUD) δεδομένων, η οποία δημιουργείται δυναμικά μέσω ενδοσκόπησης (introspection) και διαμορφώνεται (configured) μέσω μοντέλων διαχείρισης (admin models).

Ορισμένοι γνωστοί ιστότοποι που χρησιμοποιούν το Django είναι οι PBS (Public Broadcasting Service), Instagram, Mozilla, The Washington Times, Disqus, Bitbucket και Nextdoor.

3.3.2 React Js

Στην παρούσα διπλωματική, χρησιμοποιήθηκε η React Js για την υλοποίηση του front-end της εφαρμογής. Παρότι υπάρχουν πολλές πλατφόρμες ανοιχτού κώδικα για την ανάπτυξη του frontend μιας διαδικτυακής εφαρμογής, όπως η Angular, τα οφέλη του React έναντι άλλων ανταγωνιστικών τεχνολογιών ή πλαισίων υπερτερούν. Κάποιοι από τους λόγους που επιλέχθηκε η React Js είναι η απλότητα και η αποδοτικότητα που παρέχει. Είναι απαραίτητο, επομένως, να αναφερθούμε εκτενέστερα για το συγκεκριμένο Framework.

Πρόκειται για ένα JavaScript Framework/library που αποσκοπεί στη δημιουργία διεπαφών χρήστη (UI) εφαρμογών και συντηρείται από το Facebook και μια κοινότητα μεμονωμένων προγραμματιστών και εταιριών, το οποίο έγινε διαθέσιμο τον Μάρτιο του 2013. (React (JavaScript library) - Wikipedia, χ.χ.) Η βιβλιοθήκη απαλλάσσει τον προγραμματιστή από την αυστηρή διαχείριση σε χαμηλό επίπεδο αντικειμένων που υποβάλλονται στο DOM, δίνοντάς του ένα απλούστερο αφαιρετικό μοντέλο και δυναμικά υψηλότερες επιδόσεις. Το ReactJs περιστρέφεται γύρω από τη χρήση δηλωτικής λογικής μέσω αντικειμένων τα οποία ονομάζονται Components (Εξαρτήματα). Κάθε Component αποτελεί ένα μικρό κομμάτι της διεπαφής χρήστη υλοποιώντας κάποιες συγκεκριμένες λειτουργίες. Δίνονται παρακάτω κάποιες βασικές έννοιες απαραίτητες για την κατανόηση του πως λειτουργεί μία εφαρμογή ReactJs:

- **JSX:** Σημαίνει JavaScript XML και είναι μία επέκταση της JavaScript που επιτρέπει την περιγραφή των components που θα συγκροτήσουν την εφαρμογή. Η χρήση JSX είναι προαιρετική, καθώς μετά την προ-επεξεργασία της καταλήγει σε κώδικα JavaScript, μπορεί επομένως κάποιος εύκολα να το αποφύγει. Η σύνταξη JSX φαίνεται παρακάτω με ένα απλό παράδειγμα όπου θέλουμε να δηλώσουμε ένα HTML στοιχείο τύπου h1.

```
const element = <Text>Hello, world!</Text>;
```

Εικόνα 1. Η σύνταξη της JSX

Η μεταβλητή element αντιπροσωπεύει ένα component τύπου Text, που χρησιμοποιείται για την προβολή κειμένου. (React - Introducing JSX, n.d.)

- **Props:** Όλα σχεδόν τα components μπορούν να προσαρμοστούν κατά τη δήλωσή τους μέσω παραμέτρων που ονομάζουμε props. Οι παράμετροι αυτοί είναι διαθέσιμοι εντός των components, και μπορούν να χρησιμοποιηθούν για τον έλεγχο της όψης και συμπεριφοράς τους. Ένα component τύπου Image για παράδειγμα χρησιμοποιείται για την προβολή

εικόνας, και κάποια από τα κύρια props του είναι το source, που χρησιμοποιείται για τον εντοπισμό της εικόνας που θα προβληθεί, και το style που χρησιμοποιείται για την οπτική προσαρμογή της εικόνας. Τα props ενός component τίθενται κατά τη δήλωση του, και δεν μπορούν να μεταβληθούν από κώδικα εντός του component. (Props, n.d.)

- **State:** Τα components είναι επίσης stateful, έχουν δηλαδή state (κατάσταση) που συμβολίζεται με ένα JavaScript αντικείμενο. Η κατάσταση ενός component χρησιμοποιείται για τον ίδιο σκοπό που χρησιμοποιούνται τα props, την προσαρμογή του Component. Αντίθετα όμως με τα props, το state τίθεται αρχικά εντός του constructor και μπορεί να μεταβληθεί από τον κώδικα εντός του Component, μέσω μίας μεθόδου που ονομάζεται setState().

Μερικά από τα πλεονεκτήματα της χρήσης ReactJs για την ανάπτυξη εφαρμογών είναι:

- Η χρήση ενός εικονικού DOM που είναι ένα αντικείμενο JavaScript. Αυτό έχει ως επακόλουθο τη βελτίωση της απόδοσης της εφαρμογής, μιας και το εικονικό αυτό DOM είναι πιο γρήγορο από το κανονικό
- Μπορεί να χρησιμοποιηθεί στην πλευρά και του client και του server καθώς και παράλληλα με άλλα framework
- Η χρήση των component και των μοτίβων δεδομένων βελτιώνουν την αναγνωσιμότητα και τη διαφάνεια του κώδικα, τα οποία βοηθάνε στη συντήρηση μεγαλύτερων εφαρμογών

3.3.3 Redux

Τα περισσότερα frameworks όπως React, Angular κ.λπ. είναι κατασκευασμένα με τρόπο ώστε τα «components» να διαχειρίζονται εσωτερικά το «state» τους, χωρίς να χρειάζονται κάποια εξωτερική βιβλιοθήκη ή εργαλείο, πράγμα που είναι χρήσιμο για εφαρμογές με λίγα «Components». Ωστόσο, καθώς η εφαρμογή επεκτείνεται, η διαχείριση των μοιραζόμενων «state» μεταξύ των «components» γίνεται όλο και πιο αναγκαία. Σε μια εφαρμογή όπου τα δεδομένα είναι μοιρασμένα μεταξύ των «components», είναι συγκεχυμένο σε ποιο «component» θα πρέπει να ανήκει κάθε «state». Την λύση σε όλα αυτά τα προβλήματα δίνουν τα εργαλεία διαχείρισης του «state». Στην παρούσα διπλωματική επιλέχτηκε το εργαλείο Redux για την επίλυση αυτών των προβλημάτων.

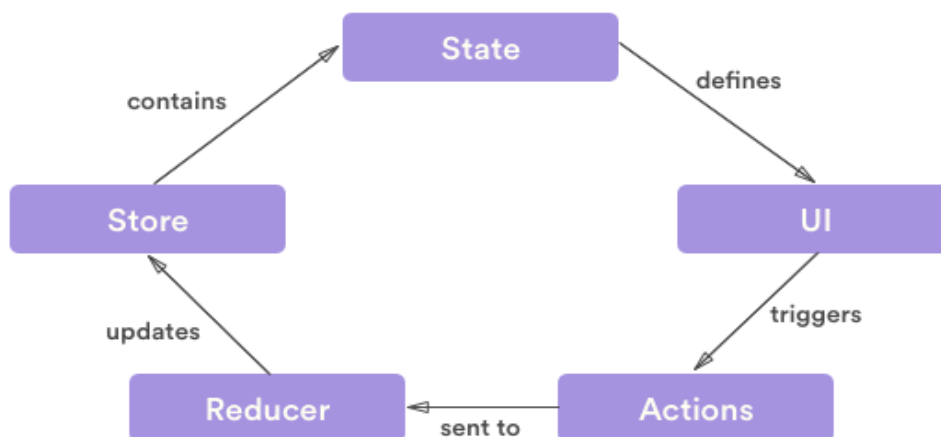
Το Redux (Redux · A Predictable State Container for JS Apps, n.d.) είναι μια βιβλιοθήκη JavaScript ανοιχτού κώδικα για τη διαχείριση της κατάστασης (State) μιας εφαρμογής. Χρησιμοποιείται συνήθως μαζί με βιβλιοθήκες όπως ReactJs ή Angular για το χτίσιμο UIs. Εμπνευσμένη από την αρχιτεκτονική της βιβλιοθήκης Flux του Facebook, φτιάχτηκε από τον Dan Abramov και τον Andrew Clark το 2015. (Redux (JavaScript library) - Wikipedia, n.d.)

Η βιβλιοθήκη έχει τη δική της ορολογία, λογική και συμπεριφορά που ο αναγνώστης δεν απαιτείται να γνωρίζει, οπότε κρίθηκε σκόπιμο να γίνει απλώς μια σύντομη αναφορά στις τέσσερις σημαντικότερες έννοιες που αυτή χρησιμοποιεί:

- **State:** Το State (ή αλλιώς δένδρο state) είναι μια πολύ ευρεία έννοια, αλλά σύμφωνα με το API του Redux συνήθως αναφέρεται στη μοναδική τιμή κατάστασης που διαχειρίζεται από το Store (αναφέρεται παρακάτω) και επιστρέφεται από την κλήση της `getState()`. Αναπαριστά ολόκληρη την κατάσταση μιας εφαρμογής Redux, που συχνά είναι ένα βαθιά εμφωλευμένο αντικείμενο.
- **Action:** Το Action (δράση) είναι ένα απλό αντικείμενο που αντιπροσωπεύει την πρόθεση αλλαγής της κατάστασης. Τα Action είναι ο μόνος τρόπος για να περαστούν τα δεδομένα στο Store. Οποιαδήποτε δεδομένα, είτε προέρχονται από συμβάντα UI, κλήσεις δικτύου είτε άλλες πηγές όπως τα WebSockets, πρέπει τελικά να αποστέλλονται ως Action.
- **Reducer:** Ένας Reducer είναι μια συνάρτηση που δέχεται την τωρινή κατάσταση (State) της εφαρμογής και μια τιμή (από το action) και επιστρέφει ένα νέο State. Το χαρακτηριστικό των Reducer είναι ότι είναι pure συναρτήσεις, δηλαδή για δεδομένες τιμές του τωρινού State και της νέας τιμής εισόδου επιστρέφει δεδομένη έξοδο.
- **Store:** Το Store είναι το object μέσα στο οποίο «ζει» η κατάσταση της εφαρμογής και παρέχει μερικές βοηθητικές μεθόδους για να επιτευχθεί η πρόσβαση στο State, για την αποστολή ενεργειών και για την καταγραφή των listeners. Το σύνολο του State αντιπροσωπεύεται από ένα μόνο Store.

Οποιαδήποτε action επιστρέφει ένα νέο State μέσω των Reducer. Αυτό καθιστά το Redux πολύ απλό και προβλέψιμο.

Στην παρακάτω εικόνα φαίνεται σχηματικά πως αυτά αλληλοεπιδρούν.



Εικόνα 2. Αλληλεπίδραση μεταξύ των εργαλείων του Redux

Κεφάλαιο 4. Ανάλυση Απαιτήσεων Συστήματος

Η εφαρμογή που πραγματεύεται η παρούσα διπλωματική εργασία προορίζεται για χρήση πάνω σε συσκευές που είναι συνδεδεμένες σε δίκτυο. Σκοπός της εφαρμογής είναι να βελτιστοποιήσει τη διαδικασία της ιατρικής παρακολούθησης και της καταγραφής του ιατρικού ιστορικού και να προσφέρει μια ολοκληρωμένη εμπειρία στους χρήστες της. Αυτό σημαίνει ότι ο χρήστης θα μπορεί να δημιουργήσει τον προσωπικό του λογαριασμό, ώστε να μοιράζεται με τους ιατρούς του ή συγγενικά του πρόσωπα το ιστορικό υγείας του. Ο χρήστης θα μπορεί να περιηγηθεί στην κεντρική διεπιφάνεια της εφαρμογής ώστε να δει τις αναρτήσεις του, να αναρτήσει νέες δημοσιεύσεις ή σχόλια σχετικά με την υγεία του και να αλληλοεπιδράσει με άλλους χρήστες. Η εφαρμογή θα βοηθάει στο συντονισμό όλων των ατόμων που φροντίζουν τον χρήστη, είτε αυτοί είναι οι ιατροί, είτε είναι οι συγγενείς. Το παρόν κεφάλαιο καταπιάνεται με την ανάλυση των απαιτήσεων του συστήματος. Θα γίνει μια εκτενής επεξήγηση κάθε λειτουργικότητας της εφαρμογής και θα παρουσιαστούν οι βασικές αρχές που τις διέπουν.

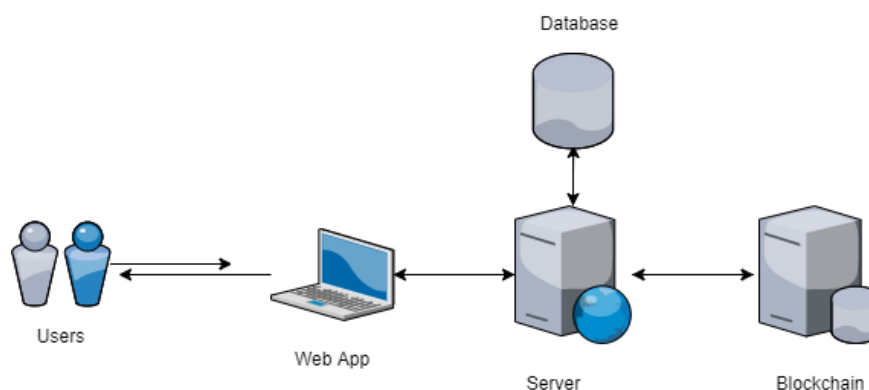
4.1 Γενική Περιγραφή

Η εφαρμογή αποτελείται από πέντε μέρη: (α) τον client, (β) τον server, (γ) τη βάση δεδομένων, (δ) τον server του Blockchain και (ε) την βάση δεδομένων του Blockchain. Ο client είναι η διεπιφάνεια όπου δρα ο χρήστης, ενώ ο server αποτελείται από το σύνολο των μεθόδων υπεύθυνων για την εκτέλεση αιτημάτων από τον client (βλ. Εικόνα 16). Ο client αποτελείται από ένα δρομολογητή (router) που συνίσταται από τις διεπιφάνειες της εφαρμογής με τις οποίες μπορεί να αλληλεπιδράσει ο χρήστης. Η κεντρική διεπιφάνεια αποτελείται από το χρονολόγιο του χρήστη. Ο router απαρτίζεται ακόμη από τις σελίδες Profile, Health Score, Calendar και Community. Καθεμία από αυτές θα αναλυθεί σε επόμενη ενότητα. Η υλοποίηση και ο έλεγχος του client έγιναν με τη βοήθεια φυσικής συσκευής με λειτουργικό Ubuntu 16.04.

Ο server καθορίζει τις λειτουργίες προς εκτέλεση που αντιστοιχούν στις αιτήσεις που δημιουργεί ο χρήστης μέσω του UI της εφαρμογής. Οι λειτουργίες αυτές συνοπτικά είναι η αποθήκευση των διαπιστευτηρίων και των προσωπικών πληροφοριών του χρήστη, η δημιουργία λογαριασμού, η δημιουργία και επεξεργασία δημοσιεύσεων, η δημιουργία και επεξεργασία σχολίων, η δημιουργία και αφαίρεση της αντίδρασης «Like», η λήψη και η αποστολή αιτημάτων φιλίας κλπ. Η κατασκευή του server έχει δομή τέτοια ώστε να αποσκοπεί στη συνεργατική δράση του με τον client, με αποδοτικό και αποτελεσματικό τρόπο.

Η εφαρμογή θα διαθέτει επίσης μια βάση δεδομένων για την αποθήκευση των χρηστών, των δημοσιεύσεων, των σχολίων και άλλων στοιχείων που αφορούν το χρήστη, είτε άμεσα (στοιχεία λογαριασμού, δημοσιεύσεις χρήστη, σχολιασμοί χρήστη κλπ), είτε έμμεσα (σχόλια σε αναρτήσεις του χρήστη, σχόλια σε δημοσιεύσεις άλλων χρηστών κλπ.).

Τέλος, η εφαρμογή αλληλοεπιδρά έμμεσα με το Blockchain για να διασφαλιστεί ότι οι δημοσιεύσεις στο χρονολόγιο του χρήστη είναι έγκυρες. Το Blockchain παρέχει στο σύστημα την ασφάλεια και την αμεταβλητότητα των δεδομένων του χρήστη.



Εικόνα 16. Περιβάλλον συστήματος εφαρμογής

4.2 Απαιτήσεις Συστήματος

Σε αυτή την ενότητα αναφέρονται συνοπτικά οι απαιτήσεις του συστήματος που εξασφαλίζουν την ολοκληρωμένη λειτουργία της εφαρμογής. Πρόκειται για τις λειτουργικότητες της εφαρμογής οι οποίες προσφέρουν την επιθυμητή εμπειρία στο χρήστη. Στο σύνολό τους, οι λειτουργικότητες θα πρέπει να εξυπηρετούν το σκοπό για τον οποίο αναπτύχθηκε η εφαρμογή. Επίσης, προκειμένου να εξασφαλιστεί η ολοκληρωμένη και καλή εμπειρία του χρήστη, είναι απαραίτητο να ληφθούν υπόψη και κάποιες μη λειτουργικές απαιτήσεις. Οι μη λειτουργικές απαιτήσεις δεν έχουν να κάνουν με την υλοποίηση της διεπιφάνειας του χρήστη, αλλά με τη γενικότερη λειτουργικότητα της εφαρμογής.

4.2.1 Λειτουργικές Απαιτήσεις Συστήματος

Η διεπιφάνεια του χρήστη θα πρέπει να φέρει τις παρακάτω λειτουργικότητες:

- **Εγγραφή χρήστη**
Ο χρήστης μπορεί να πραγματοποιήσει εγγραφή στην εφαρμογή, με τη συμπλήρωση κατάλληλης φόρμας.
- **Σύνδεση χρήστη**
Υπάρχοντες χρήστες μπορούν να πραγματοποιήσουν σύνδεση στην εφαρμογή, μέσω της συμπλήρωσης κατάλληλης φόρμας με τα διαπιστευτήριά τους.
- **Ταυτοποίηση χρήστη**
Η ταυτότητα του χρήστη πιστοποιείται μέσω της έκδοσης token.
- **Ανάρτηση Δημοσίευσης**
Ο χρήστης μπορεί να αναρτήσει μια δημοσίευση, σχετικά με την υγεία του. Η ανάρτηση θα εμφανίζεται στο χρονολόγιο της κεντρικής σελίδας. Κατά τη δημιουργία μιας δημοσίευσης είναι εφικτές οι ακόλουθες επιλογές:
(α) προσθήκη του τύπου της δημοσίευσης (Υπάρχουν επτά διαφορετικοί τύποι δημοσίευσης: Text, Article, Doctor Recommendation, Appointment, Prescription, Personal Measurements και Medicine Dosage)

(β) προσθήκη αποστολέα (ώστε η δημοσίευση να σταλθεί σε άλλον χρήστη, προαιρετικά)

(γ) προσθήκη κειμένου

○ **Δημιουργία Σχολίων**

Ο χρήστης μπορεί να σχολιάσει σε όλες τις δημοσιεύσεις (δικές του και άλλων χρηστών).

○ **Προβολή δημοσιεύσεων**

Ο χρήστης μπορεί να δει τη λίστα με όλες τις δημοσιεύσεις. Η λίστα είναι ταξινομημένη με τη πιο πρόσφατη δημοσίευση να φαίνεται πρώτη και την παλαιότερη να φαίνεται τελευταία.

○ **Προβολή σχολίων και «Like» των δημοσιεύσεων**

Ο χρήστης μπορεί να επιβλέπει τα σχόλια όλων των δημοσιεύσεων (δικών του και άλλων χρηστών) καθώς επίσης και τους χρήστες που αντέδρασαν «Like». Επίσης, υπάρχει ένας δείκτης για τα σχόλια που είναι υπεύθυνος για τον αριθμό των σχολίων μιας δημοσίευσης, ενώ ο δείκτης των «Like» δείχνει τον αριθμό των αντιδράσεων «Like».

○ **Επιλογή φίλτρου αναζήτησης**

Ο χρήστης μπορεί να επιλέξει μια κατηγορία δημοσιεύσεων και να πραγματοποιήσει αναζήτηση στις δημοσιεύσεις που ανήκουν σε αυτή.

○ **Αναίρεση φίλτρου αναζήτησης**

Ο χρήστης μπορεί να καταργήσει τα επιλεγμένα φίλτρα για να επιλέξει νέο.

○ **Μπάρα Αναζήτησης**

Ο χρήστης μπορεί να πληκτρολογήσει το όνομα ενός άλλου χρήστη ή μέρος αυτού ώστε να πραγματοποιήσει αναζήτηση ανάμεσα στους χρήστες της εφαρμογής.

○ **Επιλογή στο κεντρικό μενού πλοήγησης**

Ο χρήστης μπορεί να πατήσει στο κεντρικό μενού και να επιλέξει τη σελίδα στην οποία θέλει να μεταβεί. Ο χρήστης μπορεί να πλοηγηθεί στις παρακάτω σελίδες: Profile, Health Score, Calendar και Community.

○ **Προβολή προσωπικών στοιχείων**

Ο χρήστης μπορεί να μεταβεί στο profile του και να δει τα προσωπικά του στοιχεία.

○ **Επεξεργασία προσωπικών στοιχείων**

○ Ο χρήστης μπορεί να επεξεργαστεί τα προσωπικά του στοιχεία και να ανανεώσει το προσωπικό του λογαριασμό μέσω κατάλληλης φόρμας.

○ **Αποστολή αιτήματος σχέσης «Circle of Trust» ή «Care Team» ή «Friends»**

○ Για κάθε χρήστη υπάρχουν τρία διαφορετικά είδη σχέσεων με άλλα άτομα, «Circle of Trust» (που είναι για τους πολύ στενούς συγγενείς ή φίλους), «Care Team» (που είναι οι επιβλέποντες ιατροί του χρήστη) και «Friends» (που είναι οι απλοί φίλοι). Ο χρήστης μπορεί να στείλει τα παραπάνω τρία διαφορετικά είδη αιτημάτων φιλίας σε άλλους χρήστες.

○ **Ακύρωση αιτήματος**

○ Ο χρήστης μπορεί να ακυρώσει το αίτημα που έχει στείλει, όσο ο αιτούμενος δεν έχει απαντήσει.

○ **Προβολή των υπαρχόντων «Circle of Trust», «Protected Members» «Care Team» και «Friends»**

- Ο χρήστης μπορεί να εμφανίσει την λίστα των ατόμων που ανήκουν στο «Circle of Trust» του ή στο «Care Team» του ή στους «Protected Members» ή στους «Friends» του.
- **Διαγραφή ατόμων από το «Circle of Trust», «Care Team» ή τους «Friends» του**
Ο χρήστης μπορεί να αφαιρέσει από τις λίστες των σχέσεων του, όποιον χρήστη επιθυμεί.
 - **Σύνδεση στον λογαριασμό του προστατευόμενου μέλους**
Ο χρήστης μπορεί να συνδεθεί και να περιηγηθεί από τον λογαριασμό του προστατευόμενου μέλους του.
 - **Αποσύνδεση από τον λογαριασμό του προστατευόμενου μέλους**
Ο χρήστης μπορεί να αποσυνδεθεί από τον λογαριασμό του προστατευόμενου μέλους του που νωρίτερα είχε εισέλθει.
 - **Δημιουργία σκορ για την κατάσταση υγείας**
Ο χρήστης μπορεί να δημιουργήσει ένα νέο σκορ για την κατάσταση της υγείας του, με την συμπλήρωση κατάλληλης φόρμας ερωτήσεων.
 - **Προβολή στατιστικών μετρήσεων της κατάστασης υγείας**
Ο χρήστης μπορεί να δει τα στατιστικά δεδομένα σχετικά με όλα τα σκορ για την κατάσταση της υγείας του στην εφαρμογή, όπως ένα διάγραμμα που απεικονίζει το ιστορικό των καταχωρήσεων κλπ.
 - **Προβολή ειδοποιήσεων αιτημάτων φιλίας**
Ο χρήστης μπορεί να δει όλα τα αιτήματα φιλίας που έχει λάβει και έχει στείλει αντίστοιχα.
 - **Προβολή υπενθύμισης για την καταχώρηση της κατάστασης υγείας**
Ο χρήστης ενημερώνεται με ειδοποίηση κάθε μέρα, ώστε να συμπληρώσει το ερωτηματολόγιο σχετικά με την κατάσταση της υγείας του.
 - **Προβολή του ιστορικού των συναλλαγών στο Blockchain**
Ο χρήστης μπορεί να δει όλα τα «transactions» των «posts» που έχουν πραγματοποιηθεί.
 - **Αποσύνδεση**
Ο χρήστης μπορεί να αποσυνδεθεί από το λογαριασμό του κάνοντας logout.

4.2.2 Μη Λειτουργικές Απαιτήσεις Συστήματος

Η εφαρμογή θα πληροί και ορισμένες μη λειτουργικές απαιτήσεις για μια ολοκληρωμένη εμπειρία χρήστη. Οι απαιτήσεις αυτές απαριθμούνται στη συνέχεια.

- **Απόδοση – Performance**
Ο χρόνος απόκρισης του συστήματος στην εξυπηρέτηση αιτημάτων από το server θα πρέπει να είναι μικρός. Το σύστημα θα πρέπει να μπορεί να ανταποκριθεί σε μεγάλο αριθμό αιτημάτων.
- **Ταχύτητα ανάκτησης – Recoverability**
Το σύστημα θα πρέπει να μπορεί να ανακτήσει την επιθυμητή κατάσταση λειτουργίας γρήγορα, σε περίπτωση αποτυχίας ή απότομου τερματισμού.
- **Ασφάλεια – Security**
Η εφαρμογή θα πρέπει να εξασφαλίζει την ασφάλεια των προσωπικών δεδομένων των χρηστών της. Ευαίσθητα δεδομένα όπως κωδικοί ασφαλείας

θα πρέπει να κρυπτογραφούνται πριν την αποθήκευσή τους στη βάση, για την αποφυγή κατάχρησης από κακόβουλους χρήστες.

- **Επεκτασιμότητα – Extensibility/Scalability**

Η υλοποίηση της εφαρμογής θα πρέπει να στηρίζεται σε τεχνολογίες που επιτρέπουν την μελλοντική επέκτασή της με νέα χαρακτηριστικά και πρόσθετες εφαρμογές. Επιπλέον, η εφαρμογή θα πρέπει να μπορεί να διαχειριστεί μεγάλο αριθμό χρηστών.

4.3 Αρχιτεκτονική Εφαρμογής

Όπως είδαμε και στην Εικόνα 16, το περιβάλλον της εφαρμογής διαμορφώνεται από τέσσερις βασικούς συντελεστές: τον client, τον server, τη βάση δεδομένων και το Blockchain. Ο client θα πρέπει να είναι συνδεδεμένος σε δίκτυο προκειμένου να εδραιωθεί η επικοινωνία με τον server. Η διεπικοινωνία μεταξύ client-server είναι υπεύθυνη για την σωστή λειτουργία της εφαρμογής και την διασφάλιση μιας ολοκληρωμένης εμπειρίας από την πλευρά του χρήστη. Ο client είναι υπεύθυνος για την διαμόρφωση ενός διαδραστικού περιβάλλοντος για τον χρήστη. Το περιβάλλον αυτό αποτελείται από έναν router ο οποίος είναι υπεύθυνος για την πλοήγηση του χρήστη στις διάφορες σελίδες της εφαρμογής. Σε κάθε σελίδα, υπάρχουν διαδραστικά στοιχεία όπως γραφικές διεπιφάνειες, διεπαφές, πλήκτρα ενεργειών και στοιχεία εισόδου και διάφορα άλλα στοιχεία αλληλεπίδρασης, όλα με σκοπό την παροχή μιας πλήρους εμπειρίας στο χρήστη. Ο server αναλαμβάνει να εξυπηρετήσει τα αιτήματα που δημιουργεί ο χρήστης από την πλευρά του client. Κάθε φορά που ο χρήστης πραγματοποιεί μια ενέργεια που απαιτεί δεδομένα από τον server, τότε δημιουργείται ένα αίτημα το οποίο καλείται να εξυπηρετήσει ο server. Η προβολή κάποιας δημοσίευσης, η δημιουργία ενός σχολίου, η επεξεργασία του λογαριασμού του χρήστη, αποτελούν μερικά παραδείγματα τέτοιων αιτημάτων. Ο server με την σειρά του επικοινωνεί με το Blockchain, έτσι ώστε να ενημερώσει την νέα κατάσταση του συστήματος. Η βάση δεδομένων αποτελεί τον αποθηκευτικό χώρο της εφαρμογής. Όλα τα δεδομένα αποθηκεύονται σε συλλογές εντός της βάσης. Έτσι, κάθε φορά που ζητούνται δεδομένα, ο server πραγματοποιεί αναζήτηση κατάλληλης μορφής στη βάση, ενημερώνει το Blockchain και επιστρέφει τα απαραίτητα δεδομένα στον client. Μετά από την λήψη των δεδομένων, ο client για να επιβεβαιώσει την ορθότητα τους, επικοινωνεί με το Blockchain έτσι ώστε να λάβει το «hash» των δεδομένων αυτών. Ο client αναλαμβάνει να κατακερματίσει τα δεδομένα που έλαβε από την βάση και να συγκρίνει το «hash» που κατασκεύασε με αυτό που έλαβε από το Blockchain.

4.3.1 Υποδομή Client και Σενάρια Χρήσης

Στις επόμενες ενότητες ακολουθεί μια επεξηγηματική ανάλυση της αρχιτεκτονικής του παραπάνω συστήματος. Θα παρουσιαστούν οι οθόνες της εφαρμογής και θα αναλυθούν όλα τα πιθανά σενάρια χρήσης (use cases) της εφαρμογής.

4.3.1.1 Οθόνες Εφαρμογής

Η εφαρμογή δομείται σε έξι κύρια μέρη: ταυτοποίηση, κεντρική σελίδα των δημοσιεύσεων, το «profile», το «community», το «health score» και το «calendar» . Οι αντίστοιχες οθόνες (screens) αυτών είναι:

(α) Οθόνη Σύνδεσης (Login Screen)

Εκεί μεταφέρεται ο χρήστης την πρώτη φορά που ανοίγει την εφαρμογή, ή όταν αποσυνδεθεί από το λογαριασμό του. Η οθόνη σύνδεσης αποτελείται από μια φόρμα (στην οποία ζητείται το όνομα χρήστη και το password).

(β) Οθόνη Εγγραφής (Sign Up Screen)

Η οθόνη εγγραφής περιέχει φόρμα κατάλληλης μορφής την οποία καλείται να συμπληρώσει ο χρήστης.

(γ) Οθόνη χρονολογίου (Home Screen)

Η κεντρική οθόνη αποτελείται από την λίστα όλων των δημοσιεύσεων που έχει το δικαίωμα να δει ο χρήστης.

(δ) Οθόνη προσωπικού profile (Profile Screen)

Σε αυτή την οθόνη, ο χρήστης μπορεί να προβάλλει τα στοιχεία του λογαριασμού του.

(ε) Οθόνη επεξεργασίας profile (Edit Profile Screen)

Εάν ο χρήστης επιθυμεί να αλλάξει τα στοιχεία του προσωπικού του λογαριασμού ή να επεξεργαστεί κάποια από αυτά, μπορεί να το κάνει σε αυτή τη σελίδα. Η οθόνη αποτελείται από μια φόρμα η οποία περιέχει τα αρχικά στοιχεία του χρήστη πριν από την αλλαγή.

(στ) Οθόνη κοινότητας (Community Screen)

Η οθόνη κοινότητας αποτελείται από 5 tabs («Requests», «Friends», «Circle Of Trust», «Protected Members» και «Care Team»). Στο tab των αιτημάτων, ο χρήστης μπορεί να δει τα αιτήματα φιλίας και να τα δεχτεί ή να τα απορρίψει. Κάθε ένα από τα υπόλοιπα περιέχει την λίστα των ατόμων που ανήκουν στην αντίστοιχη σχέση.

(ζ) Οθόνη του σκορ υγείας (Health Score Screen)

Στην οθόνη του σκορ υγείας παρουσιάζονται στατιστικά δεδομένα σχετικά με το ιστορικό των σκορ υγείας που έχει καταχωρίσει ο χρήστης.

(η) Οθόνη Ημερολογίου (Calendar Screen)

Στην οθόνη του ημερολογίου απεικονίζονται όλα τα επερχόμενα ραντεβού, που ο χρήστης έχει σημειώσει.

(θ) Οθόνη Ιστορικού Blockchain (Blockchain History Screen)

Στην οθόνη του ιστορικού Blockchain απεικονίζονται τα hash όλων των συναλλαγών που πραγματοποιήθηκαν για την κατασκευή των «posts».

4.3.1.2 Σενάρια Χρήσης Εφαρμογής

Εγγραφή Χρήστη στην Εφαρμογή (Sign Up)

- Ο χρήστης ανοίγει την εφαρμογή και πατάει είτε το tab «Sign Up» που βρίσκεται πάνω δεξιά της οθόνης είτε το link «Sign Up» που βρίσκεται στο κάτω δεξιό μέρος της οθόνης σύνδεσης (Login Screen).

- Ο χρήστης συμπληρώνει τη φόρμα εγγραφής, έτσι ώστε να εισάγει όλες τις απαιτούμενες πληροφορίες. Τα απαιτούμενα πεδία είναι τα εξής: πλήρες όνομα (fullname), όνομα χρήστη (username), κωδικός (password), επαλήθευση κωδικού (confirm password), ημερομηνία γέννησης (birthdate) και γένος χρήστη (gender). Πατάει το κουμπί «Sign Up» για να ολοκληρωθεί η εγγραφή.
- Στην περίπτωση που δεν υπάρχουν σφάλματα κατά την υποβολή της φόρμας εγγραφής, ο χρήστης εγγράφεται στην εφαρμογή με επιτυχία. Η εφαρμογή τον πηγαίνει αυτόματα στην κεντρική σελίδα του Χρονολόγιου (Home Screen).

Σύνδεση Χρήστη στην Εφαρμογή (Login)

- Ο χρήστης ανοίγει την εφαρμογή και συμπληρώνει τη φόρμα σύνδεσης. Στη φόρμα ο χρήστης πρέπει να εισάγει όλες τις απαιτούμενες πληροφορίες. Τα απαιτούμενα πεδία είναι τα εξής: το όνομα χρήστη (username) και ο κωδικός πρόσβασης (password)
- Στην περίπτωση που δεν υπάρχουν σφάλματα κατά την υποβολή της φόρμας σύνδεσης, ο χρήστης επαναφέρεται αυτόματα στην αρχική σελίδα (Home Screen).

Αποσύνδεση χρήστη από την Εφαρμογή (Logout)

- Ο χρήστης πατάει το δεξί κουμπί «Logout» της πάνω μπάρας.
- Ο χρήστης μεταφέρεται στην Οθόνη Σύνδεσης.

Δημιουργία Δημοσίευσης

- Ο χρήστης πατάει πάνω στο πλήκτρο «Create a Post» στο πάνω δεξιά μέρος της οθόνης στην Home Screen.
- Ο χρήστης συμπληρώνει την φόρμα που εμφανίζεται σε ένα αναδυόμενο παράθυρο (modal) για την δημιουργία της δημοσίευσης.
- Ο χρήστης πατάει το πλήκτρο «Post» για να καταχωρήσει την νέα δημοσίευση.
- Στην περίπτωση που δεν υπάρχουν σφάλματα κατά την δημιουργία της νέας δημοσίευσης, η διαδικασία ολοκληρώνεται επιτυχώς και ο χρήστης μεταφέρεται αυτόματα στην αρχική σελίδα. Το νέο δημοσίευμα θα εμφανίζεται στο Χρονολόγιο (Home Screen).

Επεξεργασία Δημοσίευσης

- Ο χρήστης πατάει το κουμπί που βρίσκεται πάνω δεξιά της δημοσίευσης που θέλει να επεξεργαστεί.
- Επιλέγει από την λίστα που του εμφανίζεται την επιλογή «Edit».
- Ο χρήστης επεξεργάζεται τα επιθυμητά στοιχεία της φόρμας που εμφανίζεται σε ένα αναδυόμενο παράθυρο (modal) και πατάει το «Save», που βρίσκεται στο κάτω μέρος του παραθύρου.

- Σε περίπτωση που δεν υπάρχουν σφάλματα, η διαδικασία ολοκληρώνεται επιτυχώς και οι αλλαγές αποθηκεύονται. Η δημοσίευση εμφανίζεται πλέον με τις νέες αλλαγές στην οθόνη του Χρονολογίου (Home Screen).

Διαγραφή Δημοσίευσης

- Ο χρήστης πατάει το κουμπί που βρίσκεται πάνω δεξιά της δημοσίευσης που θέλει να διαγράψει.
- Ο χρήστης επιλέγει από την λίστα που του εμφανίζεται την επιλογή «Delete».
- Το δημοσίευμα διαγράφεται από την λίστα των δημοσιευμάτων

Προβολή Σχολίων

- Ο χρήστης πατάει το κουμπί «Comments» που βρίσκεται κάτω δεξιά της δημοσίευσης.
- Τα σχόλια εμφανίζονται κάτω από την δημοσίευση.

Δημιουργία Σχολίου

- Ο χρήστης ανοίγει την λίστα των σχολίων ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο Προβολή Σχολίων.
- Ο χρήστης πλοηγείται στο κάτω μέρος της λίστας των σχολίων που εμφανίζονται.
- Ο χρήστης εισάγει το σχόλιό του στο «comment box» με επιγραφή «Add a comment...».
- Μόλις ο χρήστης ολοκληρώσει το σχόλιό του, πατάει στο πλήκτρο «Post».
- Σε περίπτωση που το σχόλιο δεν έχει σφάλματα, η διαδικασία ολοκληρώνεται επιτυχώς και το νέο σχόλιο εμφανίζεται τελευταίο στο τμήμα σχολίων της οθόνης.

Επεξεργασία Σχολίου

- Ο χρήστης ανοίγει την λίστα των σχολίων ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο Προβολή Σχολίων.
- Ο χρήστης πατάει το κουμπί επιπλέον επιλογών που βρίσκεται στα δεξιά του σχολίου.
- Ο χρήστης πατάει την επιλογή «Edit» από την λίστα που εμφανίζεται.
- Ο χρήστης αλλάζει το σχόλιο του στο comment box και πατάει το κουμπί «Update».
- Σε περίπτωση που το σχόλιο δεν έχει σφάλματα, το σχόλιο εμφανίζεται πλέον με τις νέες αλλαγές στην λίστα των σχολίων.

Διαγραφή Σχολίου

- Ο χρήστης ανοίγει την λίστα των σχολίων ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο Προβολή Σχολίων.

- Ο χρήστης πατάει το κουμπί επιπλέον επιλογών που βρίσκεται στα δεξιά του σχολίου .
- Ο χρήστης πατάει την επιλογή «Delete» από την λίστα που εμφανίζεται.
- Το σχόλιο διαγράφεται από την λίστα.

Επιλογή «Like»

- Ο χρήστης πατάει το κουμπί «Like» που βρίσκεται στο κάτω αριστερά μέρος της δημοσίευσης.
- Το «Like» ενεργοποιείται.

Αναίρεση της επιλογής «Like»

- Ο χρήστης ξανά πατάει το κουμπί «Like» που βρίσκεται στο κάτω αριστερά μέρος της δημοσίευσης.
- Το «Like» απενεργοποιείται.

Προβολή των ατόμων που έχουν κάνει «Like» σε μια δημοσίευση

- Ο χρήστης πατάει στον δείκτη των «Likes» που βρίσκεται κάτω αριστερά της δημοσίευσης (ακριβώς πάνω από το κουμπί του «Like»).
- Εμφανίζεται στον χρήστη ένα αναδυόμενο παράθυρο, το οποίο έχει τα ονόματα όλων των χρηστών που έχουν κάνει «Like».

Αναζήτηση Δημοσιεύσεων με Φίλτρο

- Στην κεντρική σελίδα, ο χρήστης μπορεί να επιλέξει στην φόρμα, που βρίσκεται πάνω δεξιά της οθόνης, ποιόν τύπο δημοσιεύσεων επιθυμεί να δει. Οι διαθέσιμες κατηγορίες είναι «Text», «Article», «Doctor Recommendation», «Appointment», «Prescription», «Personal Measurements» και «Medicine Dosage».
- Τα αποτελέσματα εμφανίζονται στην κεντρική οθόνη αυτόματα. Τα νέα αποτελέσματα φορτώνονται κάθε φορά που αλλάζει ο τύπος της δημοσίευσης.

Προβολή Προσωπικού Λογαριασμού (Profile)

- Ο χρήστης πατάει πάνω στην επιλογή «Profile» του «menu» που βρίσκεται στο δεξιά μέρος της οθόνης.
- Ο χρήστης μεταφέρεται στη σελίδα προσωπικού λογαριασμού. Σε αυτήν την οθόνη ο χρήστης μπορεί να προβάλει τα προσωπικά του στοιχεία.

Επεξεργασία Προσωπικού Λογαριασμού

- Ο χρήστης μεταφέρεται στη σελίδα προσωπικού λογαριασμού, ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή Προσωπικού Λογαριασμού»

- Ο χρήστης πατάει το πλήκτρο «Edit Profile» που βρίσκεται στο επάνω αριστερά μέρος της οθόνης.
- Ο χρήστης μεταφέρεται στη σελίδα επεξεργασίας του προσωπικού λογαριασμού. Η οθόνη αυτή αποτελείται από μια φόρμα με τα αρχικά στοιχεία του χρήστη. Ο χρήστης μπορεί να επεξεργαστεί τα προσωπικά του στοιχεία και στην συνέχεια να πατήσει το πλήκτρο «Save» που βρίσκεται πάνω αριστερά της φόρμας.
- Σε περίπτωση επιτυχούς επεξεργασίας των στοιχείων του προσωπικού λογαριασμού, τα νέα στοιχεία αποθηκεύονται στο σύστημα και ο χρήστης επαναφέρεται στο «profile» του.

Προβολή Αιτημάτων

- Ο χρήστης πατάει επάνω στην επιλογή «Community», στο δεξιά μέρος της οθόνης.
- Ο χρήστης μεταφέρεται στη σελίδα των αιτημάτων που έχει λάβει. Σε αυτήν την οθόνη ο χρήστης μπορεί να δει όλα τα αιτήματα ανάλογα με το είδος της σχέσης («Care Team», «Circle Of Trust», «Friend»).

Αποδοχή Αιτήματος

- Ο χρήστης μεταφέρεται στην σελίδα των αιτημάτων ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή Αιτημάτων».
- Ο χρήστης πατάει το κουμπί «Accept» που βρίσκεται στα δεξιά του αιτήματος.
- Η σχέση αυτή αποθηκεύεται στο σύστημα και το αίτημα διαγράφεται αυτόματα από την λίστα των αιτημάτων.

Απόρριψη Αιτήματος

- Ο χρήστης μεταφέρεται στην σελίδα των αιτημάτων ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή Αιτημάτων».
- Ο χρήστης πατάει το κουμπί «Decline» που βρίσκεται στα δεξιά του αιτήματος.
- Το αίτημα διαγράφεται αυτόματα.

Προβολή Φίλων

- Ο χρήστης πατάει επάνω στην επιλογή «Community», στο δεξιά μέρος της οθόνης.
- Ο χρήστης μεταβαίνει στην σελίδα των αιτημάτων και επιλέγει το tab «Friends».
- Η λίστα των φίλων εμφανίζεται στη οθόνη.

Προσθήκη Φίλου

- Ο χρήστης μεταφέρεται στην σελίδα των φίλων ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή Φίλων».
- Ο χρήστης γράφει το όνομα ή το username του χρήστη που θέλει να προσθέσει, στην μπάρα αναζήτησης που βρίσκεται πάνω από την λίστα των υπαρχόντων φίλων.
- Επιλέγει τον χρήστη και στέλνει αυτόματα αίτημα φιλίας σε αυτόν.

Διαγραφή Φίλου

- Ο χρήστης μεταφέρεται στην σελίδα των φίλων ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή Φίλων».
- Ο χρήστης πατάει το κουμπί «Remove» που βρίσκεται στα δεξιά του κάθε χρήστη.
- Ο φίλος διαγράφεται αυτόματα από την λίστα.

Προβολή του Care Team

- Ο χρήστης πατάει επάνω στην επιλογή «Community», στο δεξιά μέρος της οθόνης.
- Ο χρήστης μεταβαίνει στην σελίδα των αιτημάτων και επιλέγει το tab «Care Team».
- Η λίστα των χρηστών που ανήκουν στο Care Team του χρήστη εμφανίζεται στη οθόνη.

Προσθήκη Χρήστη στο Care Team

- Ο χρήστης μεταφέρεται στην σελίδα των «Care Team» ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή του Care Team».
- Ο χρήστης γράφει το όνομα ή το username του χρήστη που θέλει να προσθέσει, στην μπάρα αναζήτησης που βρίσκεται πάνω από την λίστα των ατόμων που ανήκουν στο «Care Team».
- Επιλέγει τον χρήστη και στέλνει αυτόματα αίτημα σε αυτόν.

Διαγραφή χρήστη από το Care Team

- Ο χρήστης μεταφέρεται στην σελίδα των «Care Team» ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή του Care Team».
- Ο χρήστης πατάει το κουμπί «Remove» που βρίσκεται στα δεξιά του χρήστη.
- Ο χρήστης διαγράφεται αυτόματα από την λίστα.

Προβολή του Circle Of Trust

- Ο χρήστης πατάει επάνω στην επιλογή «Community», στο δεξιά μέρος της οθόνης.
- Ο χρήστης μεταβαίνει στην σελίδα των αιτημάτων και επιλέγει το tab «Circle Of Trust».

- Η λίστα των χρηστών που ανήκουν στο Circle Of Trust του χρήστη εμφανίζεται στη οθόνη.

Προσθήκη Χρήστη στο Circle Of Trust

- Ο χρήστης μεταφέρεται στην σελίδα των «Circle Of Trust» ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή του Circle Of Trust».
- Ο χρήστης γράφει το όνομα ή το username του χρήστη που θέλει να προσθέσει, στην μπάρα αναζήτησης που βρίσκεται πάνω από την λίστα των ατόμων που ανήκουν στο «Circle Of Trust».
- Επιλέγει τον χρήστη και στέλνει αυτόματα αίτημα σε αυτόν.

Διαγραφή χρήστη από το Circle Of Trust

- Ο χρήστης μεταφέρεται στην σελίδα των «Circle Of Trust» ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή του Circle of Trust».
- Ο χρήστης πατάει το κουμπί «Remove» που βρίσκεται στα δεξιά του χρήστη.
- Ο χρήστης διαγράφεται αυτόματα από την λίστα.

Προβολή των προστατευόμενων μελών (Protected Members)

- Ο χρήστης πατάει επάνω στην επιλογή «Community», στο δεξιά μέρος της οθόνης.
- Ο χρήστης μεταβαίνει στην σελίδα των αιτημάτων και επιλέγει το tab «Protected Members».
- Η λίστα των χρηστών που ανήκουν στο «Protected Members» του χρήστη εμφανίζεται στη οθόνη.

Σύνδεση στον λογαριασμό του προστατευόμενου μέλους

- Ο χρήστης μεταφέρεται στην σελίδα των «Protected Members» ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή των προστατευόμενων μελών».
- Ο χρήστης πατάει το κουμπί «Enter» που βρίσκεται πάνω δεξιά του προστατευόμενου μέλους.
- Ο χρήστης μεταβαίνει στην αρχική σελίδα του προστατευόμενου μέλους και πλέον περιηγείται σε όλες τις πληροφορίες του λογαριασμού αυτού.

Αποσύνδεση από τον λογαριασμό του προστατευόμενου μέλους

- Ο χρήστης όντας συνδεδεμένος στον λογαριασμό του προστατευόμενου μέλους, πατάει το κουμπί «Exit» που βρίσκεται στο δεξί μέρος της επάνω μπάρας.
- Ο χρήστης μεταβαίνει στον δικό του λογαριασμό.

Διαγραφή χρήστη από τα προστατευόμενα μέλη (Protected Members)

- Ο χρήστης μεταφέρεται στην σελίδα των «Protected Members» ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή των προστατευόμενων μελών».
- Ο χρήστης πατάει το κουμπί «Remove» που βρίσκεται στα δεξιά του χρήστη.
- Ο χρήστης διαγράφεται αυτόματα από την λίστα.

Αναζήτηση Χρηστών

- Στη κεντρική σελίδα, ο χρήστης πατάει εντός της μπάρας αναζήτησης με επιγραφή «Search» που βρίσκεται στο πάνω μέρος της οθόνης.
- Ο χρήστης πληκτρολογεί το όνομα ή κάποια άλλη λέξι-κλειδί που σχετίζεται με το όνομα του χρήστη.
- Τα αποτελέσματα εμφανίζονται κάτω από την μπάρα αναζήτησης.

Προβολή Προσωπικού Λογαριασμού άλλων χρηστών

- Ο χρήστης αναζητεί άλλους χρήστες, ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Αναζήτηση Χρηστών».
- Ο χρήστης επιλέγει το χρήστη από την λίστα που εμφανίζεται.
- Ο χρήστης μεταφέρεται στον προσωπικό λογαριασμό του χρήστη.

Προβολή στατιστικών δεδομένων σχετικά με το σκορ υγείας του χρήστη

- Ο χρήστης πατάει επάνω στην επιλογή «Health Score», στο δεξιά μέρος της οθόνης.
- Ο χρήστης μεταβαίνει στην σελίδα στατιστικών δεδομένων του σκορ υγείας. Η οθόνη διαιρείται σε τρία τμήματα:
 - (α) το πιο πρόσφατο σκορ υγείας
Εμφανίζεται το τελευταίο σκορ υγείας, που ο χρήστης έχει καταχωρίσει.
 - (β) ιστορικό των σκορ υγείας
Εμφανίζεται σε μορφή διαγράμματος όλα τα σκορ υγείας που έχουν καταχωρηθεί.
 - (γ) οι πιο συχνές απαντήσεις
Εμφανίζεται σε μορφή στατιστικών πιτών, για κάθε ερώτηση, το ποσοστό που απαντήθηκε η κάθε επιλογή.

Υποβολή Σκορ Υγείας

- Ο χρήστης μεταφέρεται στην σελίδα των στατιστικών δεδομένων για το σκορ υγείας ακολουθώντας τη διαδικασία που περιγράφεται στην παράγραφο «Προβολή στατιστικών δεδομένων σχετικά με το σκορ υγείας του χρήστη».
- Ο χρήστης πατάει το κουμπί «New Health Score», το οποίο βρίσκεται στο πάνω αριστερό μέρος της οθόνης.
- Ο χρήστης συμπληρώνει μια φόρμα πολλαπλών ερωτήσεων, που εμφανίζεται σε ένα αναδυόμενο παράθυρο και πατάει το κουμπί «Submit».

- Στην περίπτωση που δεν υπάρχουν σφάλματα κατά τη δημιουργία νέου σκορ υγείας, η διαδικασία ολοκληρώνεται επιτυχώς. Το νέο σκορ υγείας θα εμφανίζεται σαν το πιο πρόσφατο σκορ υγείας.

Προβολή Ιστορικού Blockchain

- Ο χρήστης πατάει επάνω στην επιλογή «Blockchain History», στο δεξιά μέρος της οθόνης.
- Ο χρήστης μεταβαίνει στην σελίδα ιστορικού Blockchain. Τα hash όλων των συναλλαγών που πραγματοποιήθηκαν για την δημιουργία των posts εμφανίζονται σε ένα πίνακα, παρουσιάζοντας την εγκυρότητα ή μη αυτών των συναλλαγών.

4.3.2 Διεπικοινωνία Μεταξύ Υποδομών

Στην προηγούμενη ενότητα, παρουσιάστηκαν οι βασικές λειτουργικότητες που προσφέρει η εφαρμογή στους χρήστες της. Αναλύθηκαν οι υπηρεσίες της εφαρμογής καθώς επίσης και τα σενάρια με τα οποία ο χρήστης μπορεί να αποκτήσει πρόσβαση σε αυτές. Έτσι, όταν ο χρήστης επιθυμεί για παράδειγμα να δημιουργήσει μια νέα δημοσίευση, αρκεί να πλοηγηθεί στην οθόνη δημιουργίας νέας δημοσίευσης, να συμπληρώσει την αντίστοιχη φόρμα και να την υποβάλλει. Αυτές είναι οι ενέργειες που πρέπει να γίνουν από την πλευρά του client.

Το σύστημα client-server επικοινωνεί μέσω αιτημάτων συγκεκριμένου σκοπού. Κάθε φορά που ο χρήστης πραγματοποιεί μια ενέργεια εντός της εφαρμογής, ο client δημιουργεί ένα αίτημα προς στον server. Το αίτημα αυτό δρομολογείται στον κατάλληλο εξυπηρετητή από την πλευρά του server. Έπειτα, ακολουθεί η ανάθεση του αιτήματος στον αντίστοιχο πόρο και η εκτέλεση των απαραίτητων λειτουργιών. Ο server δημιουργεί μια εντολή προς το Blockchain. Η εντολή αυτή εκτελείται από το Blockchain και ενημερώνει την κατάσταση του, επιστρέφοντας στον server μήνυμα επιτυχίας της εντολής. Το αποτέλεσμα επηρεάζει τα δεδομένα στη βάση και επιστρέφει με κατάλληλη μορφή πίσω στον client. Η παραπάνω διαδικασία γίνεται βέλτιστα, μέσω ασύγχρονων μεθόδων διαχείρισης δεδομένων από και προς τη βάση δεδομένων. Η διαδικασία εξυπηρέτησης αιτημάτων είναι παραλληλοποιημένη ώστε πολλά αιτήματα να εξυπηρετούνται από διαφορετικούς πόρους του συστήματος ταυτόχρονα.

Κεφάλαιο 5. Λειτουργικότητα και Υλοποίηση του Συστήματος

5.1 Λειτουργικότητα του συστήματος

Σκοπός της παρούσας διπλωματικής αποτέλεσε η υλοποίηση της web app εκδοχής του Impilo, μιας καινοτόμας εφαρμογής κοινωνικής δικτύωσης για την υγεία, η οποία διαθέτει και αντίστοιχη εφαρμογή native (που τρέχει σε κινητές συσκευές). Η εφαρμογή απευθύνεται τόσο σε ασθενείς, όσο και σε γιατρούς καθώς και σε συγγενείς των ασθενών, οι οποίοι έχουν τη δυνατότητα φτιάχνοντας λογαριασμό να αξιοποιούν με διαφορετικό τρόπο τη δικτύωση που τους παρέχεται. Οι μεν ασθενείς, μπορούν να μοιράζονται με τους γιατρούς το ιστορικό της υγείας τους, καθώς και να ενημερώνονται από τους γιατρούς τους με ό,τι τους αφορά, όπως συνταγογραφήσεις, παροτρύνσεις, επικείμενα ραντεβού κ.ά. (σχέση Care Team). Οι δε γιατροί μπορούν να μένουν πάντα ενήμεροι σχετικά με την κατάσταση της υγείας των ασθενών τους και να αλληλοεπιδρούν με αυτούς, με το Impilo να διευκολύνει τόσο του μεν όσο και τους δε. Επιπρόσθετα, οι συγγενείς των ασθενών μπορούν να ενημερώνουν το ιστορικό υγείας του ασθενή οι ίδιοι, σε περίπτωση που αυτός δεν είναι σε θέση να χρησιμοποιήσει την εφαρμογή. Αυτό, μάλιστα, το Impilo το ονομάζει σχέση «Circle of Trust». Τέλος, οι ασθενείς μπορούν να «γίνονται φίλοι» με άλλους ασθενείς και να γνωστοποιούν και σε αυτούς το ιστορικό τους στο χρονολόγιό τους (σχέση Friends), στο οποίο μάλιστα μπορούν να φιλτράρουν το είδος των δημοσιεύσεων που φαίνονται.

Αξίζει μάλιστα να σημειωθεί, ότι η πλατφόρμα ομοιάζει σε μεγάλο βαθμό διαδεδομένα κοινωνικά δίκτυα που χρησιμοποιούνται από την πλειοψηφία του πληθυσμού στη σύγχρονη κοινωνία (πχ Facebook), διαθέτοντας χρονολόγιο, δημοσιεύσεις, «like» και σχόλια. Έτσι, καθιστά μηδαμινό το χρόνο εξοικείωσης του χρήστη με την πλατφόρμα και ενώ του δίνει παράλληλα ένα φιλικό και γνώριμο περιβάλλον χρήσης. Σε αυτό το μήκος κύματος, η εφαρμογή διαθέτει κουμπί και πλαίσιο ειδοποιήσεων στην πάνω μπάρα, ενημερώνοντάς τους σχετικά με υποχρεώσεις τους π.χ. συμπλήρωση ημερήσιου σκορ υγείας ή με αιτήματα σύνδεσης με άλλους χρήστες, μέσω των προαναφερόμενων σχέσεων, μία ακόμα γνώριμη λειτουργία των τυπικών κοινωνικών δικτύων. Παράλληλα, κάθε χρήστης διαθέτει το προσωπικό του «προφίλ», το οποίο αποτελεί τη βιτρίνα του σε σχέση με τους άλλους χρήστες, κοινοποιώντας σε αυτό μερικές σημαντικές πληροφορίες για αυτόν.

Ταυτόχρονα, υπάρχουν και πρωτοπόρες δυνατότητες που δε διαθέτουν τα «κανονικά» κοινωνικά δίκτυα. Καταρχάς, υπάρχει η σελίδα «Health Score», το οποίο αποτελεί τη σελίδα στατιστικών δεδομένων του σκορ υγείας με τρόπο αναλυτικό και την ίδια ώρα ξεκάθαρο (βλ. επόμενη ενότητα). Έπειτα, υπάρχει η σελίδα με το ημερολόγιο, στο οποίο τόσο ο ασθενής όσο και ο γιατρός μπορεί να δει επερχόμενα ραντεβού με γιατρό ή με ασθενείς, αντίστοιχα, με τρόπο παραστατικό και διαδραστικό, υπό την έννοια ότι καθίσταται δυνατή και η τροποποίηση αυτών. Το Impilo θα είναι αυτό που θα αναλαμβάνει να ενημερώσει τον εμπλεκόμενο στο ραντεβού στην περίπτωση αυτή.

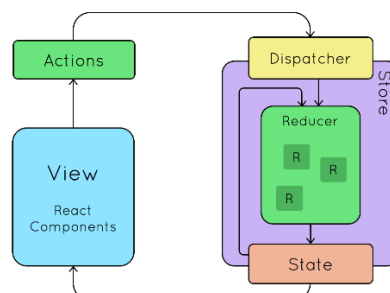
Τέλος, είναι σημαντικό να αναφερθεί ότι την ακεραιότητα των διάφορων ειδών αλληλεπιδράσεων μεταξύ των χρηστών «προστατεύει» ένα δίκτυο blockchain. Συγκεκριμένα, κάθε «συναλλαγή» που γίνεται στο Imriilo μεταβιβάζεται στην εφαρμογή μόνο σε περίπτωση ασφαλούς διαπίστευσης του χρήστη, καθώς και η συνεχής ενημέρωση των δεδομένων της εφαρμογής ελέγχει την εγκυρότητα της διαπίστευσης του χρήστη. Έτσι, εξασφαλίζεται η αξιοπιστία της εφαρμογής, δεδομένου ότι αποτελεί ένα εργαλείο στο οποίο ανταλλάσσονται εμπιστευτικά αρχεία και κρίσιμες, προσωπικές πληροφορίες. Αυτή, μάλιστα, θα μπορούσε να θεωρηθεί η μεγαλύτερη καινοτομία του Imriilo σε σχέση με άλλα ιατρικά κοινωνικά δίκτυα, στην προσπάθεια της εφαρμογής να πείσει τον χρήστη να μοιραστεί τις προσωπικές του πληροφορίες σε αυτήν.

5.2 Υλοποίηση του Συστήματος

5.2.1 Γενική Περιγραφή

Η εφαρμογή αναπτύχθηκε σε περιβάλλον Node.js χρησιμοποιώντας τον Webpack development server για developing και testing, και αργότερα εξήχθη σε production με χρήστη του Webpack (webpack, n.d.) ως bundler και του Babel ως transpiler ώστε να εξαχθεί σε κώδικα JavaScript ES5 και να υποστηρίζεται και σε παλιότερες εκδόσεις browser.

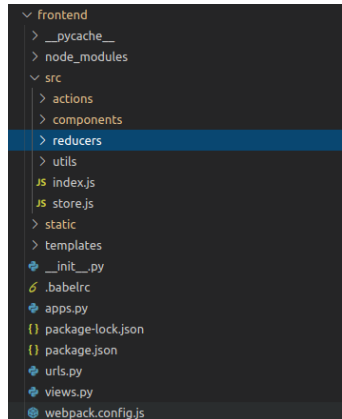
Όπως έχει ήδη αναφερθεί, η εφαρμογή είναι γραμμένη σε React μαζί με Redux. Σε ό,τι αφορά το πρώτο, αυτό σημαίνει ότι το πρόγραμμα είναι χωρισμένο σε διακριτά Component τα οποία αναλαμβάνουν το χτίσιμο του UI, καλώντας για το σκοπό αυτό και άλλα Component. Σε ό,τι αφορά το δεύτερο, η χρήση του Redux αυτόματα σημαίνει ότι η κατάσταση της εφαρμογής “ζει” στο Store. Δηλαδή, οποιαδήποτε αλληλεπίδραση του χρήστη με την εφαρμογή σημάνει κάποιο Action (δράση), το οποίο στη συνέχεια φτάνει στους Reducers που αναλαμβάνουν να τροποποιήσουν το Store, άρα και το State. Προφανώς και η δομή αυτή είναι που ευνοεί την πτυχή της συνεργατικότητας της εφαρμογής, το να μπορούν δηλαδή τα διάφορα Components να κάνουν dispatch τα Actions. Επομένως, απαραίτητη ήταν επίσης η δημιουργία νέων Actions καθώς και Reducers που θα αναλαμβάνουν το χειρισμό αυτών. Στην παρακάτω εικόνα παρουσιάζεται η σύνδεση της React με την Redux και πως αυτά αλληλοεπιδρούν.



Εικόνα 17. Η αλληλεπίδραση της React με την Redux

5.2.2 Λεπτομέρειες Υλοποίησης Διεπαφών Χρήστη

Η κύρια δομή του καταλόγου φακέλων του προγράμματος είναι η παρακάτω:



Εικόνα 18. Δομή του καταλόγου φακέλου του προγράμματος

Μια από τις πρώτες προγραμματιστικές αποφάσεις που πάρθηκαν ήταν ο τρόπος σερβιρίσματος του front-end. Για την αποφυγή της χρήσης CORS στον server, επιλέχθηκε το front-end να σερβίρεται από το Django. Το front-end μέρος της εφαρμογής αποτελεί ένα από τα «service» του Django, για αυτό τον λόγο βρίσκεται μέσα στο «repository» του Impilo Backend. Στον φάκελο src εμπεριέχονται όλοι οι κώδικες του front-end, οι οποίοι είναι ταξινομημένοι στα components, στα utils, στα actions και στους reducers. Επομένως, η πλειοψηφία του κώδικα που γράφτηκε αποτελείται από components, actions και reducers. Επειδή το πλήθος και ο όγκος τους είναι πολύ μεγάλα (), παρατίθενται ενδεικτικά ένα component, τα αντίστοιχα actions και οι αντίστοιχοι reducers.

Παρακάτω παρατίθεται ο κώδικας του component του health score:

```
import React, { Component } from "react";
import { Row, Col, Button } from "reactstrap";
import Box from "../cards/Box";
import BoxHeader from "../cards/BoxHeader";
import HealthScoreModal from "../helppages/HealthScoreModal";

import { connect } from "react-redux";
import PropTypes from "prop-types";
import ReactEcharts from "echarts-for-react";

import {
  getHealthScores,
  getLatestHealthScore
} from "../../actions/healthscore";

const timeConverter = UNIX_timestamp => {
  const a = new Date(UNIX_timestamp);
  const months = [
    "Jan",
    "Feb",
    "Mar",
    "Apr",
    "May",
    "Jun",
    "Jul",
    "Aug",
    "Sep",
    "Oct",
    "Nov",
```

```

    "Dec"
  ];
  const year = a.getFullYear();
  const month = months[a.getMonth()];
  const date = a.getDate();
  const hour = a.getHours();
  const min = a.getMinutes();
  const sec = a.getSeconds();
  const time =
    date + " " + month + " " + year + "\n" + hour + ":" + min + ":" + sec;
  return time;
};

class HealthScore extends Component {
  constructor(props) {
    super(props);
    this.state = {
      isModalOpen: false
    };
  }
  componentDidMount() {
    this.props.getHealthScores();
    this.props.getLatestHealthScore();
  }

  toggleModal = () => {
    this.setState(prevState => ({
      isModalOpen: !prevState.isModalOpen
    }));
  };

  render() {
    const { healthScores } = this.props.healthscore;
    const { score } = this.props.healthscore.latestHealthScore;
    const data = healthScores.map(hscore => {
      const datetime = hscore.dateCreated.concat(" UTC");
      return [Date.parse(datetime), hscore.score];
    });
    const res = healthScores.reduce(
      (acc, currentValue) => {
        Object.keys(acc).forEach(key => {
          acc[key][currentValue[key]] = acc[key][currentValue[key]] + 1;
        });
        return acc;
      },
      {
        anxietyDepression: { "1": 0, "2": 0, "3": 0 },
        painDiscomfort: { "1": 0, "2": 0, "3": 0 },
        mobility: { "1": 0, "2": 0, "3": 0 },
        usualActivities: { "1": 0, "2": 0, "3": 0 },
        selfCare: { "1": 0, "2": 0, "3": 0 }
      }
    );

    const optionDiagram = {
      title: {
        text: "Health Score",
        subtext: "EQ-5D 3L",
        left: "center"
      },
      xAxis: {
        type: "category",
        boundaryGap: false,
        // axisLine: { onZero: false },
        axisLabel: {
          formatter: value => timeConverter(value)
        },
        splitLine: {
          show: false
        }
      },
      yAxis: {
        type: "value",
        splitLine: {
          show: false
        }
      }
    },
  };

```

```

series: [
  {
    data: data.reverse(),
    type: "line",
    animationDelay: idx => idx * 250
  }
],
tooltip: {
  trigger: "axis",
  formatter: value => {
    return `${timeConverter(value[0].data[0])}<br/>${value[0].marker}${
      value[0].data[1]
    }`;
  }
}
};

const optionPie = {
  legend: {},
  title: [
    {
      text: "Mobility",
      left: "13%",
      top: "6%"
    },
    {
      text: "Self Care",
      left: "45%",
      top: "30%"
    },
    {
      text: "Usual Activities",
      right: "13%",
      top: "6%"
    },
    {
      text: "Pain/Discomfort",
      left: "13%",
      top: "63%"
    },
    {
      text: "Anxiety/Depression",
      right: "11%",
      top: "63%"
    }
  ],
  tooltip: {},
  dataset: {
    source: [
      [
        "Category",
        "Mobility",
        "Self Care",
        "Usual Activities",
        "Pain/Discomfort",
        "Anxiety/Depression"
      ],
      [
        "Extreme problems",
        res.mobility[3],
        res.selfCare[3],
        res.usualActivities[3],
        res.painDiscomfort[3],
        res.anxietyDepression[3]
      ],
      [
        "Some problems",
        res.mobility[2],
        res.selfCare[2],
        res.usualActivities[2],
        res.painDiscomfort[2],
        res.anxietyDepression[2]
      ],
      [
        "No problems",
        res.mobility[1],
        res.selfCare[1],

```



```

        res.usualActivities[1],
        res.painDiscomfort[1],
        res.anxietyDepression[1]
    ]
  ],
  series: [
    {
      type: "pie",
      radius: 60,
      center: ["17%", "25%"],
      encode: {
        itemName: "Category",
        value: "Mobility"
      }
    },
    {
      type: "pie",
      radius: 60,
      center: ["50%", "50%"],
      encode: {
        itemName: "Category",
        value: "Self Care"
      }
    },
    {
      type: "pie",
      radius: 60,
      center: ["80%", "25%"],
      encode: {
        itemName: "Category",
        value: "Usual Activities"
      }
    },
    {
      type: "pie",
      radius: 60,
      center: ["20%", "85%"],
      encode: {
        itemName: "Category",
        value: "Anxiety/Depression"
      }
    },
    {
      type: "pie",
      radius: 60,
      center: ["80%", "85%"],
      encode: {
        itemName: "Category",
        value: "Pain/Discomfort"
      }
    }
  ]
};

return (
  <>
  <HealthScoreModal
    isOpen={this.state.isModalOpen}
    toggle={this.toggleModal}
  />
  <Row>
    <Box>
      <Row className="mb-3">
        <Col md={4}>
          <Button color="info" block onClick={this.toggleModal}>
            New Health Score
          </Button>
        </Col>
        <Col md={4}></Col>
        <BoxHeader size={4} header="Latest Health Score">
          <Row>
            <Col>
              {score ? (
                <h5 style={{ textAlign: "center" }}>{score}</h5>
              ) : (
                <p style={{ textAlign: "center" }}>

```

```

        <strong>
          You have not submitted any health score yet
        </strong>
      </p>
    )}
  </Col>
</Row>
</BoxHeader>
</Row>
<Row>
  <Box>
    <ReactEcharts
      option={optionDiagram}
      style={{ height: "300px" }}
    />
  </Box>
</Row>
{healthScores.length ? (
  <Row>
    <Box>
      <ReactEcharts
        option={optionPie}
        style={{ height: "600px" }}
      />
    </Box>
  </Row>
) : null}
</Box>
</Row>
</>
);
}
}

HealthScore.propTypes = {
  getHealthScores: PropTypes.func.isRequired
};

const mapStateToProps = state => ({
  healthscore: state.healthscore
});

const mapDispatchToProps = { getHealthScores, getLatestHealthScore };

export default connect(mapStateToProps, mapDispatchToProps)(HealthScore);

```

Παρακάτω παρατίθεται ο κώδικας των actions του health score:

```

import {
  HEALTH_SCORES_LOADING,
  HEALTH_SCORES_LOADED,
  HEALTH_SCORES_FAIL,
  CREATE_HEALTH_SCORE_SUCCESS,
  CREATE_HEALTH_SCORE_FAIL,
  LATEST_HEALTH_SCORE_LOADING,
  LATEST_HEALTH_SCORE_LOADED,
  LATEST_HEALTH_SCORE_FAIL
} from "../actions/types";

const initialState = {
  isLoading: false,
  healthScores: [],
  isLoadingLatest: false,
  latestHealthScore: {}
};

export default function(state = initialState, action) {
  switch (action.type) {
    case HEALTH_SCORES_LOADING:
      return {
        ...state,
        isLoading: true
      };
    case HEALTH_SCORES_LOADED:
      return {

```

```

        ...state,
        isLoading: false,
        healthScores: action.payload
    };
    case HEALTH_SCORES_FAIL:
        return {
            ...state,
            isLoading: false,
            healthScores: []
        };
    case LATEST_HEALTH_SCORE_LOADING:
        return {
            ...state,
            isLoadingLatest: true
        };
    case LATEST_HEALTH_SCORE_LOADED:
        return {
            ...state,
            isLoadingLatest: false,
            latestHealthScore: action.payload
        };
    case LATEST_HEALTH_SCORE_FAIL:
        return {
            ...state,
            isLoadingLatest: false,
            latestHealthScore: []
        };
    case CREATE_HEALTH_SCORE_SUCCESS:
    case CREATE_HEALTH_SCORE_FAIL:
    default:
        return state;
    }
}

```

Παρακάτω παρατίθεται ο κώδικας των actions του health score:

```

import axios from "axios";
import { createMessage, returnErrors } from "../messages";
import { tokenConfig } from "../user";
import {
    HEALTH_SCORES_LOADING,
    HEALTH_SCORES_LOADED,
    HEALTH_SCORES_FAIL,
    CREATE_HEALTH_SCORE_SUCCESS,
    CREATE_HEALTH_SCORE_FAIL,
    LATEST_HEALTH_SCORE_LOADING,
    LATEST_HEALTH_SCORE_LOADED,
    LATEST_HEALTH_SCORE_FAIL
} from "../types";

// GET LATEST HEALTH SCORE
export const getLatestHealthScore = () => (dispatch, getState) => {
    dispatch({ type: LATEST_HEALTH_SCORE_LOADING });
    axios
        .get("/app/latest/health/score/", tokenConfig(getState))
        .then(res => {
            dispatch({
                type: LATEST_HEALTH_SCORE_LOADED,
                payload: res.data
            });
        })
        .catch(err => {
            dispatch(returnErrors(err.response.data, err.response.status));
            dispatch({
                type: LATEST_HEALTH_SCORE_FAIL
            });
        });
});

//GET HISTORY OF HEALTH SCORES
export const getHealthScores = () => (dispatch, getState) => {
    dispatch({ type: HEALTH_SCORES_LOADING });
    axios
        .get("/app/health/score/", tokenConfig(getState))
        .then(res => {

```

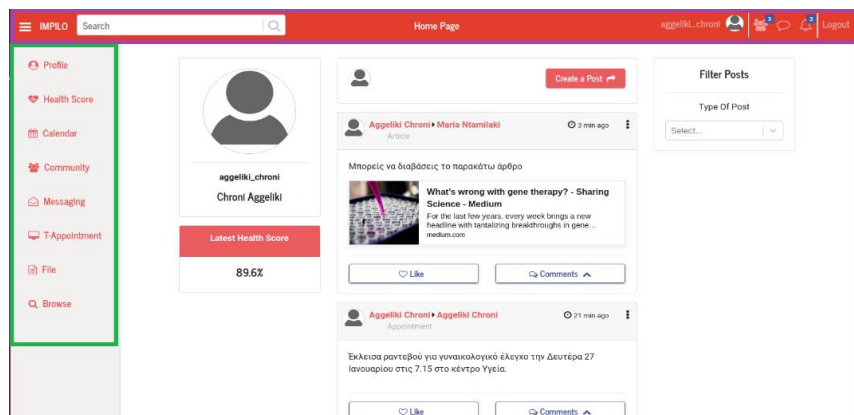
```

    dispatch({
      type: HEALTH_SCORES_LOADED,
      payload: res.data
    });
  });
  .catch(err => {
    dispatch(returnErrors(err.response.data, err.response.status));
    dispatch({
      type: HEALTH_SCORES_FAIL
    });
  });
};

//CREATE A HEALTH SCORE
export const createHealthScore = healthscore => (dispatch, getState) => {
  axios
    .post("/app/health/score/", healthscore, tokenConfig(getState))
    .then(res => {
      dispatch(
        createMessage({ successMessage: "Health Score for today submitted" })
      );
      dispatch({
        type: CREATE_HEALTH_SCORE_SUCCESS
      });
      dispatch(getHealthScores());
      dispatch(getLatestHealthScore());
    })
    .catch(err => {
      dispatch(returnErrors(err.response.data, err.response.status));
      dispatch({
        type: CREATE_HEALTH_SCORE_FAIL
      });
    });
};
};

```

Βασική μέριμνα του προγραμματισμού των οθονών της εφαρμογής αποτέλεσε η χρήση οικουμενικών διεπαφών. Η δυνατότητα επαναληψιμότητας ορισμένων προγραμματιστικών τεχνικών, συνέβαλλε στην απλοποίηση της διαδικασίας υλοποίησης και την ευκολία χειρισμού των διεπιφανειών. Τα κυριότερα χαρακτηριστικά των διεπαφών αυτών είναι η χρήση ενός κεντρικού πλοηγητή για την περιήγηση στις οθόνες της εφαρμογής, η χρήση επικεφαλίδων με τίτλο για την κάθε οθόνη και η χρήση της κεντρικής πάνω μπάρας για την αναζήτηση και την προβολή των ειδοποιήσεων.



Εικόνα 19. Απεικόνιση του κεντρικού πλοηγητή (με πράσινη περιγράμμιση) και της κεντρικής πάνω μπάρας με (μωβ περιγράμμιση)

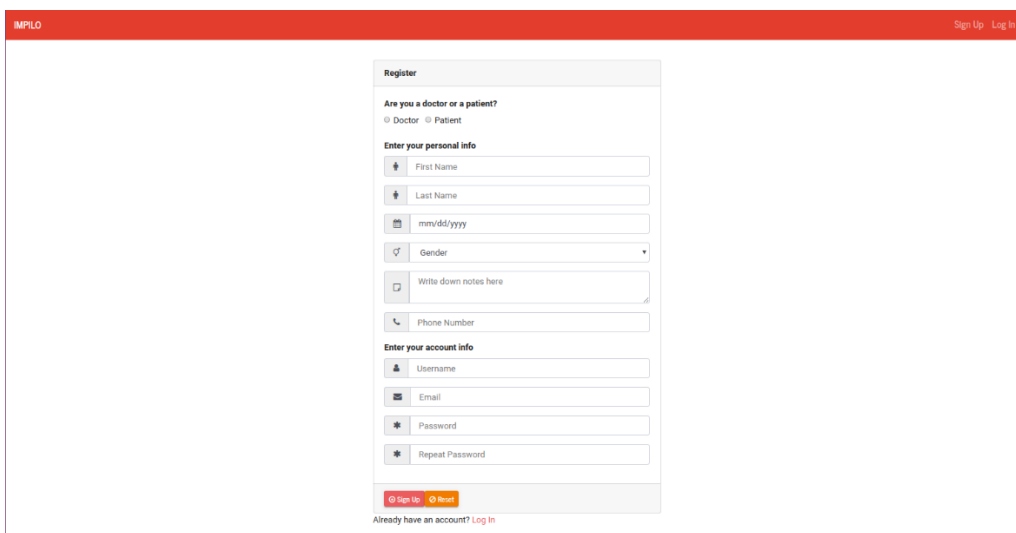
5.2.3 Το τελικό σύστημα

Το τελικό πρόγραμμα φαίνεται παρακάτω. Η αρχική σελίδα της εφαρμογής και η πρώτη οθόνη που βλέπει ο χρήστης όταν εισέρχεται στην εφαρμογή είναι η ακόλουθη:



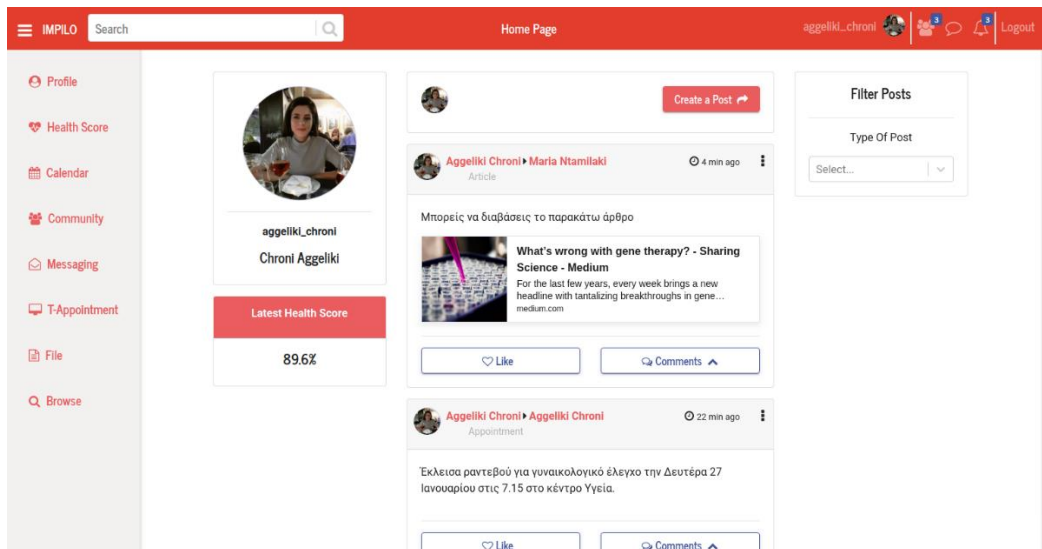
Εικόνα 20. Οθόνη Σύνδεσης

Εάν ο χρήστης δεν έχει λογαριασμό, μπορεί να δημιουργήσει συμπληρώνοντας την παρακάτω φόρμα.



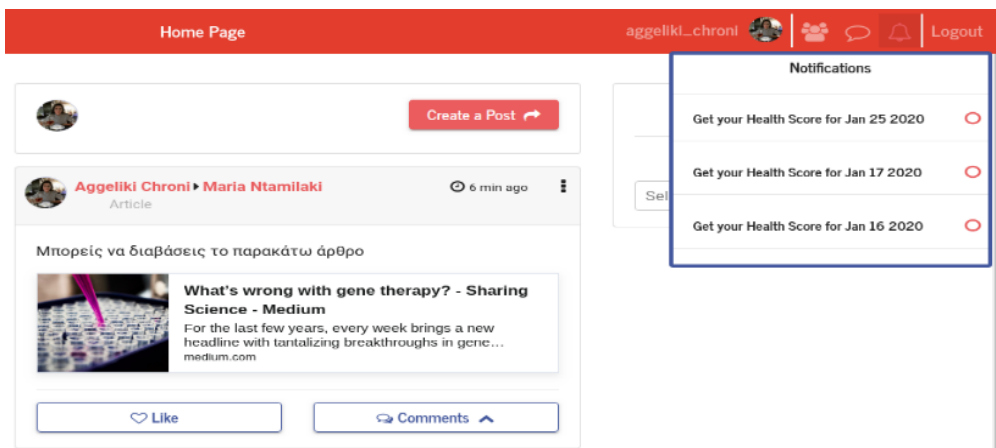
Εικόνα 21. Οθόνη Εγγραφής

Στην κεντρική σελίδα (Home Page), ο χρήστης μπορεί να δει μια προεπισκόπηση του προφίλ όπως επίσης και το πιο πρόσφατο σκορ υγείας που έχει καταγραφεί. Ακόμη, μπορεί να περιηγηθεί στο Χρονολόγιο του και να εφαρμόσει το φίλτρο αναζήτησης των δημοσιεύσεων ανάλογα με τον τύπο της δημοσίευσης.

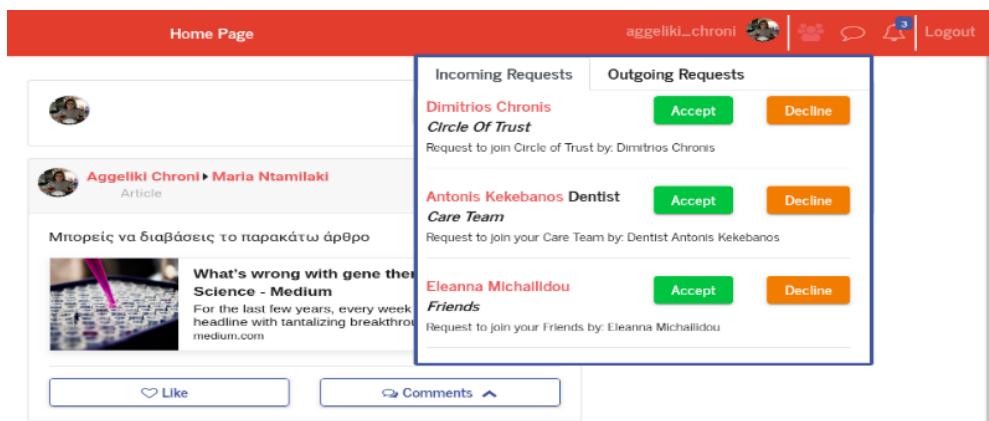


Εικόνα 22. Η κεντρική σελίδα της εφαρμογής

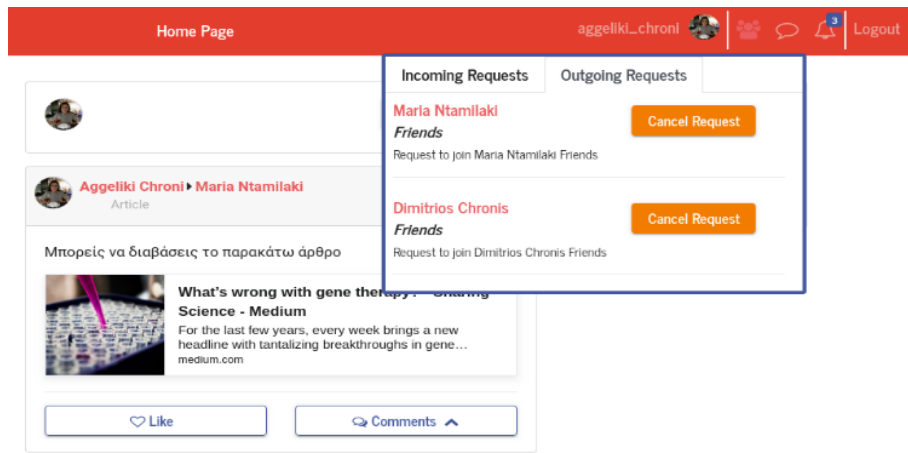
Επιπρόσθετα, στην κεντρική σελίδα προβάλλονται οι ειδοποιήσεις σχετικά με τα αιτήματα φίλιας, όπως επίσης και οι υπενθυμίσεις για την υποβολή του σκορ υγείας.



Εικόνα 23. Προβολή των Υπενθυμίσεων

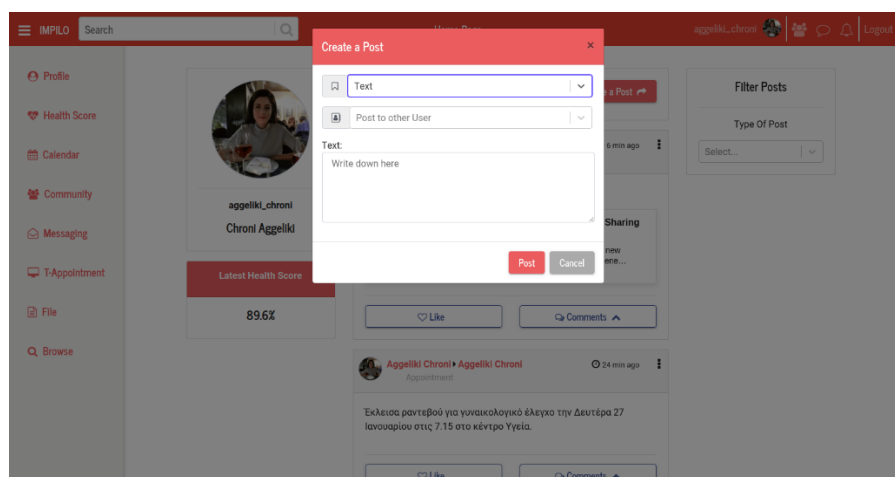


Εικόνα 24. Προβολή των εισερχόμενων αιτημάτων



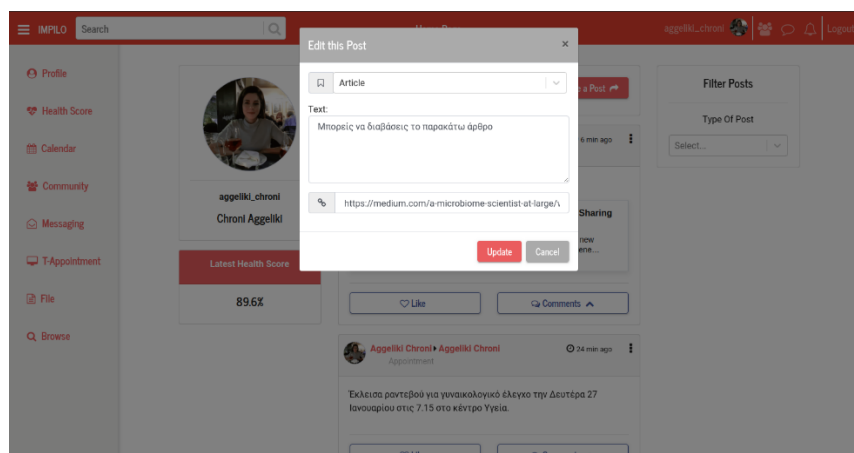
Εικόνα 25. Προβολή των εξερχόμενων αιτημάτων

Ο χρήστης για την δημιουργία νέας δημοσίευσης, θα πρέπει να συμπληρώσει την παρακάτω φόρμα.



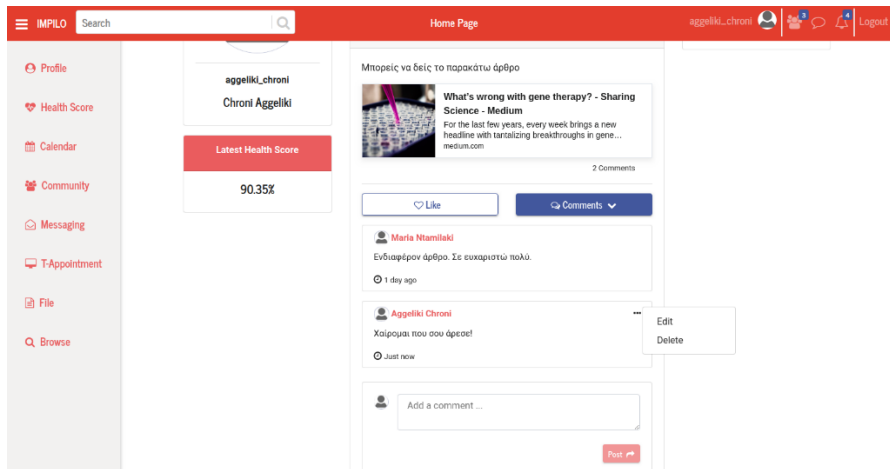
Εικόνα 26. Αναδύμενο παράθυρο δημιουργίας δημοσίευσης

Ο χρήστης μπορεί να επεξεργαστεί μια δημοσίευση του συμπληρώνοντας τη καρτέλα που θα του εμφανιστεί.

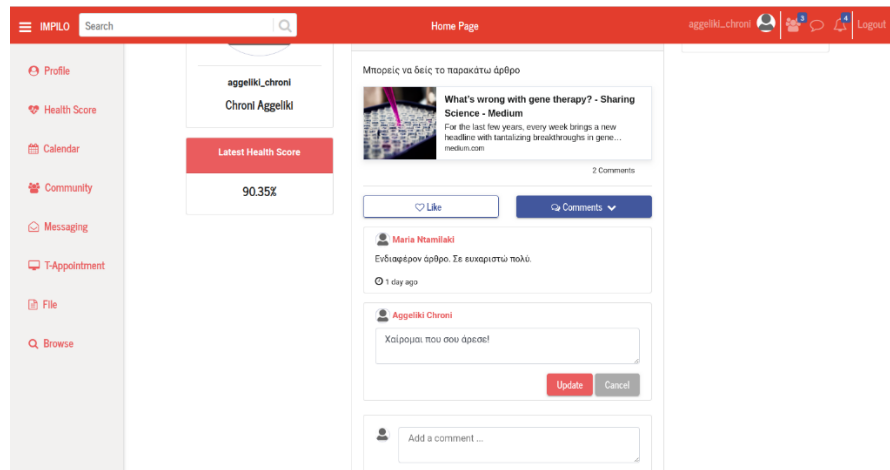


Εικόνα 27. Αναδύμενο παράθυρο επεξεργασίας δημοσίευσης

Επίσης, ανοίγοντας την καρτέλα των σχολίων μιας δημοσίευσης, ο χρήστης μπορεί είτε να προσθέσει ένα σχόλιο, είτε να επεξεργαστεί κάποιο ήδη υπάρχων σχόλιο του.

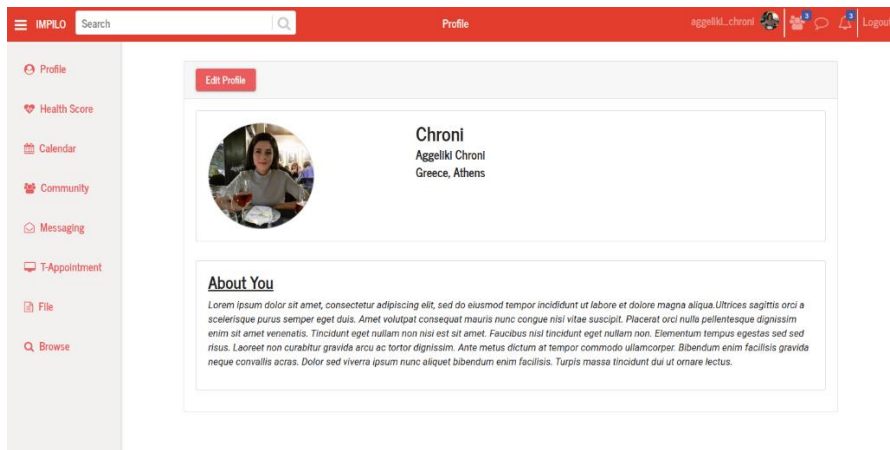


Εικόνα 28. Προβολή των σχολίων

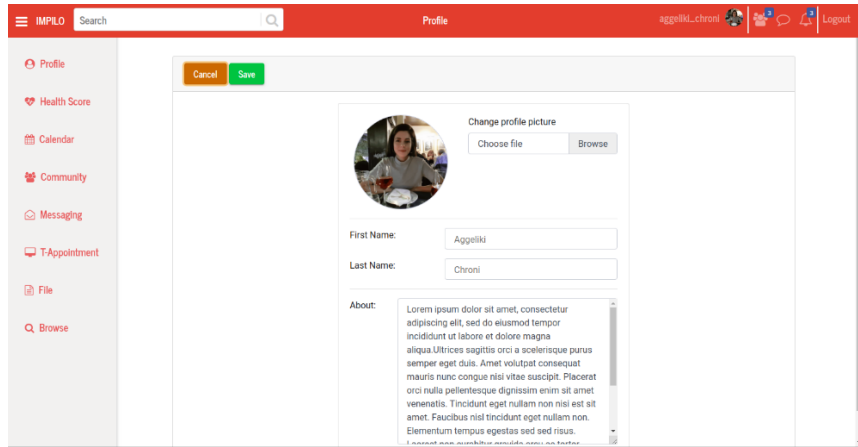


Εικόνα 29. Επεξεργασία Σχολίου

Επιπλέον, ο χρήστης μπορεί να περιηγηθεί μέσω του κεντρικού μενού στη οθόνη προσωπικού του λογαριασμού και να επεξεργαστεί τις πληροφορίες του.



Εικόνα 30. Προβολή προφίλ



Εικόνα 31. Επεξεργασία προφίλ

Η οθόνη σχετικά με τα στατιστικά δεδομένα του σκορ υγείας απεικονίζεται στην ακόλουθη εικόνα.



Εικόνα 32. Διάγραμμα ιστορικού των σκορ υγείας



Εικόνα 33. Στατιστικές πίτες σχετικά με τα ποσοστά των απαντήσεων

Για την δημιουργία του σκορ υγείας, απαιτείται η συμπλήρωση ενός ερωτηματολογίου των πέντε ερωτήσεων, το οποίο βασίζεται στο πρότυπο EQ-5D 3L. Η φόρμα η οποία πρέπει να συμπληρωθεί για την καταχώρηση του σκορ υγείας είναι η εξής:

Questions

MOBILITY

- I have no problems in walking about
- I have some problems in walking about
- I am confined to bed

SELF-CARE

- I have no problems with self-care
- I have some problems washing or dressing myself
- I am unable to wash or dress myself

USUAL ACTIVITIES

- I have no problems with performing my usual activities
- I have some problems with performing my usual activities
- I am unable to perform my usual activities

PAIN/DISCOMFORT

- I have no pain or discomfort
- I have moderate pain or discomfort
- I have extreme pain or discomfort

ANXIETY/DEPRESSION

- I am not anxious or depressed
- I am moderately anxious or depressed
- I am extremely anxious or depressed

Submit Cancel

Εικόνα 34. Ερωτηματολόγιο για το σκορ υγείας

Ο χρήστης μπορεί να σημειώσει ηλεκτρονικά τα ραντεβού του μέσω του ηλεκτρονικού του Ημερολογίου, ώστε να ανατρέξει σε αυτά ανά πάσα στιγμή.

Calendar

Today	Back	Next	January 2020						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	04		
29	30	31	01	02	03	04			
05	06	07	08	09	10	11			
12	13	14	15	16	17	18			
19	20	21	22	23	24	25			
26	27	28	29	30	31	01			

Upcoming Events

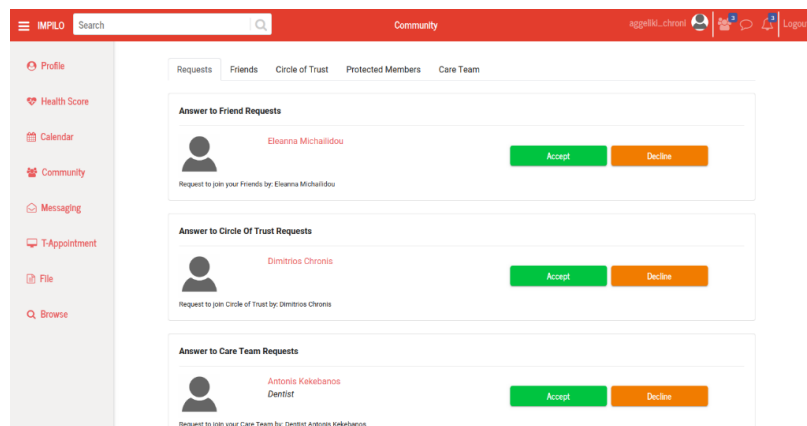
Today
Sat Jan 25 2020
2:12:28 PM - 8:12:28 PM

Point in Time Event
Sat Jan 25 2020
5:12:28 PM - 5:12:28 PM

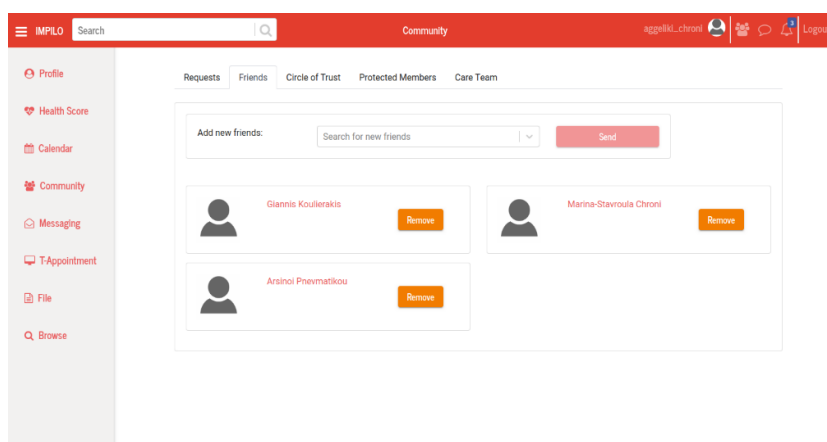
Upcoming Teleappointment

Εικόνα 35. Οθόνη Ημερολογίου

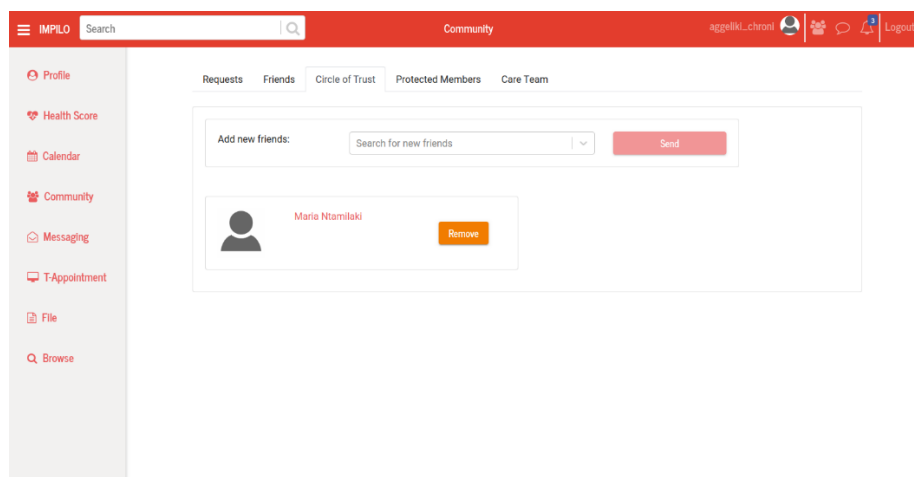
Ο χρήστης μπορεί να περιηγηθεί μέσω του κεντρικού μενού στη οθόνη του «Community», έτσι ώστε να προβάλλει τους φίλους του ή τα άτομα που ανήκουν στο «Care Team» ή στο «Circle Of Trust» του και να επεξεργαστεί τα αιτήματα που έχει λάβει.



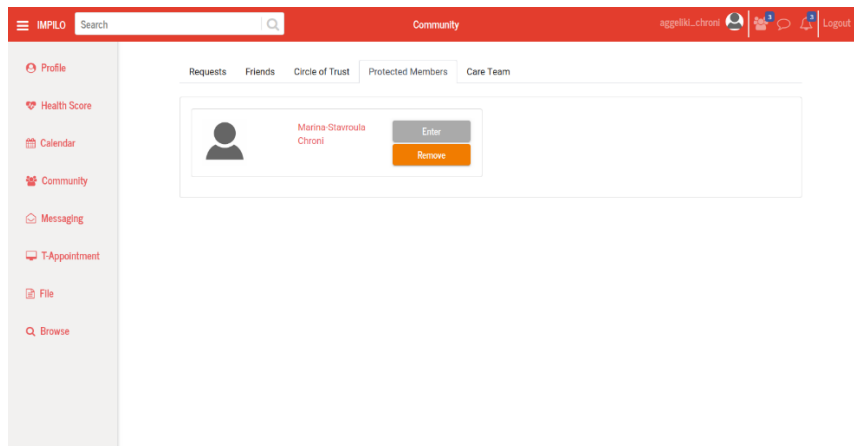
Εικόνα 36. Προβολή των αιτημάτων



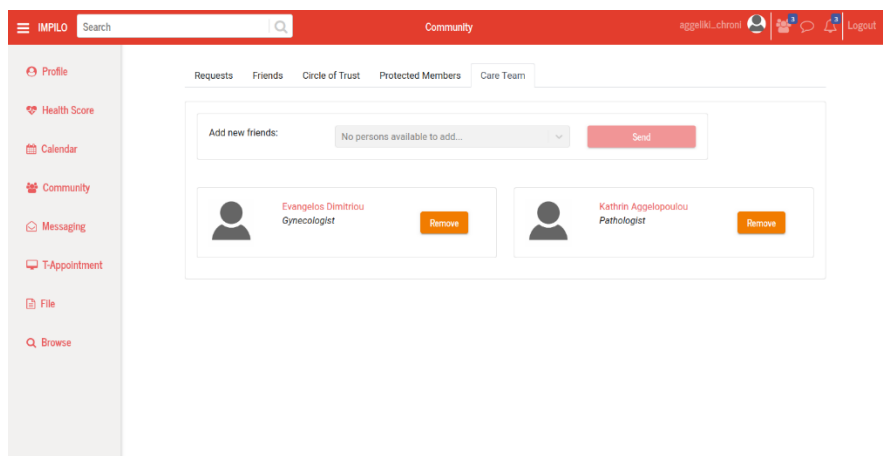
Εικόνα 37. Προβολή των Φίλων



Εικόνα 38. Προβολή των «Circle of Trust»

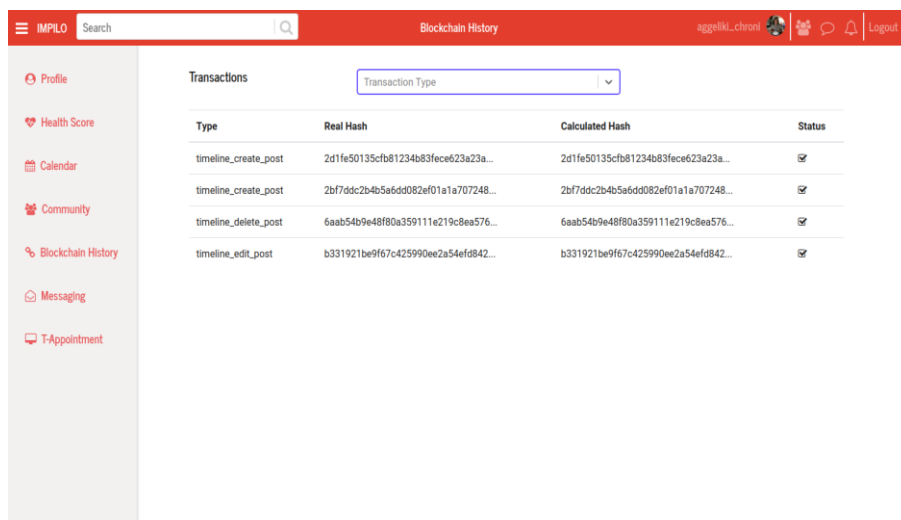


Εικόνα 39. Προβολή των «Protected Members»



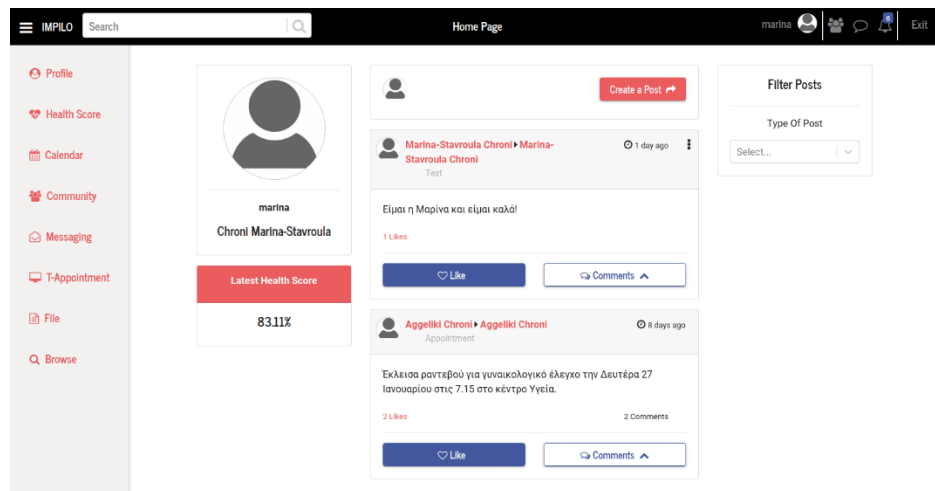
Εικόνα 40. Προβολή των «Care Team»

Ο χρήστης μπορεί να περιηγηθεί μέσω του κεντρικού μενού στη οθόνη του «Blockchain History», έτσι ώστε να προβάλλει όλα τα hashes των transactions που έχουν δημιουργηθεί και σχετίζονται με τα posts.



Εικόνα 41. Προβολή του «Blockchain History»

Ο χρήστης μπορεί να συνδεθεί πατώντας το κουμπί «Enter» στον λογαριασμό του προστατευόμενου μέλους και να περιηγηθεί σε όλες τις οθόνες της εφαρμογής.



Εικόνα 42. Προβολή του λογαριασμού του προστατευόμενου μέλους

Κεφάλαιο 6. Επίλογος

6.1 Σύνοψη

Η παρούσα διπλωματική εργασία επιχειρήσε να προσεγγίσει τις εξελικτικές ανάγκες της ψηφιοποίησης στον τομέα της υγείας και να βελτιώσει την ποιότητα υπηρεσιών που αυτή προσφέρει στους χρήστες. Το project Impilo αποτελεί μια επιτυχημένη προσπάθεια δημιουργίας ενός ηλεκτρονικού φακέλου υγείας , του οποίου η ασφάλεια και η εγκυρότητα των δεδομένων είναι εξαρχής εδραιωμένα. Το τελευταίο αυτό στοιχείο σε συνδυασμό με τις διεπαφές που είναι φιλικές προς τον χρήστη, καθιστά το Impilo αρκετά αποδοτικό προκειμένου να υποστηρίξει μεγάλο πλήθος χρηστών.

6.2 Μελλοντικοί Στόχοι

Το Web App του Impilo μπορεί να ενισχυθεί με νέες λειτουργικότητες, που θα αξιοποιήσουν τόσο οι ασθενείς όσο και οι ιατροί και θα τους προσφέρει μια πιο ολοκληρωμένη εμπειρία χρήσης. Μια από αυτές είναι η λειτουργία των ηλεκτρονικών ραντεβού, όπου η εφαρμογή Impilo θα δίνει την δυνατότητα βιντεοκλήσης μεταξύ των ασθενών και των ιατρών. Ο χρήστης θα μπορεί να κλείσει ηλεκτρονικά ένα απομακρυσμένο ραντεβού με τον ιατρό που επιθυμεί και ο ιατρός θα μπορεί να δεχτεί και να πληρωθεί για αυτήν την εξέταση μέσω της εφαρμογής. Επιπρόσθετα, θα μπορούσε να ενσωματωθεί στις δυνατότητες του Impilo, η ηλεκτρονική συνταγογράφηση ώστε οι ασθενείς με σοβαρά κινητικά προβλήματα να μπορούν να εξυπηρετηθούν άμεσα. Έχει δειχθεί πως η ψηφιοποίηση στον τομέα της υγείας μπορεί να παρακάμψει εύκολα προβλήματα γραφειοκρατίας και να διευκολύνει τόσο το ιατρικό προσωπικό όσο και ερευνητές διαφορετικών ειδικοτήσεων να έχουν άμεσα όλες τις ιατρικές πληροφορίες των ασθενών τους.

Βιβλιογραφία

- Attili S, L. S. (2016). Blockchain: the chain of trust and its potential to transform healthcare – our point of view. *ONC/NIST Use of Blockchain for Healthcare and Research Workshop*.
- Azaria A, E. A. (2016). MedRec: using blockchain for medical data access and permission management. *International Conference on Open and Big Data (OBD)*, σσ. 25-30.
- Baliga, D. A. (2017). Understanding Blockchain. Ανάκτηση από <https://pdfs.semanticscholar.org/da8a/37b10bc1521a4d3de925d7ebc44bb606d740.pdf>
- Bender JL, J.-M. M. (2011 , Φεβρουάριος 4). Seeking support on facebook: a content analysis of breast cancer groups. *PubMed Central (PMC)* , σ. 16.
- Bitcoin Wiki. *Elliptic Curve Digital Signature*. (χ.χ.). Ανάκτηση από https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm
- Blockchain Public / Private Key Cryptography In A Nutshell. (χ.χ.). Ανάκτηση από <https://medium.com/coinmonks/blockchain-public-private-key-cryptography-in-a-nutshell-b7776e475e7c>
- Blough D, A. M. (2008). MedVault: Ensuring security and privacy for electronic medical records. *NSF CyberTrust Principal Investigators Meeting*.
- C B, K. B. (2016). Blockchain: securing a new health interoperability experience. *ONC/NIST Use of Blockchain for Healthcare and Research Workshop*.
- D, I. (2016). Moving Toward a Blockchain-based Method for the Secure Storage of Patient Records. *ONC/NIST Use of Blockchain for Healthcare and Research Workshop*.
- Different Blockchain Consensus Mechanisms - By Blockgenic*. (χ.χ.). Ανάκτηση από <https://hackernoon.com/different-blockchain-consensus-mechanisms-d19ea6c3bcd6>
- Digital Signature Standard (DSS). (2017, Απρίλιος 6). *FIPS PUB 186-4*, . Ανάκτηση από <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>
- EJ, T. (2016). Money back guarantees for non-reproducible results. *BMJ*.
- Ekblaw A, A. A. (2016). A Case Study for Blockchain in Healthcare: “MedRec” Prototype for Electronic Health Records and Medical Research Data . *Use of Blockchain for Healthcare and Research Workshop*.
- Fedor Muratov, A. L. (2018, September 3). YAC: BFT Consensus Algorithm for Blockchain. Ανάκτηση από <https://arxiv.org/pdf/1809.00554.pdf>
- Gareth, B. (2016). Can Blockchain Revolutionise EPRs? *ITNOW(581)*, σσ. 38-39.
- Google Trends - Ενδιαφέρον για Αναζήτηση Ιστού: *adobe flex, extjs, gwt, vaadin - Παγκοσμίως, 2004 - παρόν*,. (χ.χ.). Ανάκτηση από <http://www.google.com/trends/explore#q=adobe++flex,extjs,gwt,vaadin>
- Google Trends - Ενδιαφέρον για Αναζήτηση Ιστού: *adobe flex, microsoft silverlight, java applet, html5, gwt - Παγκοσμίως, 2004 - παρόν*,. (χ.χ.). Ανάκτηση από <http://www.google.com/trends/explore#q=adobe++flex,microsoft+silverlight,java++applet,HTML5,gwt>
- Guardtime Industrial Blockchain*. (χ.χ.). Ανάκτηση από [https://guardtime.com/Hashing in Blockchain explained | Online Hash Crack](https://guardtime.com/Hashing%20in%20Blockchain%20explained%20|%20Online%20Hash%20Crack). (χ.χ.). Ανάκτηση από <https://www.onlinehashcrack.com/how-to-hashing-in-blockchain-explained.php>

- Health / Gem.* (χ.χ.). Ανάκτηση από Gem Health Network: <https://gem.co/health/>
- Healthbank.coop.* (χ.χ.). Ανάκτηση από HealthBank: <https://www.healthbank.coop/>
- HTML - Wikipedia, the free encyclopedia.* (χ.χ.). Ανάκτηση από <http://en.wikipedia.org/wiki/HTML>
- Hunt, B. (χ.χ.). *How HTML, CSS and JavaScript work together in making web pages - guide to using xHTML, HTML, Javascript and CSS together.* Ανάκτηση από <http://webdesignfromscratch.com/html-css/how-html-css-js-work-together/>
- Hyperledger Iroha documentation.* (χ.χ.). Ανάκτηση από <https://iroha.readthedocs.io/en/latest/>
- K, C. (2016). Blockchain Technologies: A whitepaper discussing how the claims process can be improved. *ONC/NIST Use of Blockchain for Healthcare and Research Workshop.*
- K, Y. (2016). Blockchain & alternate payment models. *ONC/NIST Use of Blockchain for Healthcare and Research Workshop.*
- Krawiec R, B. D. (2016). Blockchain: opportunities for health care. *ONC/NIST Use of Blockchain for Healthcare and Research Workshop.*
- Kuo T-T, H. C.-N.-M. (2016). ModelChain: Decentralized Privacy-Preserving Healthcare Predictive Modeling Framework on Private Blockchain Networks. (O. U. Workshop, Επιμ.)
- L., M. (2017, Απρίλιος 6). *Blockchain vs. Relational Database: Which is right for your Application?* Ανάκτηση από TechBeacon: <https://techbeacon.com/Blockchain-relational-database-which-right-for-your-application>
- Linn LA, K. M. (2016). Blockchain for health data and its potential use in health IT and health care related research. *ONC/NIST Use of Blockchain for Healthcare and Research Workshop.*
- Matthias, M. (2016). Blockchain technology in healthcare: The revolution starts here. *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, σσ. 1-3.
- McConaghy T, M. R. (2016, Ιούλιος 30). BigchainDB: A Scalable Blockchain Database. Ανάκτηση από <https://www.bigchaindb.com/whitepaper/>
- P, T. (χ.χ.). *SecuringIndustry.com - Applying blockchain technology to medicine traceability.* Ανάκτηση από SecuringIndustry.com.
- Peterson K, D. R. (2016). A blockchain-based approach to health information exchange networks. *ONC/NIST Use of Blockchain for Healthcare and Research Workshop.*
- Props, R. N. (χ.χ.). Ανάκτηση από <https://facebook.github.io/react-native/docs/props.html>
- React - Introducing JSX.* (χ.χ.). Ανάκτηση από <https://reactjs.org/docs/introducing-jsx.html>
- React (JavaScript library) - Wikipedia.* (χ.χ.). Ανάκτηση από [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))
- Redux (JavaScript library) - Wikipedia.* (χ.χ.). Ανάκτηση από [https://en.wikipedia.org/wiki/Redux_\(JavaScript_library\)](https://en.wikipedia.org/wiki/Redux_(JavaScript_library))
- Redux · A Predictable State Container for JS Apps.* (χ.χ.). Ανάκτηση από <https://redux.js.org/>
- Rosic, A. (2017). What Is Hyperledger? [The Most Comprehensive Step-by-Step-Guide!]. Ανάκτηση από <https://blockgeeks.com/guides/ethereum/>
- S. J. Alsunaidi, F. A. (2019). A Survey of Consensus Algorithms for Blockchain Technology,. *IEEE.*

- S., M. (2017, April 6). *Blockchain Technology— a Very Special Kind of Distributed Database*. Ανάκτηση από Medium: <https://medium.com/@sbmeunier/blockchain-technology-a-very-special-kind-of-distributed-database-e63d00781118>
- Sarwal A, I. P. (χ.χ.). *BitHealth / Devpost*. Ανάκτηση από <https://devpost.com/software/bithealth>
- Secure Hash Standard (SHS). (2017, Απρίλιος 6). *FIPS PUB 180-4*. Ανάκτηση από <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>
- ServiceWorker API - Web API Interfaces / MDN*. (χ.χ.). Ανάκτηση από https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API
- SHA-256 - Bitcoin Wiki*. (χ.χ.). Ανάκτηση από <https://en.bitcoin.it/wiki/SHA-256>
- SharedWorker - Web API Interfaces / MDN*. (χ.χ.). Ανάκτηση από <https://developer.mozilla.org/en-US/docs/Web/API/SharedWorker>
- Snow P, D. B. (2015). *Ledger by Consensus*. *Factom*. Ανάκτηση από <https://www.factom.com/>
- Social Networks in Improvement of Health Care*. (χ.χ.). Ανάκτηση από <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3732339/>
- Syngress, C. T. (2010). *Seven Deadliest Social Network Attacks*. Elsevier Inc.
- Thomas, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine.
- Tsung-Ting Kuo, H.-E. K.-M. (2017, Σεπτέμβριος 08). *Blockchain distributed ledger technologies for biomedical and health care applications*. *Journal of the American Medical Informatics Association*, σσ. 1211–1220.
- Use of Blockchain in Healthcare and Research Workshop - ONC Tech Lab Innovation - Confluence*. (2016, Δεκέμβριος 20). Ανάκτηση από <https://oncprojecttracking.healthit.gov/wiki/display/TechLabI/Use+of+Blockchain+in+Healthcare+and+Research+Workshop>
- Vian K, V. A.-S. (2016). *A blockchain profile for medicaid applicants and recipients*. *ONC/NIST Use of Blockchain for Healthcare and Research Workshop*.
- W3Counter: Free Web Stats and Website Widgets*. (χ.χ.). Ανάκτηση από www.w3counter.com
- Wasserman S., F. K. (1994). *Social Network Analysis: Methods and Applications*. *Web Services Addressing (WS-Addressing)*. (χ.χ.). Ανάκτηση από <https://www.w3.org/Submission/ws-addressing/>
- What is a RESTful API (REST API) and How Does it Work?* (χ.χ.). Ανάκτηση από <https://searcharchitecture.techtarget.com/definition/RESTful-API>
- What is npm?* (χ.χ.). Ανάκτηση από <https://docs.npmjs.com/gettingstarted/what-is-npm>
- WindowTimers.setInterval() - Web API Interfaces / MDN*. (χ.χ.). Ανάκτηση από <https://developer.mozilla.org/en-US/docs/Web/API/WindowOrWorkerGlobalScope/setInterval>
- WindowTimers.setTimeout() - Web API Interfaces*. (χ.χ.). Ανάκτηση από <https://developer.mozilla.org/en-US/docs/Web/API/WindowOrWorkerGlobalScope/setTimeout>
- World Wide Web Consortium (W3C)*. (χ.χ.). Ανάκτηση από <http://www.w3.org/>
- XMLHttpRequest - Wikipedia, the free encyclopedia*. (χ.χ.). Ανάκτηση από <https://en.wikipedia.org/wiki/XMLHttpRequest>
- Y. Xiao, N. Z. (2019). *A Survey of Distributed Consensus Protocols for Blockchain*. Ανάκτηση από <https://arxiv.org/pdf/1904.04098.pdf>

Yue X, W. H. (2016, Οκτώβριος). Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *J Med Syst*.
Ανάκτηση από <https://www.ncbi.nlm.nih.gov/pubmed/27565509>
Μέσα κοινωνικής δικτύωσης - Βικιπαίδεια. (χ.χ.). Ανάκτηση από
https://el.wikipedia.org/wiki/Μέσα_κοινωνικής_δικτύωσης