



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ CLOUD-NATIVE
ΕΦΑΡΜΟΓΗΣ ΙΑΤΡΙΚΟΥ ΦΑΚΕΛΟΥ ΜΕ ΧΡΗΣΗ
MySQL

Ζωή Γεωργίου

Επιβλέπων Καθηγητής: Δημήτριος Δημόπουλος, ΠΥ

Αθήνα, Οκτώβριος, 2021

**DESIGN AND DEVELOPMENT OF CLOUD-NATIVE
MEDICAL RECORD APPLICATION USING MySQL**

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Αρτα, 01/11/2021

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής
Δημήτριος Δημόπουλος, ΠΥ
2. Μέλος επιτροπής
Νικόλαος Γιαννακέας, ΔΕΠ, Επίκουρος Καθηγητής
3. Μέλος επιτροπής
Αλέξανδρος Τζάλλας, ΔΕΠ, Επίκουρος Καθηγητής

© Γεωργίου, Ζωή, 2021.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Γεωργίου Ζωή

Υπογραφή

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κ. Δημήτριο Δημόπουλο για την εμπιστοσύνη που μου έδειξε, την υπομονή που έκανε κατά τη διάρκεια υλοποίησης της πτυχιακής εργασίας, και για την πολύτιμη βοήθεια και καθοδήγησή του.

Θα ήθελα επίσης να απευθύνω τις ευχαριστίες μου στους γονείς μου, οι οποίοι στήριξαν τις σπουδές μου με διάφορους τρόπους, φροντίζοντας για την καλύτερη δυνατή μόρφωση μου.

Τέλος, στους φίλους και τους συνάδελφούς μου, που άντεξαν όλα τα ψυχολογικά скаμπανεβάσματα που είχα καθ' όλη την διάρκεια της εκπόνηση της παρούσας εργασίας.

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας πτυχιακής εργασίας είναι να παρουσιάσει την ιδέα του Ηλεκτρονικού Φακέλου Ασθενούς καθώς και τα πλεονεκτήματα αλλά και τις προκλήσεις που εμφανίζονται κατά την εφαρμογή του. Θα παρουσιαστεί η υλοποίηση του Ελληνικού συστήματος καθώς και προτάσεις άλλων ιδιωτικών εταιριών του εξωτερικού. Τέλος, θα ακολουθήσει μια εξομοίωση της λειτουργικότητάς του μέσω της κατασκευής μιας εφαρμογής που θα παρουσιάζει το σύνολο των τεχνολογιών που εμπλέκονται στην υλοποίηση της εφαρμογής.

Στο πρώτο μέρος της εργασίας, θα παρουσιαστεί το ήδη διαφοροποιημένο τοπίο από πλευράς Ελληνικού νομικού πλαισίου, καθώς και διαφόρων ενδεικτικών προτάσεων υλοποίησης. Θα παρουσιαστεί το Ελληνικό σύστημα όπως επίσης και άλλες λύσεις ιδιωτικών εταιριών του εξωτερικού.

Στο δεύτερο μέρος θα αναφερθούν οι τεχνολογίες που χρησιμοποιούνται κατά την υλοποίηση της εφαρμογής. Θα αναπτυχθούν έννοιες που σχετίζονται με την αποθήκευση των δεδομένων στις βάσεις δεδομένων, θέματα που σχετίζονται με την υπολογιστική νέφους, που αποτελεί μια από τις πλέον καταξιωμένες τεχνικές και εφαρμόζεται κατά στη πλειονότητα των υπολογιστικών συστημάτων, καθώς και στις επιμέρους τεχνολογίες που ενορχηστρώνουν την εφαρμογή.

Στο τρίτο μέρος θα παρουσιαστεί η υλοποίηση της εφαρμογής. Συγκεκριμένα, θα παρουσιαστεί η ανάλυση απαιτήσεων του συστήματος, η βάση δεδομένων καθώς και η αρχιτεκτονική του. Τα βήματα της υλοποίησης θα αναπτυχθούν αναλυτικά, ώστε να εξοικειωθεί ο αναγνώστης με τις τεχνικές ανάπτυξης των εφαρμογών ιστού.

Τέλος, το τέταρτο και τελευταίο κεφάλαιο της εργασίας, θα αναφερθεί σε μελλοντικές προσθήκες που στοχεύουν στη βελτίωση της λειτουργικότητας της εφαρμογής, τα σημεία που δυσκόλεψαν την συγγραφέα καθώς και σε μελλοντικές προκλήσεις που απαιτούν την προσοχή των ερευνητών.

Λέξεις-κλειδιά: Ιατρικός Φάκελος Ασθενούς, Υπολογιστική Νέφους, Συστήματα Διαχείρισης Βάσεων Δεδομένων, PHP, MySQL, JavaScript.

ABSTRACT

The purpose of this dissertation is to present the idea of the Electronic Patient File as well as the advantages and challenges that arise during its implementation. The implementation of the Greek system will be presented as well as proposals of other companies abroad. Finally, a simulation of its functionality will follow through the development of an application that will present all the technologies involved in the implementation of the application.

In the first part of the thesis, there will be a presentation of the already differentiated landscape in terms of the Greek legal framework as well as various indicative implementation proposals. The Greek system will be presented as well as other solutions of private companies abroad.

The second part will refer to the technologies used during the implementation of the application. Concepts related to the storage of data in databases, issues related to cloud computing, which is one of the most popular techniques and is applied in most computer systems, as well as in the individual technologies that orchestrate the application will be presented.

The third part will present the implementation of the application. The system requirements analysis, the database as well as its architecture will be presented. The whole presentation will include in detail the implementation steps to familiarize the reader with web application development techniques.

Finally, in the fourth and last chapter of the work, future additions will be mentioned that aim to improve the functionality of the application, the points that made it difficult for the authors as well as future challenges that require the attention of researchers.

Keywords: Electronic Health Record, Cloud Computing, Data Bases, PHP, MySQL, JavaScript.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ	6
ΠΕΡΙΛΗΨΗ	6
ABSTRACT	8
Πίνακας Περιεχομένων Εικόνων	11
1. Ιατρικός Φάκελος.....	13
1.1 Γέννηση Ιατρικού Φακέλου	13
1.2 Ορισμοί	14
1.3 Χαρακτηριστικά και Προϋποθέσεις Ηλεκτρονικού Ιατρικού Φακέλου	15
1.4 Πλεονεκτήματα και Μειονεκτήματα.....	15
1.5 Νομικό Πλαίσιο	16
1.6 Εφαρμογές.....	17
1.6.1 Ηλεκτρονικός Φάκελος Υγείας – ΗΔΙΚΑ.....	17
1.6.2 Microsoft HealthVault.....	23
2. Υπολογιστική Νέφος.....	25
2.1 Εισαγωγή.....	25
2.1.1 Ο Καταναλωτής (Cloud Consumer).....	26
2.1.2 Ο Πάροχος (Cloud Provider).....	28
2.1.3 Ο Σύμβουλος (Cloud Auditor)	30
2.1.4 Ο Διαμεσολαβητής (Cloud Broker)	30
2.1.5 Ο Φορέας.....	31
2.2 Google Cloud Platform	32
2.3 Δημιουργία Υποδομής	32
3. Η εφαρμογή.....	44
3.1 Περιγραφή Προβλήματος.....	44
3.2 Ανάλυση Απαιτήσεων	45
3.2.1 Λειτουργικές Απαιτήσεις	45
3.2.2 Μη Λειτουργικές Απαιτήσεις.....	46
3.3 Αρχιτεκτονική Συστήματος.....	47
3.3.1 Ο Εξυπηρετητής.....	47
3.3.2 Ο Πελάτης	48
3.3.3 Η Βάση Δεδομένων.....	49
3.4 Ο κώδικας της Εφαρμογής	50
3.4.1 Παραμετροποίηση του Web Server	50

3.4.2	Η Πύλη Εισόδου Index.php.....	51
3.4.3	Η Αρχική Σελίδα Home	52
3.4.4	Εγγραφή Χρήστη.....	53
3.4.5	Είσοδος Χρήστη.....	57
3.4.6	Ενημέρωση Στοιχείων Χρήστη	58
3.4.7	Επιλογή Ιατρών	61
3.4.8	Εξετάσεις.....	67
3.4.9	Εγγραφή Ιατρού	69
3.4.10	Είσοδος Ιατρού.....	70
3.4.11	Ενημέρωση στοιχείων Ιατρού	71
3.4.12	Προβολή Εξετάσεων Ασθενούς	73
4.	Επίλογος - Συμπεράσματα	83
4.1	Σημεία που δυσκόλεψαν την συγγραφέα	83
4.2	Μελλοντικές προσθήκες.....	84
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	85

Πίνακας Περιεχομένων Εικόνων

ΕΙΚΟΝΑ 1 - ΔΗΛΩΣΗ ΟΙΚΟΓΕΝΕΙΑΚΟΥ ΓΙΑΤΡΟΥ	17
ΕΙΚΟΝΑ 2 - ΕΙΣΑΓΩΓΗ ΣΤΟ ΣΥΣΤΗΜΑ ΜΕ ΚΩΔΙΚΟΥΣ TAXISNET. ΠΗΓΗ: ERDV_MANUAL_PUBLIC_V3.PDF	18
ΕΙΚΟΝΑ 3 - ΠΙΣΤΟΠΟΙΗΣΗ ΣΤΟΙΧΕΙΩΝ ΧΡΗΣΤΗ. ΠΗΓΗ:ERDV_MANUAL_PUBLIC_V3.PDF	19
ΕΙΚΟΝΑ 4 - ΕΠΙΒΕΒΑΙΩΣΗ ΣΤΟΙΧΕΙΩΝ. ΠΗΓΗ: ERDV_MANUAL_PUBLIC_V3.PDF	19
ΕΙΚΟΝΑ 5 - ΔΗΛΩΣΗ ΟΙΚΟΓΕΝΕΙΑΚΟΥ ΙΑΤΡΟΥ. ΠΗΓΗ: RDV_MANUAL_PUBLIC_V3.PDF	20
ΕΙΚΟΝΑ 6 - ΕΦΑΡΜΟΓΗ ΑΤΟΜΙΚΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΦΑΚΕΛΟΥ ΥΓΕΙΑΣ. ΠΗΓΗ: ERDV_MANUAL_PUBLIC_V3.PDF.....	20
ΕΙΚΟΝΑ 7 - Η ΚΑΡΤΕΛΑ ΕΠΙΣΚΕΨΕΙΣ/ΚΑΤΑΣΤΑΣΗ ΑΣΘΕΝΗ. ΠΗΓΗ:ERDV_MANUAL_PUBLIC_V3.PDF.....	21
ΕΙΚΟΝΑ 8 - ΑΤΟΜΙΚΟ ΙΣΤΟΡΙΚΟ. ΠΗΓΗ: ERDV_MANUAL_PUBLIC_V3.PDF	21
ΕΙΚΟΝΑ 9 - ΚΑΡΤΕΛΑ ΜΕ ΤΙΣ ΚΟΙΝΩΝΙΚΕΣ ΣΥΝΗΘΕΙΕΣ ΤΟΥ ΑΤΟΜΟΥ. ΠΗΓΗ:ERDV_MANUAL_PUBLIC_V3.PDF	22
ΕΙΚΟΝΑ 10 - ΟΙΚΟΓΕΝΕΙΑΚΟ ΙΣΤΟΡΙΚΟ ΑΤΟΜΟΥ. ΠΗΓΗ:ERDV_MANUAL_PUBLIC_V3.PDF.....	22
ΕΙΚΟΝΑ 11 - ΔΕΔΟΜΕΝΑ ΕΞΕΤΑΣΕΩΝ. ΠΗΓΗ: ERDV_MANUAL_PUBLIC_V3.PDF	22
ΕΙΚΟΝΑ 12 - ΤΑ ΦΑΡΜΑΚΑ ΠΟΥ ΕΧΕΙ ΛΑΒΕΙ ΤΟ ΑΤΟΜΟ. ΠΗΓΗ:ERDV_MANUAL_PUBLIC_V3.PDF.....	23
ΕΙΚΟΝΑ 13 - ΣΥΝΟΠΤΙΚΟ ΙΣΤΟΡΙΚΟ. ΠΗΓΗ : ERDV_MANUAL_PUBLIC_V3.PDF	23
ΕΙΚΟΝΑ 14 - ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΝΕΦΟΥΣ ΣΥΜΦΩΝΑ ΜΕ ΤΟ NIST (NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY) ΠΗΓΗ: HTTPS://NVLTPUBS.NIST.GOV/NISTRUBS/LEGACY/SP/NISTSPECIALPUBLICATION500-292.PDF	25
ΕΙΚΟΝΑ 15 - ΔΙΑΘΕΣΙΜΕΣ ΥΠΗΡΕΣΙΕΣ ΤΟΥ ΝΕΦΟΥΣ ΠΗΓΗ: HTTPS://NVLTPUBS.NIST.GOV/NISTRUBS/LEGACY/SP/NISTSPECIALPUBLICATION500-292.PDF	27
ΕΙΚΟΝΑ 16 - ΥΠΗΡΕΣΙΕΣ ΠΑΡΟΧΟΥ ΣΕ ΙΑΑΣ ΠΗΓΗ: HTTPS://NVLTPUBS.NIST.GOV/NISTRUBS/LEGACY/SP/NISTSPECIALPUBLICATION500-292.PDF	29
ΕΙΚΟΝΑ 17 - ΕΠΙΛΟΓΕΣ WEB HOSTING ΑΠΟ ΤΗ GOOGLE. ΠΗΓΗ : HTTPS://CLOUD.GOOGLE.COM/ARCHITECTURE/WEB-SERVING-OVERVIEW?HL=EN_US.....	33
ΕΙΚΟΝΑ 18 - Η ΠΛΑΤΦΟΡΜΑ ΤΗΣ GOOGLE ΜΕΤΑ ΤΗ ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΥΠΟΔΟΜΗΣ.....	34
ΕΙΚΟΝΑ 19 - ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΙΚΗΣ ΜΗΧΑΝΗΣ.....	35
ΕΙΚΟΝΑ 20 - ΕΠΙΠΛΕΟΝ ΕΠΙΛΟΓΕΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΤΗΣ ΕΙΚΟΝΙΚΗΣ ΜΗΧΑΝΗΣ.....	36
ΕΙΚΟΝΑ 21 - ΤΟ ΕΙΚΟΝΙΚΟ ΜΗΧΑΝΗΜΑ ΠΟΥ ΔΗΜΙΟΥΡΓΕΙΤΑΙ ΜΕΤΑ ΤΗΝ ΟΛΟΚΛΗΡΩΣΗ ΤΗΣ ΔΙΑΔΙΚΑΣΙΑΣ ΟΠΟΥ ΦΑΙΝΟΝΤΑΙ Ο ΤΥΠΟΣ, Η IP ΔΙΕΥΘΥΝΣΗ, ΚΑΙ ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΜΗΧΑΝΗΜΑΤΟΣ ..	37
ΕΙΚΟΝΑ 22 - ΔΙΑΜΟΡΦΩΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ PUTTY. ΕΙΣΑΓΩΓΗ ΤΟΥ ΟΝΟΜΑΤΟΣ ΧΡΗΣΤΗ ΣΤΗΝ ΕΦΑΡΜΟΓΗ	38
ΕΙΚΟΝΑ 23 - ΕΙΣΑΓΩΓΗ ΚΛΕΙΔΙΟΥ ΣΤΗΝ ΕΦΑΡΜΟΓΗ.....	38
ΕΙΚΟΝΑ 24 - ΣΥΝΔΕΣΗ ΜΕΣΩ SSH.....	39
ΕΙΚΟΝΑ 25 - Η ΣΕΛΙΔΑ RHPINFO ΤΟΥ ΕΞΥΠΗΡΕΤΗΤΗ	40
ΕΙΚΟΝΑ 26 - ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ FILEZILLA. ΕΜΦΑΝΙΖΟΝΤΑΙ Η ΔΙΕΥΘΥΝΣΗ ΤΟΥ ΕΞΥΠΗΡΕΤΗΤΗ ΚΑΙ ΤΟ ΟΝΟΜΑ ΧΡΗΣΤΗ.....	41
ΕΙΚΟΝΑ 27 - ΕΙΣΑΓΩΓΗ ΤΟΥ ΑΡΧΕΙΟΥ ΚΛΕΙΔΙΟΥ ΣΤΗΝ ΕΠΙΛΟΓΗ SFTP	41
ΕΙΚΟΝΑ 28 - ΔΗΜΙΟΥΡΓΙΑ ΑΣΦΑΛΟΥΣ ΤΟΥΝΕΛ ΓΙΑ ΧΡΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ RHPMYADMIN	42
ΕΙΚΟΝΑ 29 - ΣΥΝΔΕΣΗ ΣΤΗΝ ΕΦΑΡΜΟΓΗ RHPMYADMIN	43
ΕΙΚΟΝΑ 30 - ΤΟ ΜΟΝΤΕΛΟ ΠΕΛΑΤΗ – ΔΙΑΚΟΜΙΣΤΗ (CLIENT SERVER) ΠΗΓΗ : HTTPS://EN.WIKIPEDIA.ORG/WIKI/CLIENT%E2%80%93SERVER_MODEL	47
ΕΙΚΟΝΑ 31 - ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΤΕΧΝΙΚΗ MVC ΠΗΓΗ : HTTPS://EL.WIKIPEDIA.ORG/WIKI/MODEL-VIEW- CONTROLLER.....	48
ΕΙΚΟΝΑ 32 - Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	50
ΕΙΚΟΝΑ 33 - Η ΔΟΜΗ ΤΩΝ ΑΡΧΕΙΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	51
ΕΙΚΟΝΑ 34 - CONFIG.PHP	52
ΕΙΚΟΝΑ 35 - INDEX.PHP	52
ΕΙΚΟΝΑ 36 - Η ΑΡΧΙΚΗ ΣΕΛΙΔΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	53

EΙΚΟΝΑ 37 - ΦΟΡΜΑ ΕΓΓΡΑΦΗΣ ΧΡΗΣΤΗ	55
EΙΚΟΝΑ 38 - ΦΟΡΜΑ ΕΓΓΡΑΦΗΣ ΙΑΤΡΟΥ	56
EΙΚΟΝΑ 39 - ΚΛΗΣΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗ LOGIN ΤΟΥ ΜΟΝΤΕΛΟΥ USERMODEL.....	57
EΙΚΟΝΑ 40 - ΑΠΟΘΗΚΕΥΣΗ ΤΩΝ ΤΙΜΩΝ ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ ΣΤΟ SESSION	58
EΙΚΟΝΑ 41 - Η SHOWUSER ΤΗΣ HOME.....	58
EΙΚΟΝΑ 42 - ΟΙ ΤΙΜΕΣ ΤΟΥ SESSION ΓΡΑΦΟΝΤΑΙ ΣΤΑ ΑΝΤΙΣΤΟΙΧΑ INPUTS	59
EΙΚΟΝΑ 43 - ΦΟΡΜΑ ΕΝΗΜΕΡΩΣΗΣ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΧΡΗΣΤΗ	60
EΙΚΟΝΑ 44 - ΕΝΗΜΕΡΩΣΗ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΧΡΗΣΤΗ.....	61
EΙΚΟΝΑ 45 - ΦΟΡΜΑ ΕΠΙΛΟΓΗΣ ΙΑΤΡΩΝ	61
EΙΚΟΝΑ 46 - ΕΠΙΛΟΓΗ ΙΑΤΡΩΝ ΑΠΟ ΤΗ ΛΙΣΤΑ.....	62
EΙΚΟΝΑ 47 - ΠΡΟΒΟΛΗ ΔΙΑΘΕΣΙΜΩΝ ΓΙΑ ΕΠΙΛΟΓΗ ΙΑΤΡΩΝ ΕΙΔΙΚΟΤΗΤΑΣ ΝΕΥΡΟΛΟΓΙΑΣ.....	63
EΙΚΟΝΑ 48 - ΝΕΕΣ ΕΓΓΡΑΦΕΣ ΤΟΥ ΠΙΝΑΚΑ.....	64
EΙΚΟΝΑ 49 - Η ΙΔΙΑ ΣΕΛΙΔΑ ΜΕΤΑ ΤΗΝ ΕΠΙΛΟΓΗ ΤΟΥ ΚΟΥΜΠΙΟΥ ΕΝΗΜΕΡΩΣΗ ΤΗΣ ΠΡΩΤΗΣ ΕΓΓΡΑΦΗΣ ΤΟΥ ΠΙΝΑΚΑ	65
EΙΚΟΝΑ 50 - ΠΡΟΣΘΗΚΗ ΙΑΤΡΩΝ	66
EΙΚΟΝΑ 51 - ΔΙΑΓΡΑΦΗ ΙΑΤΡΩΝ ΚΑΙ ΠΡΟΣΘΗΚΗ ΝΕΩΝ.....	67
EΙΚΟΝΑ 52 - ΕΠΙΛΕΓΟΝΤΑΣ ΤΟ ΚΟΥΜΠΙ 'ΕΞΕΤΑΣΕΙΣ' Ο ΧΡΗΣΤΗΣ ΜΕΤΑΒΑΙΝΕΙ ΣΤΗΝ ΟΜΩΝΥΜΗ ΦΟΡΜΑ	68
EΙΚΟΝΑ 53 - ΕΓΓΡΑΦΕΣ ΤΟΥ ΠΙΝΑΚΑ EXAMS ΜΕΤΑ ΤΗΝ ΕΙΣΑΓΩΓΗ ΤΩΝ ΕΞΕΤΑΣΕΩΝ ΤΟΥ ΧΡΗΣΤΗ	69
EΙΚΟΝΑ 54 - ΤΜΗΜΑ ΚΩΔΙΚΑ JAVASCRIPT. ΣΤΗ ΠΕΡΙΠΤΩΣΗ ΠΟΥ Η ΤΙΜΗ ΤΟΥ RADIO ΕΙΝΑΙ PERSON ΤΑ ΑΝΤΙΚΕΙΜΕΝΑ TITLE ΚΑΙ FORMCONTENT ΤΗΣ ΦΟΡΜΑΣ ΠΑΙΡΝΟΥΝ ΤΙΣ ΑΝΤΙΣΤΟΙΧΕΣ ΤΙΜΕΣ.....	70
EΙΚΟΝΑ 55 - ΣΤΗ ΠΕΡΙΠΤΩΣΗ ΠΟΥ Η ΤΙΜΗ ΤΟΥ RADIO ΔΕΝ ΕΙΝΑΙ PERSON ΠΑΙΡΝΟΥΝ ΤΙΣ ΤΙΜΕΣ ΤΟΥ ΙΑΤΡΟΥ	70
EΙΚΟΝΑ 56 - ΤΜΗΜΑ ΚΩΔΙΚΑ ΣΥΝΑΡΤΗΣΗΣ INDEX ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ LOGIN. ΤΟ ΜΟΝΤΕΛΟ ΕΧΕΙ ΔΗΜΙΟΥΡΓΗΣΕΙ ΕΝΑ ΑΝΤΙΚΕΙΜΕΝΟ PERSON ΤΟΥ ΟΠΟΙΟΥ ΟΙ ΤΙΜΕΣ ΑΠΟΘΗΚΕΥΟΝΤΑΙ ΣΤΟ SESSION ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΧΡΗΣΗ.....	71
EΙΚΟΝΑ 57 - ΕΝΗΜΕΡΩΣΗ ΣΤΟΙΧΕΙΩΝ ΙΑΤΡΟΥ.....	72
EΙΚΟΝΑ 58 - Η ΣΥΝΑΡΤΗΣΗ EXAMS ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ DOCCONTROLER.....	73
EΙΚΟΝΑ 59 - Η ΣΥΝΑΡΤΗΣΗ GETDOCTORSPATIENTS ΠΟΥ ΠΑΙΡΝΕΙ ΩΣ ΟΡΙΣΜΑ ΤΟ ID ΤΟΥ ΙΑΤΡΟΥ ΚΑΙ ΜΕΣΩ ΕΝΟΣ JOIN ΤΩΝ ΠΙΝΑΚΩΝ USERS ΚΑΙ USERS2DOCTORS ΕΠΙΣΤΡΕΦΕΙ ΤΟΥΣ ΧΡΗΣΤΕΣ ΠΟΥ ΕΧΟΥΝ ΟΡΙΣΕΙ ΤΟ ΣΥΓΚΕΚΡΙΜΕΝΟ ΙΑΤΡΟ.....	73
EΙΚΟΝΑ 60 - ΠΡΟΒΟΛΗ ΣΤΟΙΧΕΙΩΝ ΑΣΘΕΝΟΥΣ.....	74
EΙΚΟΝΑ 61 - ΤΜΗΜΑ ΤΟΥ ΠΙΝΑΚΑ ΙΑΤΡΩΝ. Ο ΙΑΤΡΟΣ ΑΛΕΞΙΑ ΚΩΝΣΤΑΝΤΙΝΟΥ ΕΧΕΙ ΛΑΒΕΙ ΤΗΝ ID ΤΙΜΗ 5	74
EΙΚΟΝΑ 62 - ΤΜΗΜΑ ΤΟΥ ΠΙΝΑΚΑ USERS2DOCTORS. Ο ΙΑΤΡΟΣ ΜΕ ID 5 ΕΧΕΙ ΟΡΙΣΤΕΙ ΑΠΟ ΤΟΥΣ ΑΣΘΕΝΗΣ ΜΕ ID 13 ΚΑΙ 8	75
EΙΚΟΝΑ 63 - ΤΜΗΜΑ ΤΟΥ ΠΙΝΑΚΑ USERS. ΔΙΑΚΡΙΝΟΝΤΑΙ ΟΙ ΧΡΗΣΤΕΣ ΜΕ ID 13 ΚΑΙ 8.	75
EΙΚΟΝΑ 64 - ΠΡΟΒΟΛΗ ΑΣΘΕΝΩΝ ΙΑΤΡΟΥ.....	76
EΙΚΟΝΑ 65 - ΑΣΥΓΧΡΟΝΗ ΚΛΗΣΗ ΓΙΑ ΤΗ ΛΗΨΗ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΑΣΘΕΝΟΥΣ	77
EΙΚΟΝΑ 66 - Η ΣΥΝΑΡΤΗΣΗ GETPATIENTSDETAILS ΤΟΥ ΜΟΝΤΕΛΟΥ	77
EΙΚΟΝΑ 67 - ΤΜΗΜΑ ΚΩΔΙΚΑ ΤΗΣ SHOWPATIENT ΤΟΥ SCRIPT SHOWPATENTS	78
EΙΚΟΝΑ 68 - ΠΡΟΒΟΛΗ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΣΥΓΚΕΚΡΙΜΕΝΟΥ ΑΣΘΕΝΟΥΣ. ΟΙ ΕΞΕΤΑΣΕΙΣ ΤΟΥ ΑΝΤΙΣΤΟΙΧΟΥΝ ΣΤΙΣ ΕΓΓΡΑΦΕΣ ΠΟΥ ΠΕΡΙΕΧΟΥΝ ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΡΧΕΙΟ	79
EΙΚΟΝΑ 69 - ΠΡΟΒΟΛΗ ΣΤΟΙΧΕΙΩΝ ΕΤΕΡΟΥ ΑΣΘΕΝΟΥΣ	80
EΙΚΟΝΑ 70 - ΤΜΗΜΑ ΤΟΥ ΠΙΝΑΚΑ EXAMS. ΔΙΑΚΡΙΝΟΝΤΑΙ ΟΙ ΠΕΡΙΓΡΑΦΕΣ ΤΩΝ ΕΞΕΤΑΣΕΩΝ ΤΟΥ ΧΡΗΣΤΗ 22 Ο ΟΠΟΙΟΣ ΑΝΤΙΣΤΟΙΧΕΙ ΣΤΗ ΖΩΗ ΓΕΩΡΓΙΟΥ ΤΗΣ ΠΡΟΗΓΟΥΜΕΝΗΣ ΕΙΚΟΝΑΣ.....	81
EΙΚΟΝΑ 71 - ΤΜΗΜΑ ΚΩΔΙΚΑ ΤΗΣ SHOWPATIENT ΤΟΥ SCRIPT SHOWPATENTS ΤΟ ΟΠΟΙΟ ΑΝΑΓΡΑΦΕΙ ΣΤΟ ΠΙΝΑΚΑ ΤΗ ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΚΑΘΕ ΕΞΕΤΑΣΗΣ ΚΑΙ ΜΙΑ ΦΟΡΜΑ ΓΙΑ ΤΗΝ ΑΝΑΚΤΗΣΗ ΤΟΥ ΑΡΧΕΙΟΥ.....	81
EΙΚΟΝΑ 72 - Η ΣΥΝΑΡΤΗΣΗ GETPDF ΤΟΥ DOCCONTROLER.....	82

1. Ιατρικός Φάκελος

1.1 Γέννηση Ιατρικού Φακέλου

Τα τεχνολογικά άλματα στο τομέα των υπολογιστικών συστημάτων δεν θα μπορούσαν να αφήσουν ανεπηρέαστο και τον τομέα της υγείας, ο οποίος κατά τη διάρκεια συγγραφής της πτυχιακής, λάμβανε διαφορετικές διαστάσεις, δεδομένης της επιδημίας του κορονοϊού που μαστιίζει τη παγκόσμια κοινότητα. Για πολλά χρόνια, ο ιατρικός φάκελος ασθενούς αποτελούνταν από χειρόγραφα κείμενα, δεδομένης της εξοικείωσης των ανθρώπων με το χαρτί εξαιτίας της απλότητας της μεθόδου. Η μέθοδος όμως αυτή, παρουσίαζε σημαντικά προβλήματα αναφορικά με την εξοικονόμηση χρόνου και την διαθεσιμότητα της πληροφορίας ανάμεσα στους διάφορους ιατρούς του κάθε ασθενούς.

Οι πληροφορίες που απαιτούνται για την φροντίδα ενός αρρώστου έχουν αυξηθεί εκθετικά τα τελευταία χρόνια λόγω του ότι έχει αυξηθεί και ο αριθμός των εξετάσεων που πρέπει να υποβληθεί ένας ασθενής προκειμένου να αξιολογηθεί η κατάστασή του. Οι εξετάσεις αυτές παράγουν ένα σύνολο δεδομένων, τα οποία θα πρέπει να επεξεργαστούν και να αποθηκευτούν ώστε να είναι διαθέσιμα στο σύνολο των ιατρών που θα εξετάσουν τον ασθενή, ανεξαρτήτως του χρονικού διαστήματος που έχει παρέλθει από την ημερομηνία διεξαγωγής των εξετάσεων. Φυσικά δεν θα πρέπει να αγνοηθεί και ο επιπλέον όγκος εργασίας που επιφέρουν τα δεδομένα αυτά στους ιατρούς, οι οποίοι θα πρέπει να συνεργαστούν ορθά για τη διαδικασία εκτίμησης των εξετάσεων, σύμφωνα με την εργασία του Χουλιάρη, 2006. [01]

Καθίσταται πλέον αντιληπτό στον αναγνώστη, ότι η μέθοδος της διατήρησης του ιατρικού ιστορικού του ασθενή σε χαρτί, δεν είναι πλέον αρκετή για να διαχειριστεί το σύνολο των δεδομένων που απαιτούνται. Επιπλέον, η τεχνολογία έχει εξελιχθεί και διαθέτει τα απαραίτητα εργαλεία για την διαχείριση των δεδομένων αυτών. Παράλληλα, τα εργαλεία αυτά παρέχονται σε ένα λογικό κόστος, αφού πλέον τα υλικά που απαιτούνται για την λειτουργία τους παράγονται και διατίθενται ευκολότερα. Επίσης, έχει βελτιωθεί και το μορφωτικό επίπεδο της πλειονότητας των ανθρώπων, με αποτέλεσμα να είναι πλέον σε θέση να χρησιμοποιήσουν τη τεχνολογία αυτή, χωρίς να απαιτούνται σύνθετες εκπαιδευτικές διαδικασίες.

Τις νέες αυτές προκλήσεις, που προκύπτουν από την ανάγκη για προσφορά ποιοτικότερων υπηρεσιών υγείας, έρχεται να αντιμετωπίσει η τεχνολογία, η οποία με τα μέσα που πλέον διαθέτει μπορεί να προσφέρει τον Ηλεκτρονικό Φάκελο Ασθενούς. Αυτός θα εξομοιώσει την παραδοσιακή τεχνική καταγραφής των δεδομένων σε χαρτί, αξιοποιώντας ταυτόχρονα όλα τα οφέλη που έχει να προσφέρει η τεχνολογία των υπολογιστικών συστημάτων. Πιο συγκεκριμένα, στον Ηλεκτρονικό Φάκελο Υγείας καταγράφονται όλες οι ιατρικές εξετάσεις στις οποίες θα υποβληθεί ο ασθενής σε όλο το διάστημα της ζωής του, καθώς και το σύνολο των ιατρικών δεδομένων του ασθενή.

Οι πληροφορίες αυτές είναι διαθέσιμες σε οποιοδήποτε γιατρό κριθεί απαραίτητο να είναι προσβάσιμες, ανεξαρτήτου γεωγραφικής τοποθεσίας, άμεσα και οποιαδήποτε χρονική στιγμή απαιτηθεί. Κατ' αυτόν τον τρόπο, ο ασθενής μπορεί να επισκεφθεί οποιοδήποτε ιατρό, σε οποιοδήποτε μέρος του πλανήτη, και ο ιατρός αυτός να έχει πρόσβαση στο σύνολο της διαθέσιμης πληροφορίας άμεσα και οποιαδήποτε χρονική στιγμή.

1.2 Ορισμοί

Σύμφωνα με την Ευρωπαϊκή Επιτροπή Προτυποποίησης, ο όρος Ιατρικός Φάκελος αναφέρεται στη διαδικασία αποθήκευσης του συνόλου των πληροφοριών που αφορούν το ιατρικό ιστορικό του ασθενούς έτσι ώστε να αποτελεί το θεμέλιο τόσο της διάγνωσης όσο και της θεραπευτικής αντιμετώπισης του ασθενούς, ενώ παράλληλα αποτελεί και τη βάση για επιδημιολογικές έρευνες. Πέραν αυτών, παρέχει πληροφορίες διοικητικής, οικονομικής, στατιστικής φύσης καθώς και ποιοτικού ελέγχου. [02]

Διαφορετικό ορισμό προσδίδει ο Hunter, 2003 [03] στον όρο, όπου δηλώνει ότι ο Ηλεκτρονικός Φάκελος Υγείας περιλαμβάνει το σύνολο των πληροφοριών που σχετίζονται με την φυσική ή ψυχική υγεία ή κατάσταση ενός ασθενούς, στο παρελθόν, το παρόν και το μέλλον, οι οποίες καταγράφονται σε ένα ηλεκτρονικό σύστημα με τρόπο τέτοιο ώστε να είναι δυνατή η επεξεργασία τους από τους Ηλεκτρονικούς Υπολογιστές και να είναι διαθέσιμες μέσω του Διαδικτύου, με στόχο την υγειονομική περίθαλψη και φροντίδα του ασθενούς.

1.3 Χαρακτηριστικά και Προϋποθέσεις Ηλεκτρονικού Ιατρικού Φακέλου

Ο ηλεκτρονικός ιατρικός φάκελος ασθενούς ουσιαστικά αποτελεί την συστηματική συλλογή του ιστορικού αλλά και την καταγραφή κατάστασης της υγείας του εκάστοτε ασθενούς. Δημιουργείται από κάποιον επαγγελματία φροντίδας υγείας και ενημερώνεται για την κατάσταση της υγείας του πολίτη για όλη του τη ζωή. Οι πληροφορίες που αφορούν την κατάσταση της υγείας του δύναται να περιλαμβάνουν δημογραφικά στοιχεία, ενημερωτικά σημειώματα, διαγνώσεις, φαρμακευτικές αγωγές, ιατρικό ιστορικό, εμβολιασμούς, αποτελέσματα εξετάσεων, και άλλα.

Η συγκέντρωση της πληροφορίας σε ένα κεντρικό σημείο έχει σαν αποτέλεσμα να διευκολύνεται σε μεγάλο βαθμό η αποστολή του κάθε ιατρού, αφού πλέον έχει πρόσβαση στο σύνολο της πληροφορίας του κάθε ασθενούς άμεσα, έγκυρα και οποιαδήποτε στιγμή αυτό απαιτηθεί.

Για να μπορεί όμως το σύστημα αυτό να εκτελέσει με επιτυχία την αποστολή του, θα πρέπει να πληροί τις ακόλουθες προϋποθέσεις:

- Η πληροφορία που θα προσφέρει να είναι έγκυρη και ενημερωμένη.
- Η επικοινωνία να γίνεται με τρόπο τέτοιο ώστε να μην υπάρχει κίνδυνος διαρροής προσωπικών πληροφοριών και δεδομένων.
- Να υπάρχει έλεγχος πρόσβασης στη πληροφορία.
- Το περιβάλλον να είναι φιλικό προς το χρήστη και να μην απαιτούνται εξειδικευμένες γνώσεις ως προς την χρήση του.
- Υψηλή διαθεσιμότητα.
- Επεκτασιμότητα.

1.4 Πλεονεκτήματα και Μειονεκτήματα

Το σύστημα προσφέρει σημαντικά οφέλη προς τον χρήστη, όπως:

- Απλοποίηση των διαδικασιών διαχείρισης των ασθενών που οδηγεί σε ορθότερη εξαγωγή συμπερασμάτων.

- Έγκυρη πληροφόρηση μέσω των συστημάτων κωδικοποίησης (ICD, GEHR, ICPC, κλπ.).
- Υποβοήθηση στη διάγνωση μέσω συστημάτων knowledge – based.
- Εμπλουτισμένα δεδομένα ασθενούς που περιλαμβάνουν εικόνες, βίντεο, κλπ.
- Αυτόματη και έγκυρη εισαγωγή δεδομένων από εξετάσεις χάρις της επικοινωνίας των υποσυστημάτων.

Φυσικά, όπως συμβαίνει με κάθε νέα καινοτομία, αρκετές είναι και οι προκλήσεις που εισάγει αλλά και οι κίνδυνοι που ελλοχεύουν από την εφαρμογή της:

- Η έλλειψη σαφούς καθορισμένου πλαισίου κανόνων, περιορίζει τη λειτουργικότητα του συστήματος, αφού ο κάθε επαγγελματίας υγείας λειτουργεί και καταχωρεί στο σύστημα σύμφωνα με την εμπειρία του και όχι σύμφωνα με ένα ορισμένο σύνολο κανόνων.
- Το νέο σύστημα προϋποθέτει την εξοικείωση των χρηστών με τη χρήση του, πράγμα που προβληματίζει ορισμένες κατηγορίες χρηστών, όπως γιατρούς μεγαλύτερης ηλικίας με λιγότερη εξοικείωση στα υπολογιστικά συστήματα.
- Το σύστημα θα πρέπει να προσφέρει υψηλή διαθεσιμότητα και εύκολη ανάκαμψη καθώς σε περίπτωση δυσλειτουργίας, οι ασθενείς δεν θα μπορούν να περιθλαφθούν.
- Θα πρέπει να εξασφαλίζεται η ασφάλεια των προσωπικών δεδομένων των ασθενών και το ιατρικό απόρρητο.

1.5 Νομικό Πλαίσιο

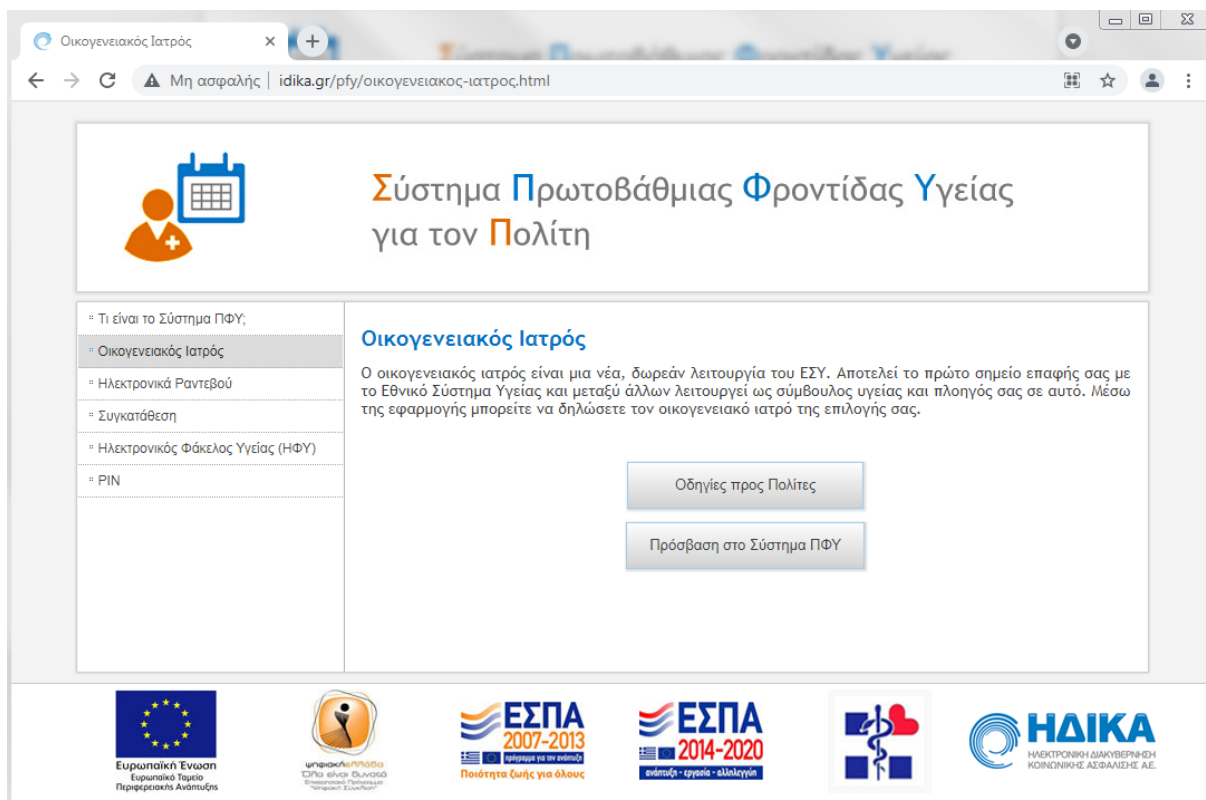
Στην Ελλάδα, ο ηλεκτρονικός φάκελος υγείας ορίζεται από το Ν. 4486/2017 και συγκεκριμένα από το άρθρο 21. Ο νόμος ορίζει ότι ο Ατομικός Φάκελος Υγείας θα περιέχει το συνοπτικό ατομικό ιστορικό υγείας και θα καθορίζει ένα ενιαίο πρότυπο σχετικά με το περιεχόμενο, τον τρόπο κατάρτισης, την ταυτοποίηση του ατόμου και την πρόσβαση σε ιατρικές πληροφορίες του φακέλου. Ορίζει επίσης, ότι ο ιατρικός φάκελος ασθενούς θα πρέπει να δημιουργείται από οικογενειακό ιατρό ή από τους επαγγελματίες υγείας της ΤΟΜΥ. Οι πιστοποιημένοι χρήστες του συστήματος, θα καταχωρούν στον φάκελο όλες τις πληροφορίες που προκύπτουν από την εξέταση - επίσκεψη του ασθενούς και είναι απαραίτητες για την παρακολούθηση - θεραπεία του ατόμου.

Για την διασφάλιση των προσωπικών δεδομένων, ο νόμος ορίζει ότι τα δεδομένα του φακέλου αποτελούν ιδιοκτησία του ατόμου και τηρούνται ασφαλή, με ευθύνη του Υπουργείου Υγείας σύμφωνα με την νομοθεσία περί προστασίας των προσωπικών δεδομένων. Παράλληλα καθορίζεται ποιος θα έχει πρόσβαση στα δεδομένα των ασθενών και ποιος θα είναι υπεύθυνος για την μεταβίβαση, επεξεργασία, αποθήκευση και φύλαξη των πληροφοριών. [04]

1.6 Εφαρμογές

1.6.1 Ηλεκτρονικός Φάκελος Υγείας – ΗΔΙΚΑ

Στην Ελλάδα, ο Ατομικός Ηλεκτρονικός Φάκελος είναι υποσύστημα του Συστήματος Πρωτοβάθμιας Φροντίδας Υγείας για τον Πολίτη και για να μπορέσει να τη χρησιμοποιήσει ο πολίτης θα πρέπει αρχικά να δηλώσει Οικογενειακό Ιατρό. Η διαδικασία αυτή ολοκληρώνεται μέσω της ιστοσελίδας : <http://www.idika.gr/pfy/οικογενειακος-ιατρος.html> .



Εικόνα 1 - Δήλωση Οικογενειακού Γιατρού

Επιλέγοντας ‘Πρόσβαση στο Σύστημα ΠΦΥ’ ο χρήστης ανακατευθύνεται στην ιστοσελίδα e-syntagografisi.gr/p-iv/p όπου εκεί καλείται να εισάγει τους κωδικούς του.

Εδώ η είσοδος μπορεί να γίνει με τρεις διαφορετικούς τρόπους:

- Είτε με τους κωδικούς Taxisnet
- είτε με τους κωδικούς που έχουν δοθεί από τον οικογενειακό ιατρό
- είτε μέσω eIDAS που αφορά ευρωπαίους πολίτες

gov.gr BETA Σύστημα Πρωτοβάθμιας Φροντίδας Υγείας ΗΔΙΚΑ

Συχνές ερωτήσεις Ειχεράδια οδηγιών Αποστολή σχολίων - παρατηρήσεων

Είσοδος με TaxisNet Είσοδος με ΑΗΦΥ Είσοδος με eIDAS

Η είσοδος γίνεται με τη χρήση των κωδικών του TaxisNet και αποκτάτε πρόσβαση στις υπηρεσίες: Άυλη Συνταγογράφηση, Ηλεκτρονικά Ραντεβού, Οικογενειακός Ιατρός. Απαιτείται η γνώση του ΑΜΚΑ και στην περίπτωση που ο ασφαλισμένος δεν τον θυμάται, μπορεί να τον αναζητήσει, πατώντας τον ακόλουθο σύνδεσμο: [Εύρεση ΑΜΚΑ](#)

Είσοδος στην υπηρεσία

Άυλη Συνταγογράφηση Αναβάθμιση Ασφάλειας Συστήματος

Σας ενημερώνουμε ότι πλέον μπορείτε, από τη Δευτέρα 23/03/2020, εάν επιθυμείτε, να λαμβάνετε με ηλεκτρονικό τρόπο τις συνταγές φαρμάκων που εκδίδονται από τους ιατρούς για εσάς (άυλη συνταγογράφηση).

Πριν ξεκινήσετε

Για την ενεργοποίηση της Άυλης Συνταγογράφησης, θα χρειαστείτε:

- Τους κωδικούς πρόσβασης στο TaxisNet, για να επιβεβαιώσετε τα στοιχεία σας
- Τον αριθμό ΑΜΚΑ σας ή/και των ασφαλιστικά εξαρτώμενων μελών σας κάτω των 18 ετών
- Το κινητό σας τηλέφωνο ή/ και πρόσβαση στο email σας, όπου θα σας αποσταλούν κωδικοί επιβεβαίωσης πριν την ενεργοποίηση

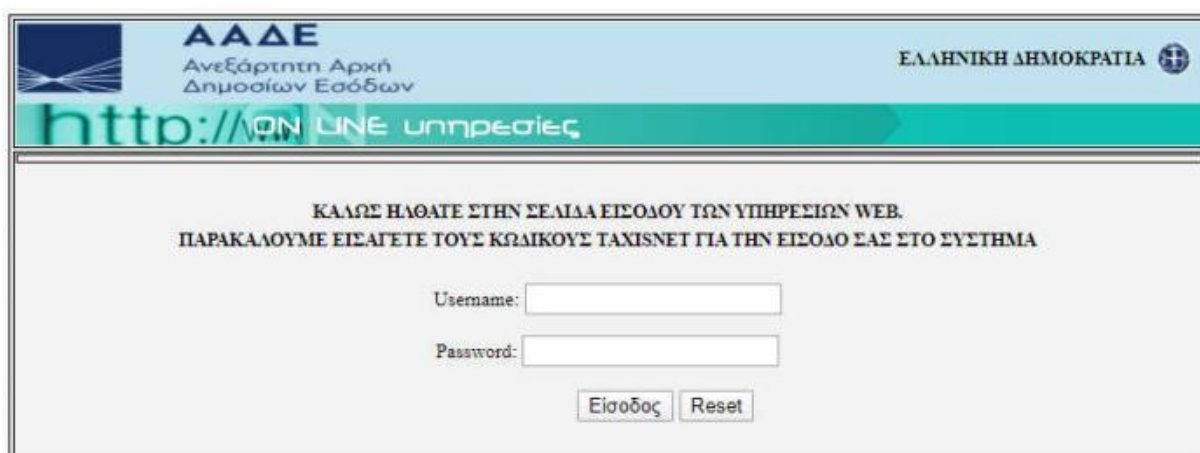
Περισσότερες πληροφορίες αλλά και οδηγίες σχετικά με την ενεργοποίηση της Άυλης Συνταγογράφησης:
[Οδηγίες για Άυλη Συνταγογράφηση \(με κωδικούς TaxisNet\)](#)
[Οδηγίες για Άυλη Συνταγογράφηση \(με κωδικούς ΑΗΦΥ\)](#)

Υπουργείο Υγείας Ministry of Health Ευρωπαϊκή Ένωση European Union ΕΣΠΑ 2014-2020 Επιχειρησιακό Πρόγραμμα ΜΕΤΑΡΡΥΘΜΙΣΗ ΔΗΜΟΣΙΟΥ ΤΟΜΕΑ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Εικόνα 2 - Εισαγωγή στο σύστημα με κωδικούς taxisnet. Πηγή: erdv_manual_public_v3.pdf

Επιλέγοντας 'Είσοδος στην υπηρεσία', το σύστημα ανακατευθύνει το χρήστη στη σελίδα της Γενικής Γραμματείας Πληροφοριακών Συστημάτων μέσω της οποίας θα γίνει η πιστοποίηση των στοιχείων.



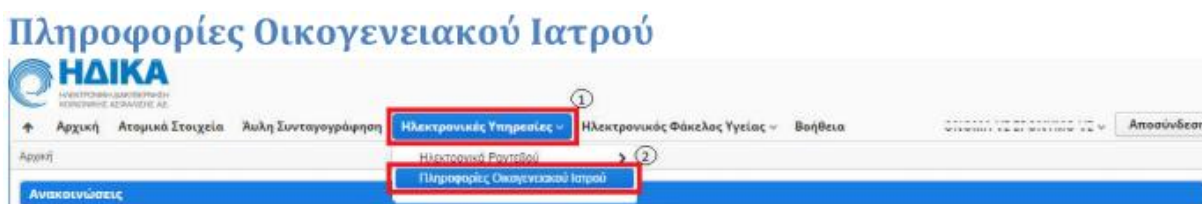
Εικόνα 3 - Πιστοποίηση Στοιχείων Χρήστη. Πηγή:erdv_manual_public_v3.pdf

Στη συνέχεια, ο χρήστης θα πρέπει να εξουσιοδοτήσει το σύστημα ΗΔΙΚΑ (Ηλεκτρονική Διακυβέρνηση Κοινωνικής Ασφάλισης) ώστε να έχει πρόσβαση στα προσωπικά του στοιχεία. Η διαδικασία ολοκληρώνεται με την εισαγωγή του ΑΜΚΑ (Αριθμός Μητρώου Κοινωνικής Ασφάλισης) του για να γίνει η επιβεβαίωση των στοιχείων του.



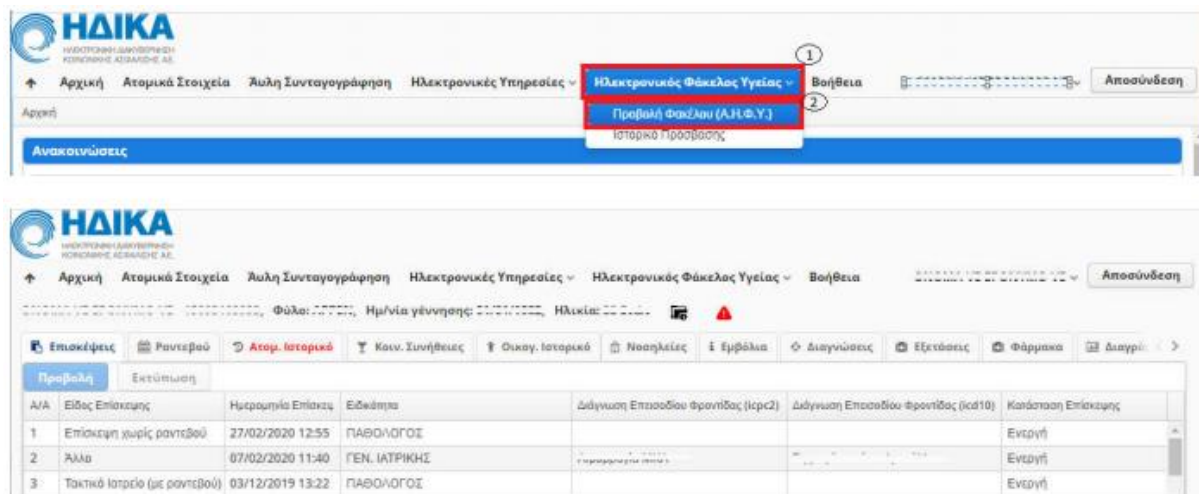
Εικόνα 4 - Επιβεβαίωση Στοιχείων. Πηγή: erdv_manual_public_v3.pdf

Όταν ολοκληρωθεί η διαδικασία διαπίστευσης του χρήστη, εμφανίζεται στο μενού η επιλογή 'Ηλεκτρονικές Υπηρεσίες' > 'Πληροφορίες Οικογενειακού Ιατρού' από όπου ο χρήστης μπορεί να υποβάλει αίτηση δήλωσης νέου οικογενειακού ιατρού επιλέγοντας έναν από τους διαθέσιμους ιατρούς. Έπειτα, το σύστημα θα διαθέσει την αίτηση για εκτύπωση, την οποία θα πρέπει να προσκομίσει στον οικογενειακό ιατρό που δήλωσε. Ο τελευταίος, θα ενεργοποιηθεί σαν χρήστης και θα έχει πλέον στη διάθεσή του, τους κωδικούς Ατομικού Ηλεκτρονικού Φακέλου με την άδεια του ασθενή.



Εικόνα 5 - Δήλωση Οικογενειακού Ιατρού. Πηγή: rdn_manual_public_v3.pdf

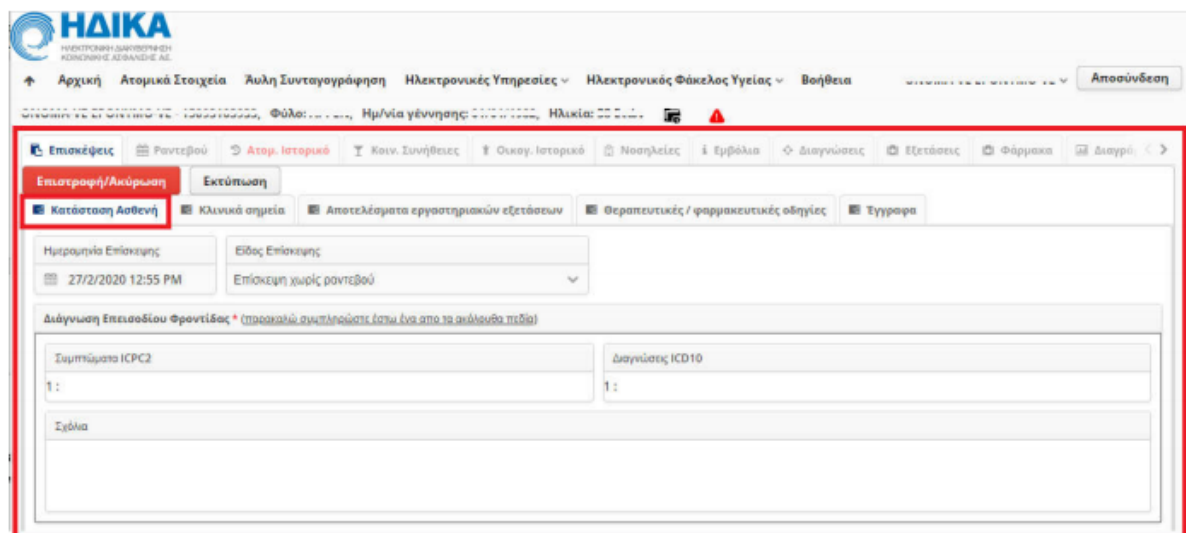
Έχοντας πλέον κωδικούς ΑΗΦΥ (Ατομικός Ηλεκτρονικός Φάκελος Υγείας), ο ασθενής μπορεί να συνδεθεί στο σύστημα και να έχει πλέον πρόσβαση στον Ιατρικό του Φάκελο.



Εικόνα 6 - Εφαρμογή Ατομικού Ηλεκτρονικού Φακέλου Υγείας. Πηγή: erdn_manual_public_v3.pdf

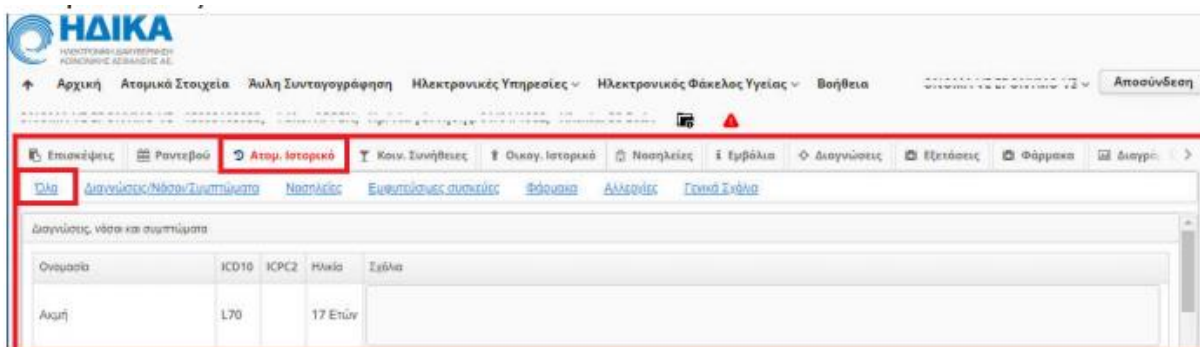
Όπως εύκολα παρατηρείται, η εφαρμογή παραθέτει ένα πλήθος καρτελών μέσω των οποίων γίνεται η διαχείριση των δεδομένων. Μέσω της καρτέλας 'Επισκέψεις', εμφανίζονται όλες οι καταχωρημένες επισκέψεις. Για κάθε επίσκεψη καταγράφεται το είδος, η ημερομηνία που πραγματοποιήθηκε, η ιδιότητα του ιατρού που την καταχώρησε καθώς και η διάγνωση του. Από την επιλογή 'Προβολή',

εμφανίζονται και οι επιμέρους ενότητες που περιλαμβάνουν τη 'Κατάσταση Ασθενή', τα 'Κλινικά σημεία', τα 'Αποτελέσματα εργαστηριακών εξετάσεων', οι 'Θεραπευτικές/φαρμακευτικές οδηγίες' και τα 'Έγγραφα'.



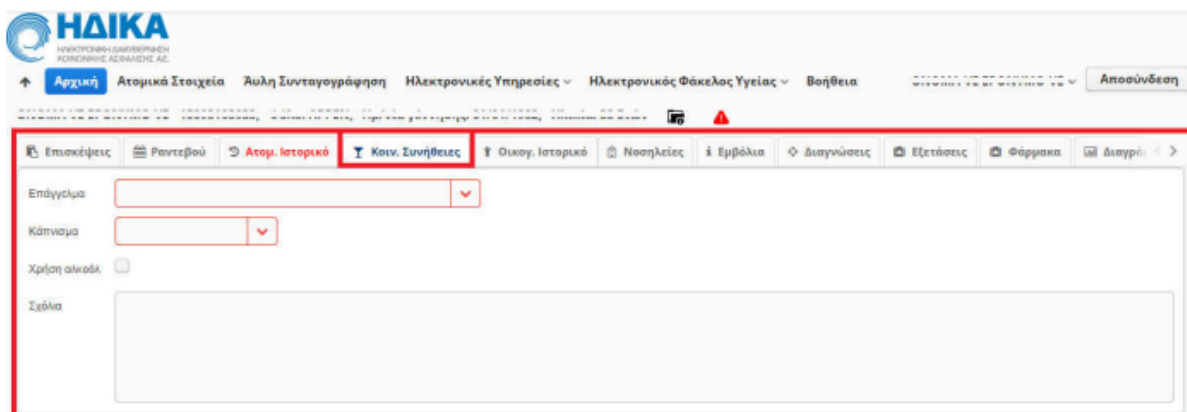
Εικόνα 7 - Η καρτέλα επισκέψεις/κατάσταση ασθενή. Πηγή: erdv_manual_public_v3.pdf

Το ατομικό ιστορικό βρίσκεται στην ομώνυμη καρτέλα και περιλαμβάνει τις ενότητες 'Διαγνώσεις/Νόσοι/Συμπτώματα', 'Νοσηλείες', 'Εμφυτεύσιμες συσκευές', 'Φάρμακα', 'Αλλεργίες' και 'Γενικά σχόλια'.



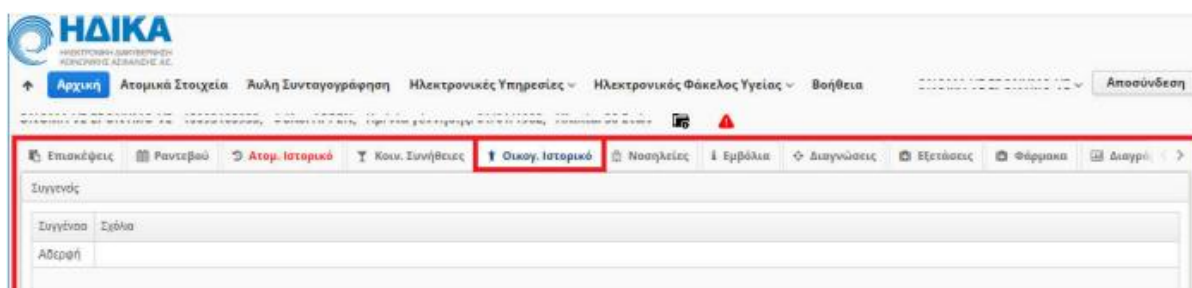
Εικόνα 8 - Ατομικό Ιστορικό. Πηγή: erdv_manual_public_v3.pdf

Οι κοινωνικές συνήθειες του ατόμου περιλαμβάνουν την επαγγελματική του κατάσταση, εάν είναι καπνιστής ή όχι, λεπτομέρειες για τη χρήση αλκοόλ και λοιπά σχόλια.



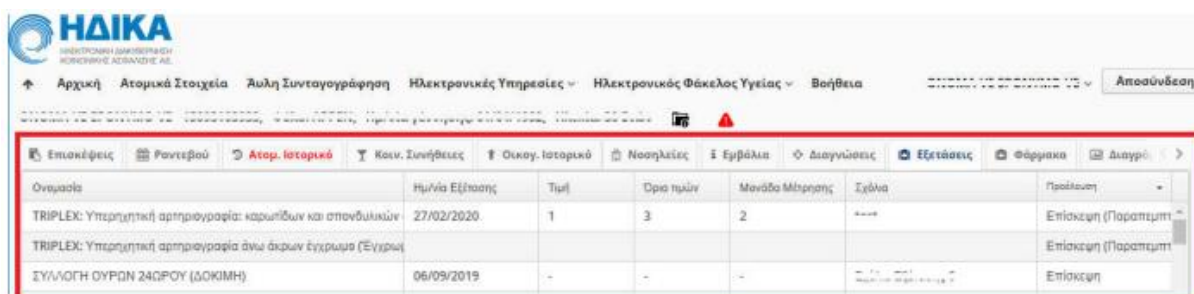
Εικόνα 9 - Καρτέλα με τις Κοινωνικές Συνήθειες του ατόμου. Πηγή:erdv_manual_public_v3.pdf

Στην καρτέλα ‘Οικογενειακό Ιστορικό’ εμφανίζονται όλες οι απαραίτητες πληροφορίες υγείας των μελών της οικογένειας του ατόμου για τα οποία έχουν καταχωρηθεί δεδομένα.



Εικόνα 10 - Οικογενειακό Ιστορικό ατόμου. Πηγή:erdv_manual_public_v3.pdf

Η επόμενη καρτέλα που θα παρουσιαστεί είναι αυτή των εξετάσεων που συγκεντρώνει το σύνολο των δεδομένων από όλες τις εξετάσεις που έχουν καταγραφεί στο σύστημα.



Εικόνα 11 - Δεδομένα Εξετάσεων. Πηγή: erdv_manual_public_v3.pdf

Τα φάρμακα που έχει λάβει το άτομο περιλαμβάνονται στην ομώνυμη καρτέλα, όπου για κάθε φάρμακο καταγράφεται η εμπορική του ονομασία, η δραστική ουσία που περιέχει, η ποσότητα πρόσληψης, η μορφή της δόσης, η διάρκεια της θεραπείας, η ημερομηνία που δόθηκε το σκεύασμα και τέλος, τη προέλευση του φαρμάκου.

Εμπορική Ονομασία	Δραστική Ουσία	Μονάδα	Μορφή Δόσης	Δόση (g)	Ημερομηνία	Προέλευση
AUGMENTIN F.C.TAB (500+125)MG/TAB BTx16	AMOXICILLIN TRIHYDRATE,CLAVULANATE POTAS	1	ΔΙΣΚΑ ΕΠΙΚΑΛ	5	29/07/2016	Συνταγογράφηση
TETAGAM-P INJ.SOL 250IU/1ML AMP BTx1PF5x1	IMMUNOGLOBULIN HUMAN ANTI-TETANUS	1	ΕΝΕΣΗ ΔΙΑΛ.ΦΥΣΙΓΓΕΣ	1	29/07/2016	Συνταγογράφηση
AURID NASPR.SUS 100MCG/DOSE FLX10MLJ20x	BUDESONIDE	1	ΡΙΝΙΚΟ ΣΠΡΑΥ ΔΟΣΕΙΣ	15	12/07/2016	Συνταγογράφηση
BILARGEN TAB 20MG/TAB BTx30 TABS	BILASTINE	1	ΔΙΣΚΑ	15	12/07/2016	Συνταγογράφηση

Εικόνα 12 - Τα φάρμακα που έχει λάβει το άτομο. Πηγή:erdv_manual_public_v3.pdf

Τέλος, το ‘Συνοπτικό Ιστορικό Υγείας’ περιλαμβάνει τα στοιχεία του προσώπου, τις καταχωρημένες διαγνώσεις και τις φαρμακευτικές αγωγές των τελευταίων έξι μηνών καθώς και άλλες ενότητες. [05]



Εικόνα 13 - Συνοπτικό Ιστορικό. Πηγή : erdv_manual_public_v3.pdf

1.6.2 Microsoft HealthVault

Ο κολοσσός της πληροφορικής δεν θα μπορούσε να απουσιάζει από την προσπάθεια δημιουργίας ενός συστήματος παροχής υπηρεσιών υγείας. Το σύστημα, που παρουσιάστηκε τον Οκτώβριο του 2007 από την Microsoft, αποτελούνταν από ένα σύστημα ιστοσελίδων το οποίο αποθήκευε και διαχειριζόταν δεδομένα υγείας. Απευθυνόταν τόσο σε ιδιώτες, όσο και σε επαγγελματίες του χώρου, και το 2010 επεκτάθηκε εκτός των Ηνωμένων Πολιτειών όπου περιλάμβανε και τους πολίτες του Ηνωμένου Βασιλείου.

Η πρόσβαση στα δεδομένα γινόταν μέσω λογαριασμού που είχε δημιουργήσει ο χρήστης, ο οποίος είχε τη δυνατότητα να διαχειρίζεται και άλλα άτομα, με σκοπό ο ίδιος ο γονιός, παραδειγματικά, να μπορεί να διαχειρίζεται τα δεδομένα των παιδιών του. Πέραν του ιστότοπου, η διαχείριση των δεδομένων μπορούσε να γίνει και μέσω εφαρμογής ή ακόμα και συσκευής η οποία μπορούσε να επικοινωνήσει με το σύστημα. Βασική προϋπόθεση ήταν ο χρήστης να εξουσιοδοτήσει την εφαρμογή ώστε να μπορέσει εκείνη να χρησιμοποιήσει τα προσωπικά του δεδομένα. Επιπλέον, το σύστημα παρείχε στο άτομο τη δυνατότητα να μοιράζεται τα δεδομένα του με άλλα άτομα, όπως ιατροί, συγγενής, κτλ.

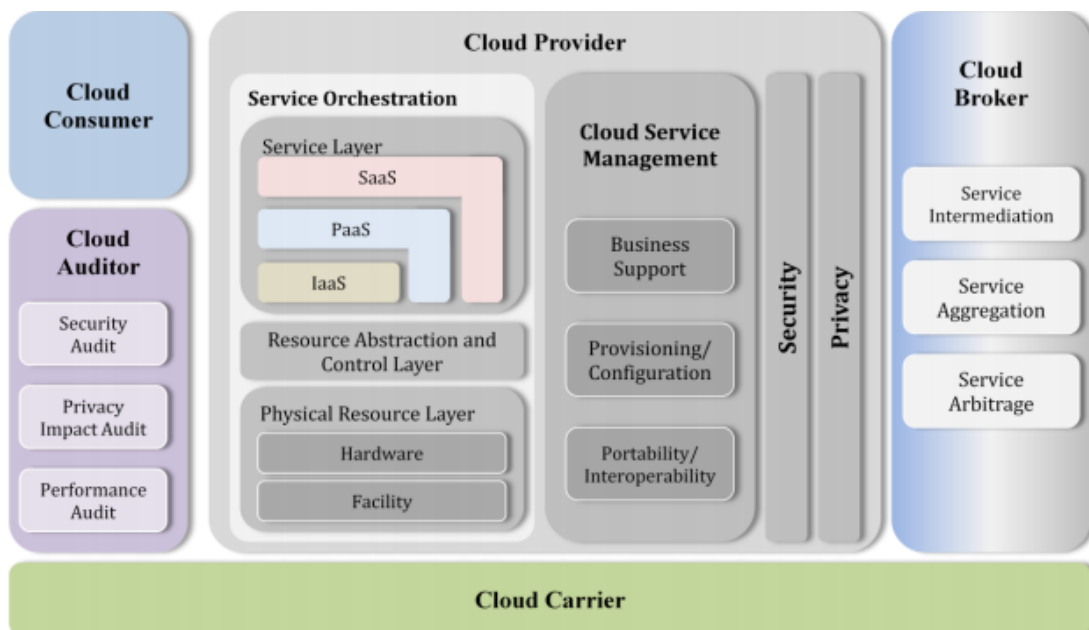
Το σύστημα μπορούσε να επικοινωνήσει και με πληθώρα άλλων συσκευών όπως συσκευές μέτρησης αρτηριακής πίεσης, μετρητές καρδιακών παλμών, ζυγαριές με WiFi, κλπ. Τις συσκευές αυτές τις εγκαθιστούσε το άτομο στο λογαριασμό του και μέσω αυτού, είχε πρόσβαση στα δεδομένα που αυτές παρήγαγαν. Επιπλέον, η εταιρεία προσέφερε και οδηγούς (drivers) για ορισμένες συσκευές. Το 2014 η εταιρεία κυκλοφόρησε ένα έξυπνο ρολόι χειρός που διέθετε και λειτουργίες καταγραφής δραστηριοτήτων. Η συσκευή κατέγραφε δεδομένα από το άτομο που την χρησιμοποιούσε και τα αποθήκευε στο HealthVault από όπου ο χρήστης μπορούσε να τα αξιολογεί. Η λειτουργία της εφαρμογής έπαψε τον Νοέμβριο του 2019, μετά από απόφαση της ίδιας της εταιρείας. [06]

2. Υπολογιστική Νέφος

2.1 Εισαγωγή

Ο όρος αναφέρεται σε οποιοδήποτε λογισμικό εκτελεί ο χρήστης το οποίο όμως δεν βρίσκεται εγκατεστημένο στον δικό του προσωπικό υπολογιστή αλλά σε κάποιο διαφορετικό, με τον οποίο επικοινωνεί μέσω Διαδικτύου. Με τον τρόπο αυτό, ο χρήστης έχει πρόσβαση στα δεδομένα του από οποιαδήποτε τοποθεσία και οποιαδήποτε χρονική στιγμή. Φυσικά, η υπηρεσία δεν απευθύνεται μόνο σε μεμονωμένους χρήστες αλλά και σε εταιρίες οι οποίες μπορούν πλέον να μεταφέρουν το σύνολο των εφαρμογών τους στο νέφος, απολαμβάνοντας τα οφέλη που έχει να τους προσφέρει, καταβάλλοντας φυσικά το ανάλογο κόστος.

Κατά καιρούς, διάφοροι ορισμοί έχουν προταθεί για την περιγραφή του όρου Υπολογιστική Νέφος, μεταξύ αυτών και η πρόταση του Εθνικού Ινστιτούτου Προτύπων και Τεχνολογίας των ΗΠΑ, κατά τον οποίο ορίζεται ως ένα μοντέλο που παρέχει την ικανότητα συνεχούς, εύκολης αλλά και υψηλών απαιτήσεων πρόσβασης σε μια κοινόχρηστη συλλογή από υπολογιστικούς πόρους, οι οποίοι τροφοδοτούνται και απελευθερώνονται με σχετικά μικρή προσπάθεια διαχείρισης και αλληλεπίδρασης. [07-08]



Εικόνα 14 - Αρχιτεκτονική Υπολογιστικής Νέφους σύμφωνα με το NIST (National Institute of Standards and Technology) Πηγή: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication500-292.pdf>

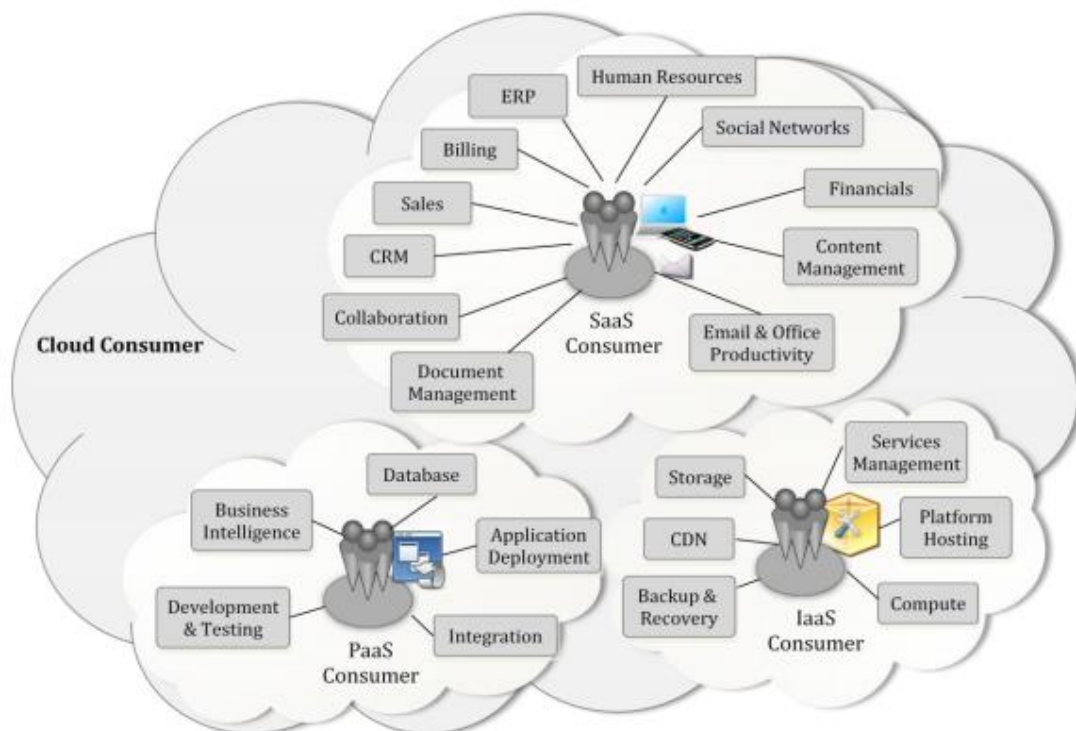
Σύμφωνα με την άνω αρχιτεκτονική, καθορίζονται οι βασικοί ρόλοι της υπηρεσίας. Συγκεκριμένα ο καταναλωτής (Cloud Consumer), ο πάροχος (Cloud Provider), ο διαμεσολαβητής (Cloud Broker) , ο σύμβουλος (Cloud Auditor) και ο φορέας (Cloud Carrier) της υπηρεσίας.

2.1.1 Ο Καταναλωτής (Cloud Consumer)

Ο καταναλωτής είναι το άτομο ή η επιχείρηση η οποία διατηρεί μια επαγγελματική σχέση με τον πάροχο και χρησιμοποιεί τις υπηρεσίες που αυτός παρέχει. Για τις διαθέσιμες αυτές υπηρεσίες, ενημερώνεται από τον πάροχο και κατόπιν δύναται να τις αξιοποιήσει όπως επιθυμεί καταβάλλοντας το προσυμφωνημένο κόστος.

Για την από κοινού διασφάλιση τους, τόσο ο καταναλωτής όσο και ο πάροχος μπορούν να συνάψουν μια σύμβαση, στην οποία να περιγράφονται οι όροι που οι συμβαλλόμενοι είναι υποχρεωμένοι να τηρούν προκειμένου να προσφέρεται η περιγραφόμενη υπηρεσία ορθά και από τις δύο πλευρές. Στις προδιαγραφές αυτές, περιλαμβάνονται θέματα σχετικά με την ασφάλεια των προσωπικών δεδομένων του καταναλωτή, την ποιότητα των υπηρεσιών που προσφέρονται καθώς και περιορισμοί αλλά και υποχρεώσεις του καταναλωτή. Ο καταναλωτής μπορεί να επιλέξει μεταξύ των παρόχων, αυτόν που θα του προσφέρει την πλέον κατάλληλη υπηρεσία, στη χαμηλότερη τιμή και με τους ευνοϊκότερους όρους. Οι όροι αυτοί, καθώς και η τιμή, συνήθως δεν είναι διαπραγματεύσιμοι παρά μόνο για περιπτώσεις που ξεφεύγουν από τη συνήθη χρήση.

Το είδος της υπηρεσίας που ο καταναλωτής θα επιλέξει να χρησιμοποιήσει, θα διαδραματίσει καθοριστικό ρόλο στις δραστηριότητες καθώς και στα σενάρια χρήσης. Στην παρακάτω εικόνα, απεικονίζονται μερικά παραδείγματα χρήσης διαφορετικών υπηρεσιών που μπορούν να προσφέρουν οι πάροχοι.



Εικόνα 15 - Διαθέσιμες υπηρεσίες του Νέφους
 Πηγή: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication500-292.pdf>

Μέσω της υπηρεσίας *Εφαρμογή σαν Υπηρεσία (SaaS)*, οι εφαρμογές που βρίσκονται στο σύννεφο είναι διαθέσιμες στους καταναλωτές μέσω διαδικτύου. Χρήστες της υπηρεσίας αυτής μπορεί να είναι, είτε οργανισμοί που προσφέρουν στους υπαλλήλους τους πρόσβαση στις εφαρμογές αυτές, είτε τελικοί χρήστες που χρησιμοποιούν απευθείας τις εφαρμογές ή ακόμη και προγραμματιστές οι οποίοι παράγουν και συντηρούν τις εφαρμογές για τους τελικούς χρήστες. Η χρέωση των υπηρεσιών μπορεί να εξαρτηθεί από τον αριθμό των χρηστών, το χρονικό διάστημα χρήσης της εφαρμογής, το εύρος ζώνης του δικτυακού καναλιού που καταλαμβάνεται ή και το μέγεθος των δεδομένων που αποθηκεύονται στην υποδομή.

Σε αντίθεση με την *Εφαρμογή σαν Υπηρεσία*, η *Πλατφόρμα σαν Υπηρεσία (PaaS)* προσφέρει τα εργαλεία αλλά και τους πόρους που απαιτούνται προκειμένου να αναπτυχθούν, να δοκιμαστούν, να διατεθούν στο κοινό αλλά και να διαχειρίζονται οι εφαρμογές των καταναλωτών. Μεταξύ των χρηστών των υπηρεσιών αυτών, περιλαμβάνονται προγραμματιστές που αναπτύσσουν εφαρμογές, που τις ελέγχουν σε συγκεκριμένα περιβάλλοντα, τεχνικοί που τις δημοσιεύουν όπως και επίσης διαχειριστές που τις διαχειρίζονται και τις παρακολουθούν. Ο τρόπος χρέωσης της υπηρεσίας δύναται να καθορίζεται από το μέγεθος της υπολογιστικής ισχύος που καταναλώνεται, τις βάσεις δεδομένων

ή τους δικτυακούς πόρους που χρησιμοποιούνται ή ακόμα και από το χρονικό διάστημα για το οποίο η υπηρεσία βρισκόταν σε χρήση.

Τέλος, η *Υποδομή σαν Υπηρεσία (IaaS)* διαθέτει πρόσβαση σε εικονικούς υπολογιστές, σε μέσα αποθήκευσης, σε υποδομές δικτύου καθώς και σε άλλες θεμελιώδεις υπολογιστικούς πόρους εντός των οποίων οι καταναλωτές μπορούν να αναπτύξουν και να εκτελέσουν το λογισμικό τους. Η υπηρεσία απευθύνεται σε προγραμματιστές εφαρμογών καθώς και διαχειριστές οι οποίοι αναπτύσσουν, εγκαθιστούν, διαχειρίζονται και παρακολουθούν εφαρμογές που εκτελούνται απευθείας στην υποδομή. Ο πάροχος τους παρέχει πρόσβαση στην αναφερόμενη υποδομή και η χρέωση τους πραγματοποιείται με βάση το μέγεθος του πόρου που χρησιμοποιούν ή τον χρόνο τον οποίο τον χρησιμοποιούν. [08]

2.1.2 Ο Πάροχος (Cloud Provider)

Ο πάροχος, είτε σαν άτομο είτε σαν επιχείρηση, είναι αυτός που ουσιαστικά καθιστά την υπηρεσία διαθέσιμη στο καταναλωτικό κοινό. Είναι αυτός που έχει στην κατοχή του την υποδομή και την διαχειρίζεται προκειμένου να προσφερθούν οι διατιθέμενες υπηρεσίες. Επιπλέον, είναι υπεύθυνος για την εκτέλεση του απαραίτητου λογισμικού καθώς και όλων των απαραίτητων διαδικασιών προκειμένου να λειτουργήσει η περιγραφόμενη υποδομή.

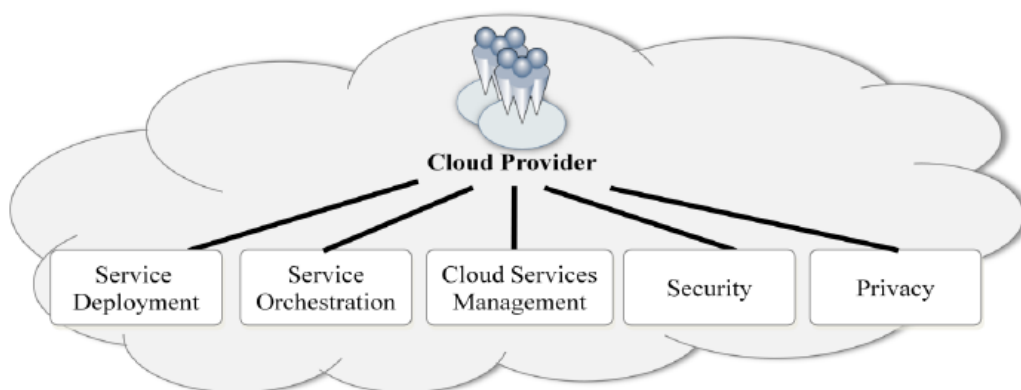
Για την Εφαρμογή σαν Υπηρεσία, ο πάροχος αναπτύσσει, παραμετροποιεί, συντηρεί και ενημερώνει τις διάφορες εφαρμογές της υποδομής ώστε να μπορούν να υποστηρίξουν τις προσφερόμενες υπηρεσίες. Παράλληλα, αναλαμβάνει την ευθύνη της διαχείρισης, τον έλεγχο των εφαρμογών αλλά και της υποδομής, ενώ ο καταναλωτής έχει περιορισμένο διαχειριστικό έλεγχο.

Στη περίπτωση της Πλατφόρμας ως Υπηρεσία, ο πάροχος διαχειρίζεται την υπολογιστική υποδομή της πλατφόρμας και εκτελεί το λογισμικό που παρέχει τα συστατικά της πλατφόρμας, όπως η στοίβα εκτέλεσης λογισμικού σε πραγματικό χρόνο, οι βάσεις δεδομένων καθώς και τα υπόλοιπα ενδιάμεσα λογισμικά. Αυτός παράλληλα, υποστηρίζει την ανάπτυξη, την εγκατάσταση και τη διαχείριση της υπηρεσίας Πλατφόρμα ως Υπηρεσία στον καταναλωτή, παρέχοντας εργαλεία όπως ολοκληρωμένα περιβάλλοντα ανάπτυξης (Integrated Development Environments), εργαλεία ανάπτυξης εφαρμογών (Software Development Tools) καθώς και άλλα εργαλεία ανάπτυξης και διαχείρισης. Από την άλλη

πλευρά, ο καταναλωτής έχει τον πλήρη έλεγχο των εφαρμογών καθώς και κάποιες δυνατότητες στο περιβάλλον φιλοξενίας, αλλά δεν έχει καμία πρόσβαση στην υποκείμενη υποδομή όπως το δίκτυο, τους εξυπηρετητές, τα λειτουργικά συστήματα ή τα μέσα αποθήκευσης.

Από την πλευρά της Υποδομής ως Υπηρεσία, ο πάροχος παρέχει την υποκείμενη υποδομή, η οποία περιλαμβάνει τους εξυπηρετητές, τα δίκτυα, τα μέσα αποθήκευσης καθώς και την υποδομή φιλοξενίας εφαρμογών. Εκτελεί το λογισμικό νέφους, το οποίο είναι αρμόδιο για να καταστήσει τους υπολογιστικούς πόρους διαθέσιμους στον καταναλωτή, διαθέτοντας ένα σύνολο διεπαφών αλλά και “γενικευμένων” (Abstract) υπολογιστικών πόρων, όπως εικονικοί υπολογιστές και εικονικές κάρτες δικτύου. Ο χρήστης της υπηρεσίας από την πλευρά του, χρησιμοποιεί τους υπολογιστικούς αυτούς πόρους, όπως οι εικονικοί υπολογιστές, για τις βασικές υπολογιστικές του ανάγκες. Σε σύγκριση με την Εφαρμογή ή την Πλατφόρμα ως Υπηρεσία, στην Υποδομή ως Υπηρεσία ο καταναλωτής έχει πρόσβαση σε θεμελιώδη τμήματα της υποδομής και κατά συνέπεια περισσότερο έλεγχο σε τμήματα όπως το λειτουργικό σύστημα και το δίκτυο. Ο πάροχος, από την άλλη πλευρά, διατηρεί τον έλεγχο στο φυσικό μέσο αλλά και στο λογισμικό νέφος, το οποίο επιτρέπει την διαχείριση της προσφερόμενης υποδομής όπως οι εξυπηρετητές, οι συσκευές δικτύου, τα μέσα αποθήκευσης, τα λειτουργικά του εκάστοτε χρήστη και την επιτήρηση των εικονικών συστημάτων.

Οι υπηρεσίες του παρόχου από την πλευρά της Υποδομής ως Υπηρεσία μπορούν να ενταχθούν σε πέντε βασικές κατηγορίες, όπως φαίνεται στην ακόλουθη εικόνα: στην ανάπτυξη των υπηρεσιών, την οργάνωση των υπηρεσιών, τη διαχείριση της υπηρεσίας νέφους, την ασφάλεια των εφαρμογών και την ιδιωτικότητα. [08]



2.1.3 Ο Σύμβουλος (Cloud Auditor)

Ο όρος αναφέρεται στον ανεξάρτητο οργανισμό, ο οποίος είναι υπεύθυνος να πραγματοποιεί ελέγχους στην υποδομή του νέφους και κατόπιν να παρουσιάζει τα συμπεράσματά του αναφορικά με το κατά πόσο ο πάροχος πληροί τις αναγνωρισμένες προϋποθέσεις. Ο έλεγχος αυτός, μπορεί να περιλαμβάνει τομείς όπως οι έλεγχοι ασφάλειας της υποδομής, η απόδοση, η ακεραιότητα των δεδομένων, κτλ.

Ο ρόλος του συμβούλου είναι απαραίτητος σε ορισμένες περιπτώσεις παρόχων οι οποίοι δεσμεύονται με ειδικές ρήτρες που τους υποχρεώνουν να επιτρέπουν στους ανεξάρτητους συμβούλους να πραγματοποιούν ελέγχους ασφαλείας. Οι έλεγχοι αυτοί αφορούν τα διαχειριστικά, λειτουργικά και τεχνικά μέτρα που έχουν αναπτυχθεί ή τις διασφαλίσεις που στόχο έχουν να προστατέψουν την εμπιστευτικότητα, την ακεραιότητα και την διαθεσιμότητα του συστήματος και των δεδομένων που φιλοξενεί. Κατά συνέπεια, ο σύμβουλος είναι σε θέση να εκτιμήσει τους ελέγχους ασφαλείας του υπολογιστικού νέφους, ώστε να καθορίσει το κατά πόσον οι έλεγχοι αυτοί έχουν υλοποιηθεί σωστά, την έκταση κατά την οποία οι έλεγχοι αυτοί αποδίδουν ικανοποιητικά και παράγουν το επιθυμητό αποτέλεσμα σεβόμενοι τις απαιτήσεις ασφαλείας του συστήματος. Παράλληλα, ο σύμβουλος είναι αυτός που θα βεβαιώσει ότι έχουν εφαρμοστεί οι κατάλληλες πολιτικές ασφαλείας και ότι δεν παραβιάζονται οι νόμοι που αφορούν τον οργανισμό.

Από τα παραπάνω γίνεται αντιληπτό, ότι η πρόσληψη ενός συμβούλου, εξασφαλίζει στον πάροχο ότι θα συμμορφώνεται ο καταναλωτής με τους νόμους αλλά και τους κανονισμούς που έχει ορίσει η εκάστοτε κυβέρνηση και θα προστατεύεται η εμπιστευτικότητα, η ακεραιότητα και η διαθεσιμότητα των δεδομένων του ατόμου σε κάθε στάδιο της ανάπτυξης αλλά και λειτουργίας της υποδομής. [08]

2.1.4 Ο Διαμεσολαβητής (Cloud Broker)

Η υπολογιστική νέφους περιλαμβάνει σύνθετους όρους και τεχνολογίες τα οποία ο μέσος καταναλωτής πιθανόν να μην είναι σε θέση να διαχειριστεί. Για την περίπτωση αυτή, ο καταναλωτής έχει τη δυνατότητα να απευθυνθεί στο διαμεσολαβητή και να ζητήσει να του παρουσιάσει τις καταλληλότερες επιλογές για τη δική του ξεχωριστή περίπτωση αντί να απευθυνθεί απευθείας στον πάροχο. Ο διαμεσολαβητής είναι αυτός που θα διαχειρίζεται τη χρήση, την απόδοση καθώς και το

τελικό αποτέλεσμα των υπηρεσιών νέφους και θα μεσολαβεί ανάμεσα στον καταναλωτή και τον πάροχο, διεκπεραιώνοντας ουσιαστικά την επικοινωνία μεταξύ των δύο.

Γενικότερα, ο διαμεσολαβητής θα μπορούσε να προσφέρει τρεις κατηγορίες υπηρεσιών:

- Διαμεσολάβηση Υπηρεσιών: Ενίσχυση μιας παρεχόμενης υπηρεσίας, βελτίωση συγκεκριμένων δυνατοτήτων αλλά και παροχής υπηρεσιών προστιθέμενης αξίας στους καταναλωτές. Πιο συγκεκριμένα, οι βελτιώσεις αυτές μπορούν αφορούν τη διαχείριση της πρόσβασης στις υπηρεσίες του νέφους, τη διαχείριση ρόλων, τις αναφορές απόδοσης, τη βελτίωση της ασφάλειας, κλπ.
- Συγκέντρωση Υπηρεσιών: Ο διαμεσολαβητής συγκεντρώνει και συνδυάζει πολλαπλές και διαφορετικές υπηρεσίες σε μία νέα, ενιαία ή σε πολλές νέες υπηρεσίες. Επιπλέον, εξασφαλίζει την ακεραιότητα των δεδομένων και φροντίζει για την ασφαλή μετακίνηση αυτών μεταξύ καταναλωτή και διαφόρων παρόχων υπηρεσιών νέφους.
- Διαιτησία Υπηρεσιών: Ο διαμεσολαβητής έχει τη δυνατότητα να επιλέξει τον πάροχο ο οποίος είναι ο πλέον κατάλληλος για την κάθε ξεχωριστή περίπτωση καταναλωτή, όχι μόνο από πλευράς απόδοσης ή υπηρεσιών, αλλά με βάση και άλλους παράγοντες, όπως κόστος, κλπ. [08]

2.1.5 Ο Φορέας

Ο φορέας λειτουργεί ως ο συνδετικός κρίκος μεταξύ του καταναλωτή και του παρόχου της υπηρεσίας, καθώς πραγματοποιεί τη διασύνδεση των δύο, και εκτελεί τις διαδικασίες που είναι απαραίτητες για τη μεταφορά των υπηρεσιών. Πιο συγκεκριμένα, ο φορέας παρέχει πρόσβαση στους καταναλωτές μέσω του δικτύου, των υπηρεσιών τηλεπικοινωνίας καθώς και των υπολοίπων συσκευών πρόσβασης. Προκειμένου ο ρόλος του να γίνει καλύτερα αντιληπτός, στο σημείο αυτό θα μπορούσε να αναφερθεί το σενάριο, όπου οι καταναλωτές μπορούν να έχουν πρόσβαση στις υπηρεσίες νέφους μέσω των Διαδικτυακών τους συσκευών, όπως υπολογιστές, laptops, κινητά τηλέφωνα κλπ. Η πρόσβαση στις υπηρεσίες αυτές πραγματοποιείται μέσω του Διαδικτύου και των τηλεπικοινωνιακών παρόχων ή ακόμα και μέσω υψηλής χωρητικότητας συσκευών μεταφοράς αρχείων, όπου η μεταφορά γίνεται με

φυσικό τρόπο μετάδοσης (Hard Disk Drives). Ο φορέας είναι δυνατόν να δεσμεύεται από καθορισμένες ρήτρες, σύμφωνα με τις οποίες θα παρέχονται οι υπηρεσίες στους καταναλωτές και θα υποχρεώνουν το φορέα να παρέχει στον καταναλωτή δεσμευμένες και ασφαλείς γραμμές. [08]

2.2 Google Cloud Platform

Η υποδομή που θα φιλοξενήσει την εφαρμογή, η οποία θα αναπτυχθεί στα πλαίσια της παρούσας εργασίας, θα δημιουργηθεί μέσω της πλατφόρμας της Google. Η εταιρεία επιλέχθηκε μεταξύ των Amazon Web Services και Microsoft Azure, ως μια αντιπροσωπευτική επιλογή του Cloud Computing.

Η εταιρεία ξεκίνησε τη δράση της στο τομέα της υπολογιστικής νέφους τον Απρίλιο του 2008, όταν ανακοίνωσε τη διάθεση της υπηρεσίας App Engine, μια πλατφόρμα για την ανάπτυξη και τη φιλοξενία εφαρμογών διαδικτύου στους δικούς της εξυπηρετητές. Η υπηρεσία προσφέρονταν με τη μορφή Πλατφόρμα ως Υπηρεσία και επέτρεπε την ανάπτυξη εφαρμογών σε Go, PHP, Java, Python, Node.js, .NET και Ruby καθώς και την φιλοξενία αυτών σε πολλαπλούς εξυπηρετητές. Παράλληλα, προσέφερε την δυνατότητα εκτέλεσης και άλλων γλωσσών προγραμματισμού μέσω ειδικά σχεδιασμένων περιβαλλόντων (custom runtimes). Η υπηρεσία υποστήριζε την αυτόματη κλιμάκωση στην περίπτωση που η υπάρχουσα υποδομή δεν μπορούσε να καλύψει την αυξανόμενη ζήτηση και η χρέωση της υπηρεσίας ήταν ικανοποιητική σε συνάρτηση με τις απαιτήσεις σε χώρο, απαιτούμενο εύρος ζώνης δικτύου ή απαιτούμενο χρόνο εκτέλεσης της εφαρμογής. Στο πέρασμα των χρόνων, η εταιρεία εμπλούτισε την υπηρεσία με νέες υπηρεσίες και το διάστημα συγγραφής της παρούσας εργασίας προσέφερε πάνω από εκατό προϊόντα. [09]

2.3 Δημιουργία Υποδομής

Για τη φιλοξενία της εφαρμογής, που θα αναπτυχθεί στα πλαίσια της παρούσας εργασίας, επιλέχθηκε η πρόταση της εικονικής μηχανής (Virtual Machine) σαν την πλέον κατάλληλη. Μέσω της επιλογής αυτής, η εταιρεία προσφέρει ένα εικονικό μηχάνημα που θα εγκατασταθεί λειτουργικό σύστημα

εφοδιασμένο με τη σουίτα LAMP (Linux, Apache, MySQL, PHP), η οποία περιέχει όλα τα απαιτούμενα προγράμματα.

Static website	Cloud Storage Firebase Hosting	Cloud Storage bucket	HTTP(S) optional	Automatically	Cloud Logging
Virtual machines	Compute Engine	Cloud SQL Admin API, Cloud Storage API, Datastore API, and Cloud Bigtable API, or you can use another external storage provider. Hard-disk-based persistent disks, called <i>standard persistent disks</i> , and solid-state persistent disks (SSD).	HTTP(S) TCP Proxy SSL Proxy IPv6 termination Network Cross-region Internal	Automatically with managed instance groups	Cloud Logging Cloud Monitoring Monitoring console
Containers	GKE	Similar to Compute Engine but interacts with persistent disks differently	Network HTTP(S)	Cluster autoscaler	Cloud Logging Cloud Monitoring Monitoring console
Managed platform	App Engine	Google Cloud services such as Cloud SQL, Firestore, Cloud Storage, and accessible third-party databases	HTTP(S) Managed by Google	Managed by Google	Cloud Logging Cloud Monitoring Monitoring console
Serverless	Cloud Run	Google Cloud services such as Cloud SQL, Firestore, Cloud Storage, and accessible third-party databases	HTTP(S) Managed by Google	Managed by Google	Cloud Logging Cloud Monitoring Monitoring console

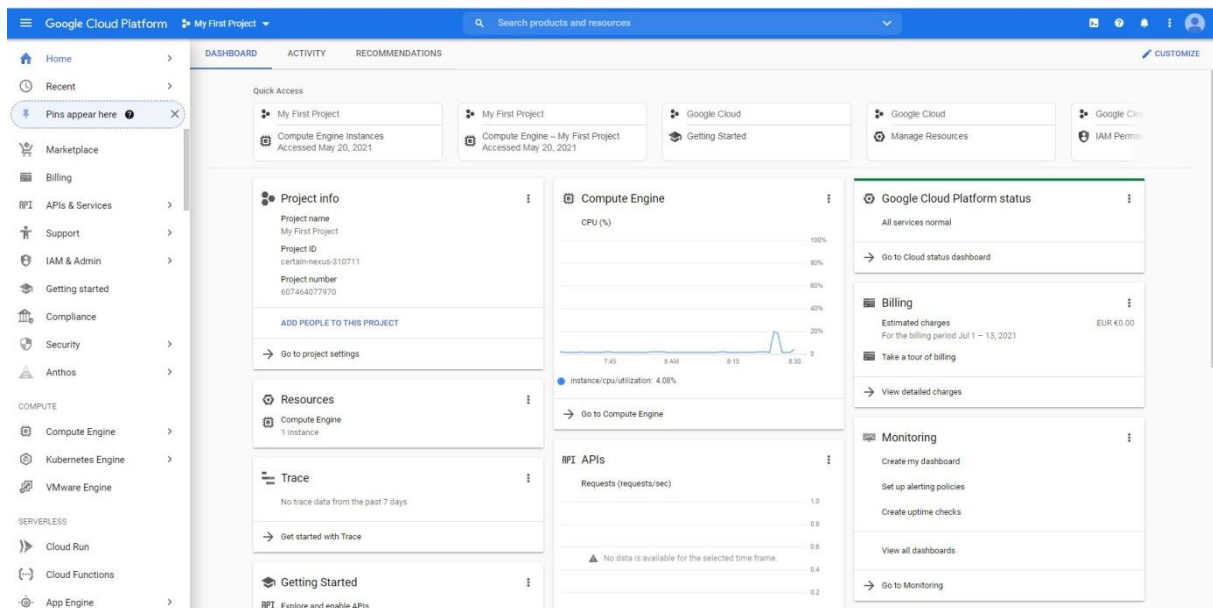
Εικόνα 17 - Επιλογές Web Hosting από τη Google.
Πηγή : https://cloud.google.com/architecture/web-serving-overview?hl=en_US

Το λειτουργικό σύστημα που θα εγκατασταθεί παρέχεται σε μορφή εικόνας (Image) από την εταιρεία Bitnami. Οι εικόνες που παρέχει η εταιρεία, είναι πλήρως συμβατές με τα συστήματα της Google, με το συγκεκριμένο που θα χρησιμοποιηθεί να αποτελεί μία έκδοση Debian που διαθέτει προεγκατεστημένα όλα τα απαραίτητα λογισμικά:

- Apache Web Server έκδοσης 2.2

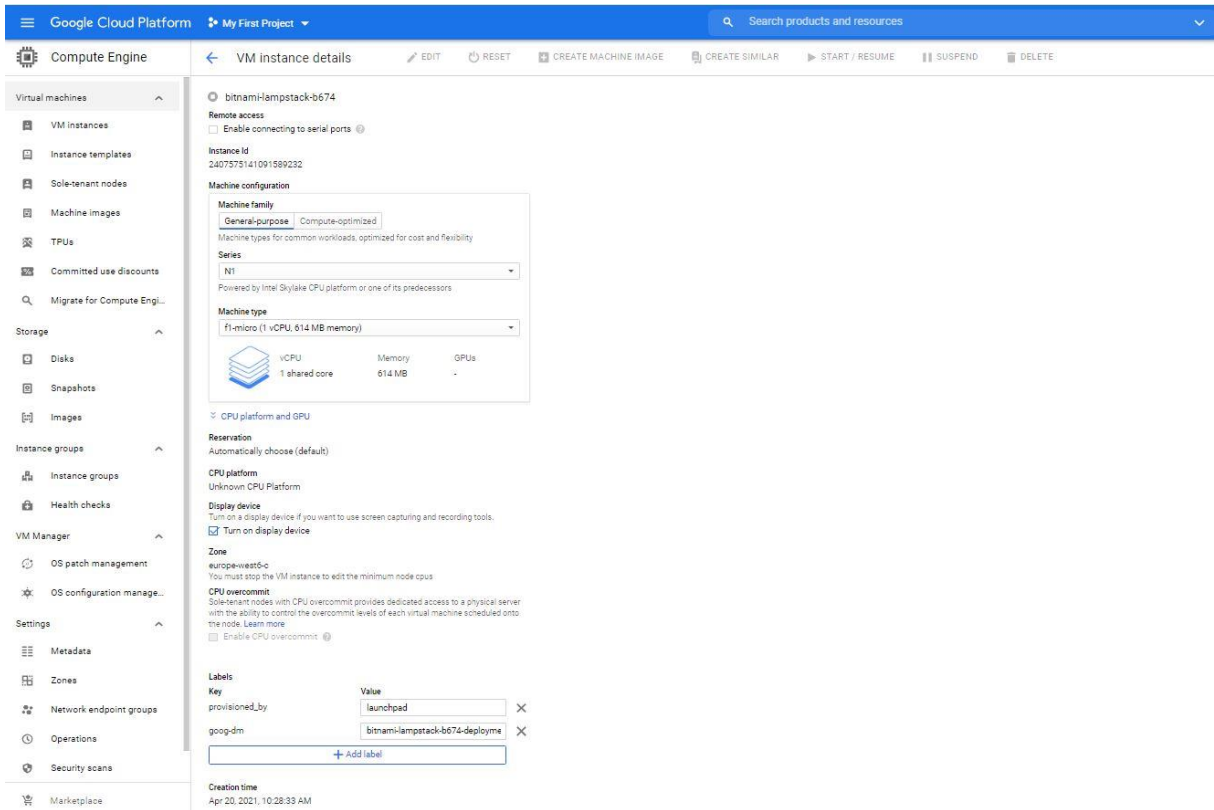
- Σύστημα διαχείρισης Βάσεων Δεδομένων MySQL έκδοσης 5.7
- PHP 7.4
- Συμπληρωματικό και υποστηρικτικό λογισμικό, όπως το phpMyAdmin έκδοσης 5

Για τη χρήση της υπηρεσίας, απαιτείται από την εταιρεία η εγγραφή του εκάστοτε χρήστη. Όταν αυτό ολοκληρωθεί, ο χρήστης εφοδιάζεται με ένα χρηματικό ποσό που του επιτρέπει τη χρήση της υπηρεσίας δωρεάν για τρεις μήνες. Παράλληλα, ο χρήστης θα πρέπει να εγγραφεί και στον ιστότοπο της εταιρείας Bitnami για τη χρήση του αρχείου εικόνας που περιέχει το λειτουργικό και για τις υπόλοιπες εφαρμογές. Αναλυτικές οδηγίες για τη δημιουργία της υποδομής με το λειτουργικό σύστημα παρέχονται από διάφορες πηγές, μεταξύ άλλων και της ίδιας της εταιρείας. Μεταξύ αυτών, αξίζει να αναφερθεί και η προσπάθεια της εταιρείας ApacheFriends, μιας μη κερδοσκοπικής εταιρείας που προωθεί τον εξυπηρετητή Apache, στον οποίο βασίζεται η εφαρμογή της παρούσας εργασίας. [<https://www.apachefriends.org/docs/hosting-xampp-on-google.html>]



Εικόνα 18 - Η πλατφόρμα της Google μετά τη δημιουργία της υποδομής

Με την ολοκλήρωση της διαδικασίας, ο χρήστης έχει στη διάθεση του ένα πλήρες λειτουργικό υπολογιστή, έτοιμο να συνδεθεί πάνω του και να αναπτύξει ή να μεταφέρει την εφαρμογή του. Μέσω της πλατφόρμας, έχει τη δυνατότητα να εκτελέσει το σύνολο των ενεργειών που προσφέρει ο πάροχος πάνω στην εικονική μηχανή του.



Εικόνα 19 - Επεξεργασία εικονικής μηχανής


The screenshot displays the 'VM instance details' configuration page in Google Cloud Platform. The interface is organized into a left-hand navigation sidebar and a main configuration area. The sidebar includes categories like 'Virtual machines', 'Storage', 'Instance groups', 'VM Manager', and 'Settings'. The main area is titled 'VM instance details' and contains several sections:

- Additional disks:** A dropdown menu set to 'Custom' and a text input field containing 'boot'. Below are buttons for '+ Add new disk' and '+ Attach existing disk'.
- Local disks:** A dropdown menu set to 'None'.
- Shielded VM:** A section with a 'None' dropdown and three checkboxes: 'Turn on Secure Boot', 'Turn on vTPM', and 'Turn on Integrity Monitoring', all of which are currently unchecked.
- Availability policies:** A section with a 'Preemptibility' dropdown set to 'Off (recommended)', an 'On host maintenance' dropdown set to 'Migrate VM instance (recommended)', and an 'Automatic restart' dropdown set to 'On (recommended)'.
- Custom metadata:** A table with 11 rows of key-value pairs, each with a delete icon (X) on the right. The keys include 'bitnami-key', 'bitnami-name', 'bitnami-version', 'bitnami-url', 'bitnami-description', 'bitnami-default-ports', 'bitnami-application-pi', 'bitnami-base-passwo', 'startup-script', 'sshKeys', and 'created_at'.
- SSH Keys:** A section with a checked checkbox for 'Block project-wide SSH keys' and a note stating 'The custom metadata 'sshKeys' blocks project-wide SSH keys'. Below, it says 'You have 0 SSH keys' with a 'Show and edit' link.
- Service account:** A dropdown menu currently set to 'No service account'.
- Access scopes:** A section with a note: 'Select a service account to enable API access'.


At the bottom of the configuration area, there are 'Save' and 'Cancel' buttons.


Εικόνα 20 - Επιπλέον επιλογές επεξεργασίας της εικονικής μηχανής

Application Info



LAMP 7.4.16-18
LAMP is an open source software stack that provides a framework for creating PHP-based high-performance websites and applications with ease...
[Learn More](#)


 [GO TO APPLICATION](#)
LAUNCHES IN A NEW WINDOW

 **CREDENTIALS** ⓘ


PASSWORD	wad40izzEkfS	HIDE
PORTS	80, 443	


Server Info


Running a minute ago





34.65.141.100

 **F1-MICRO**
\$7.74/MO (0.011/HR) ⓘ

 **MAGNETIC DISK**
10 GB (\$0.52/MO)

 **EUROPE-WEST6-C**
REGION

 **\$8.26**
ESTIMATED MONTHLY COST

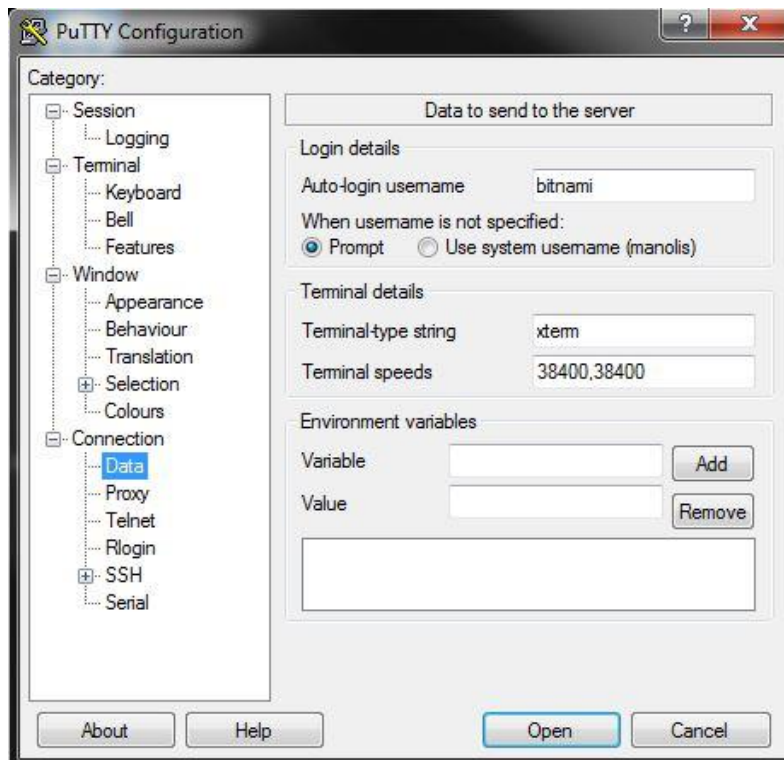
 [LAUNCH SSH CONSOLE](#)
DOWNLOAD KEY: [.PEM](#) [.PPK](#)
[Show SSH command](#)

[Show More](#) ▾

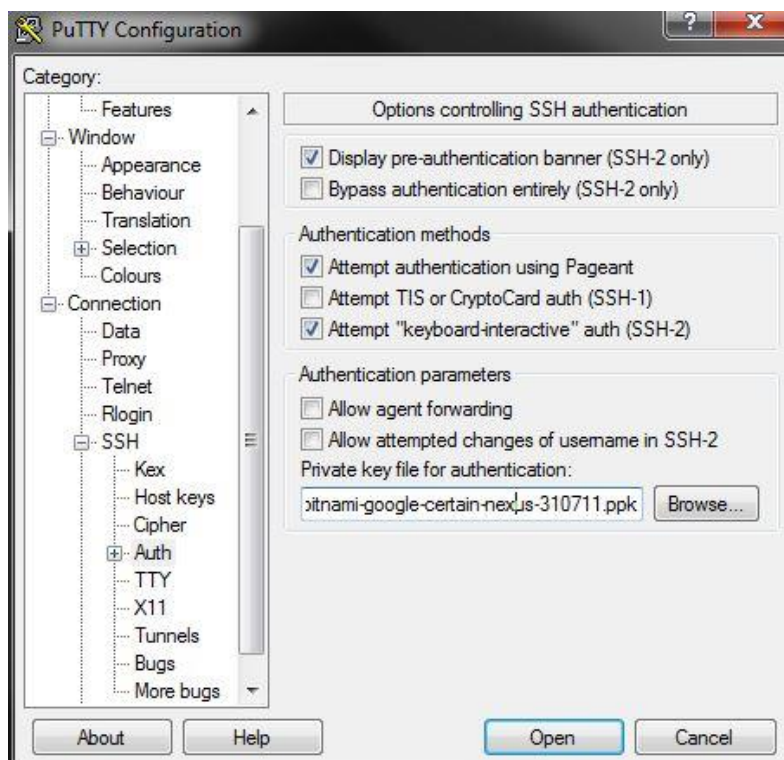
[REBOOT](#) [SHUTDOWN](#) [DELETE](#)

Εικόνα 21 - Το εικονικό μηχανήμα που δημιουργείται μετά την ολοκλήρωση της διαδικασίας όπου φαίνονται ο τύπος, η IP διεύθυνση, και τα χαρακτηριστικά του μηχανήματος

Για να μπορέσει να χρησιμοποιηθεί η αρχιτεκτονική, ο χρήστης χρειάζεται να συνδεθεί πάνω της μέσω πρωτοκόλλου SSH. Για τη χρήση του, μπορούν να χρησιμοποιηθούν διάφορες εφαρμογές, με τη δημοφιλέστερη εξ αυτών, το Putty [<https://www.putty.org/>]. Για τη σύνδεση, θα χρησιμοποιηθεί σαν όνομα χρήστη το 'bitnami' και το κλειδί ppk που δίνεται από την εταιρεία.



Εικόνα 22 - Διαμόρφωση της εφαρμογής Putty. Εισαγωγή του ονόματος χρήστη στην εφαρμογή



Εικόνα 23 - Εισαγωγή κλειδιού στην εφαρμογή

```
bitnami@bitnami-lampstack-b674: ~
Using username "bitnami".
Authenticating with public key "bitnami-hosting-key"
Linux bitnami-lampstack-b674 4.19.0-17-cloud-amd64 #1 SMP Debian 4.19.194-1 (2021-06-10) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

  _ _ _ _ _
 | | | | |
 | | | | |
 | | | | |
 | | | | |

*** Welcome to the Bitnami LAMP 7.4.16-18 ***
*** Documentation: https://docs.bitnami.com/google/infrastructure/lamp/ ***
*** https://docs.bitnami.com/google/ ***
*** Bitnami Forums: https://community.bitnami.com/ ***
Last login: Wed Jul 14 06:26:15 2021 from 5.203.210.163
bitnami@bitnami-lampstack-b674:~$
bitnami@bitnami-lampstack-b674:~$
bitnami@bitnami-lampstack-b674:~$ █
```

Εικόνα 24 - Σύνδεση μέσω SSH

Πλέον ο χρήστης μπορεί να δημιουργήσει ιστοσελίδες στον εξυπηρετητή. Για την προβολή των χαρακτηριστικών του εξυπηρετητή δημιουργείται η σελίδα `phrinfo.php`. Εν συνεχεία, ο χρήστης πληκτρολογώντας την διεύθυνση στην εφαρμογή φυλλομετρητή του υπολογιστή του, έχει πρόσβαση στη σελίδα, όπου εκεί μπορεί να δει τα χαρακτηριστικά του εξυπηρετητή.

PHP Version 7.4.16

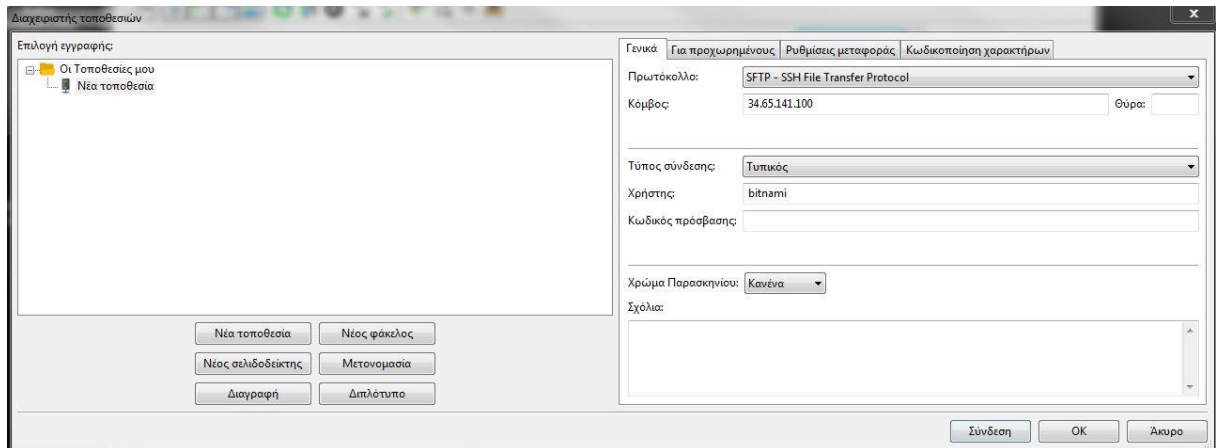
System	Linux bitnami-lampstack-b674 4.19.0-16-cloud-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64
Build Date	Apr 14 2021 21:22:06
Configure Command	'/bitnami/blacksmith-sandbox/php-7.4.16/configure' '--prefix=/opt/bitnami/php' '--with-imagick=/bitnami/blacksmith-sandbox/imagick-2007.0.0' '--with-imap-ssl' '--with-zlib-dir' '--with-zlib' '--with-libxml-dir=/usr' '--enable-soap' '--disable-rpath' '--enable-inline-optimization' '--with-bz2' '--enable-sockets' '--enable-pcntl' '--enable-xml' '--enable-bcmath' '--with-pdo-mysql=mysqlnd' '--with-mysqli=mysqlnd' '--with-png-dir=/usr' '--with-openssl' '--with-libdir=/lib/x86_64-linux-gnu' '--enable-ftp' '--enable-calendar' '--with-gettext' '--with-xmllib' '--with-xsl' '--enable-fpm' '--with-fpm-user=daemon' '--with-fpm-group=daemon' '--enable-mbstring' '--enable-cgi' '--enable-ctype' '--enable-session' '--enable-mysqlnd' '--enable-intl' '--with-iconv' '--with-pdo_sqlite' '--with-sqlite3' '--with-readline' '--with-gmp' '--with-curl' '--with-pdo-pgsql=shared' '--with-pgsql=shared' '--with-config-file-scan-dir=/opt/bitnami/php/etc/conf.d' '--enable-simplexml' '--with-sodium' '--enable-gd' '--with-pear' '--with-freetype' '--with-jpeg' '--with-webp' '--with-zip' '--with-pdo-dblib=shared' '--with-tidy' '--with-ldap=/usr' '--enable-apcu=shared' 'PKG_CONFIG_PATH=/opt/bitnami/common/lib/pkgconfig'
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/opt/bitnami/php/lib
Loaded Configuration File	/opt/bitnami/php/etc/php.ini
Scan this dir for additional .ini files	/opt/bitnami/php/etc/conf.d
Additional .ini files parsed	(none)
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v3.4.0, Copyright (c) Zend Technologies
 with Zend OPcache v7.4.16, Copyright (c), by Zend Technologies

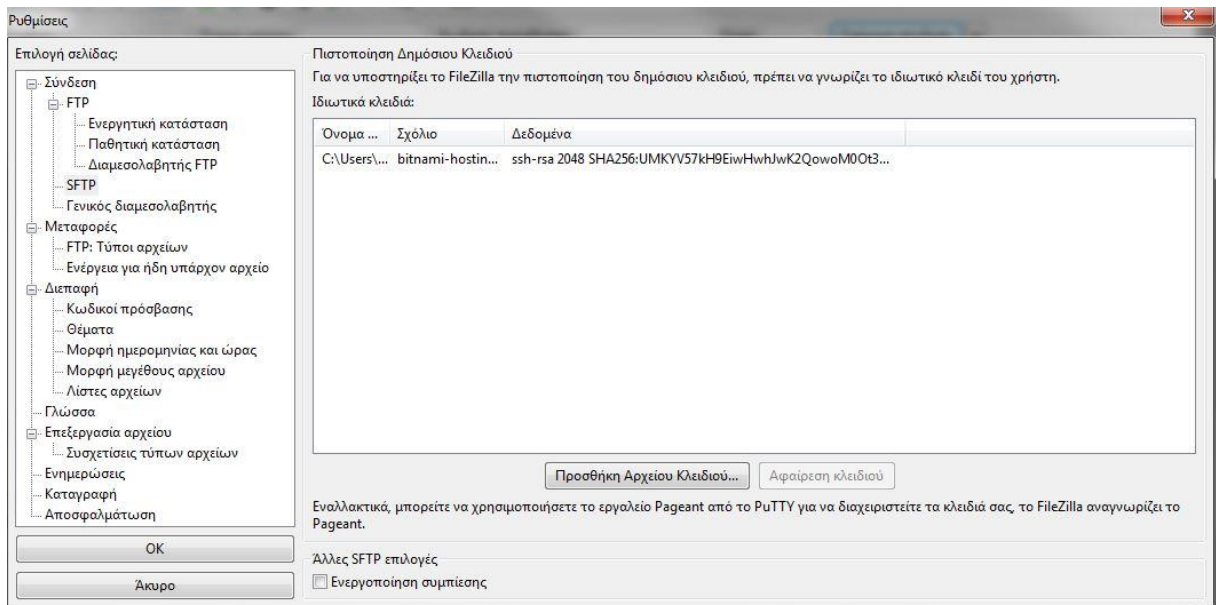
Configuration

Εικόνα 25 - Η σελίδα phpinfo του εξυπηρετητή

Για τη μεταφορά των αρχείων της εφαρμογής απαιτείται η χρήση εφαρμογής τύπου ftp client. Στα πλαίσια της παρούσας εργασίας θα χρησιμοποιηθεί αυτή της εταιρείας FileZilla η οποία διατίθεται δωρεάν μέσω του ιστοτόπου της εταιρείας [<https://filezilla-project.org/download.php?platform=win64>]. Για να πραγματοποιηθεί η σύνδεση θα δοθεί σαν όνομα χρήστη το 'bitnami' και το κλειδί prk που δίνεται από την εταιρεία.

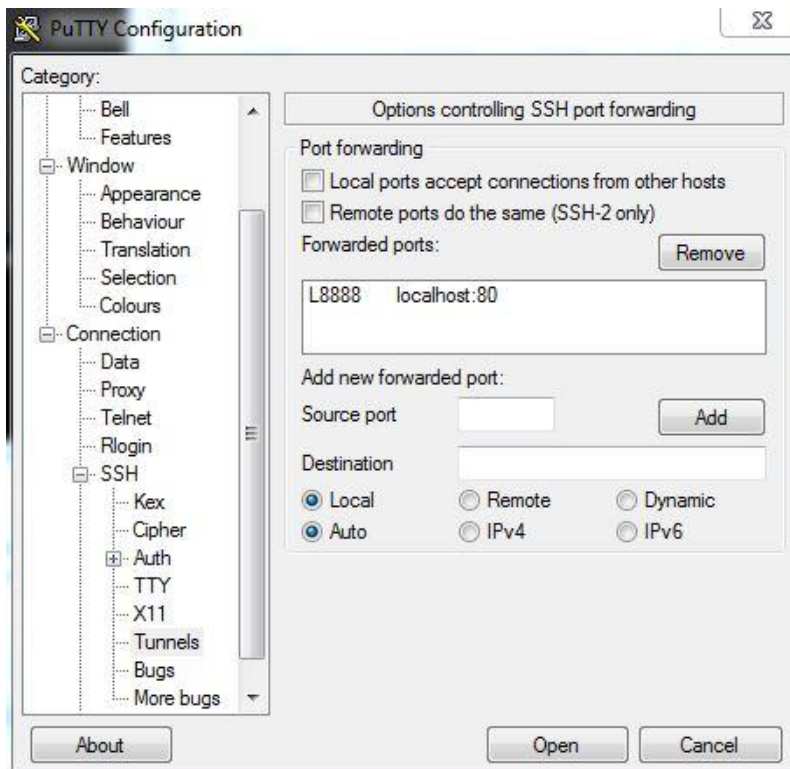


Εικόνα 26 - Παραμετροποίηση εφαρμογής Filezilla. Εμφανίζονται η διεύθυνση του εξυπηρετητή και το όνομα χρήστη



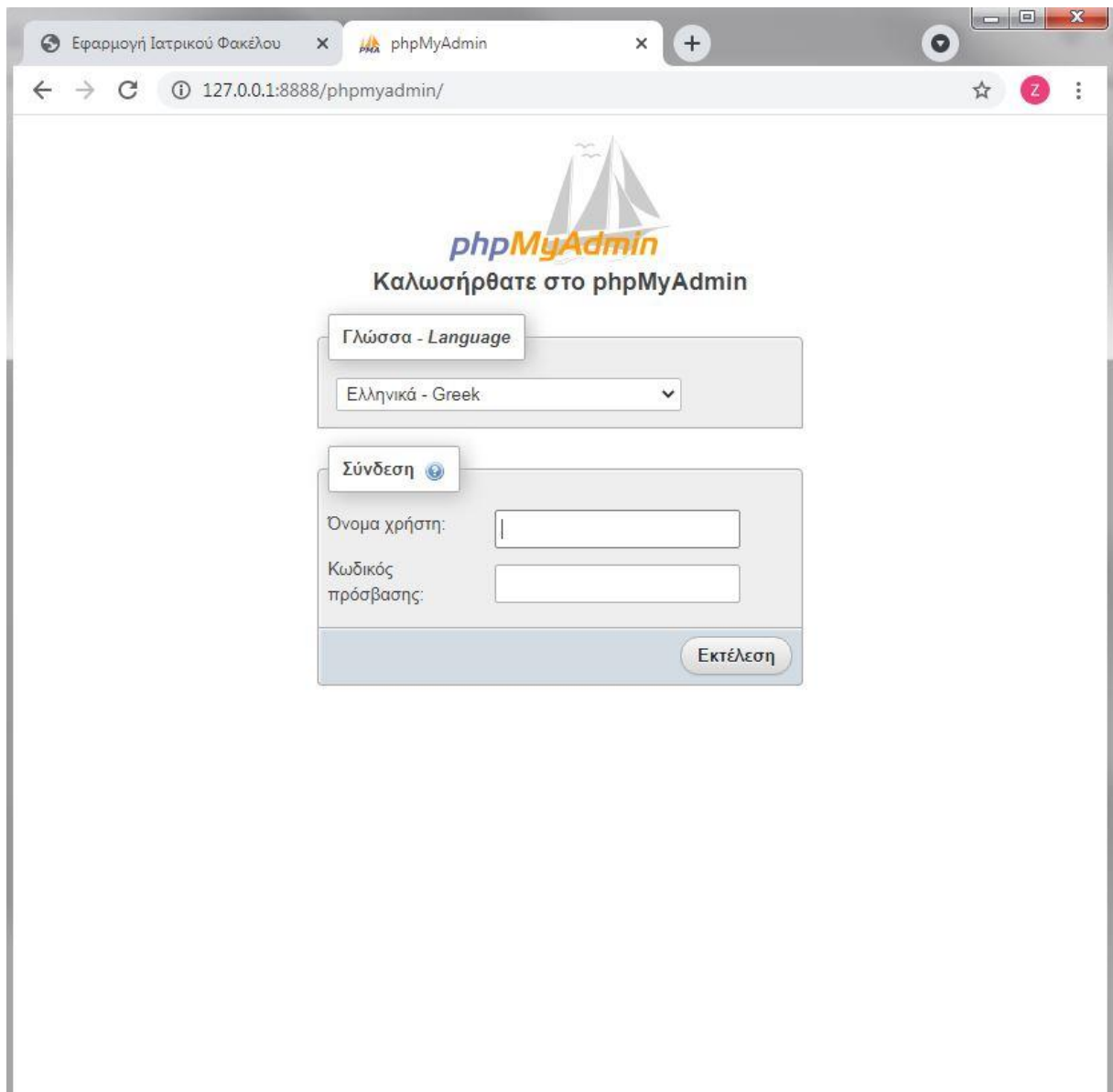
Εικόνα 27 - Εισαγωγή του αρχείου κλειδιού στην επιλογή SFTP

Η διαχείριση της βάσης δεδομένων γίνεται μέσω της εφαρμογής phpMyAdmin. Για λόγους ασφαλείας, ο χρήστης θα πρέπει να δημιουργήσει ένα εικονικό ασφαλές τούνελ για να συνδεθεί.



Εικόνα 28 - Δημιουργία ασφαλούς τούνελ για χρήση της εφαρμογής phpMyAdmin

Ακολούθως, μπορεί να συνδεθεί στην εφαρμογή πληκτρολογώντας στην εφαρμογή του φυλλομετρητή του τη διεύθυνση 127.0.0.1:8888/phpmyadmin και σαν διαπιστευτήρια στο όνομα χρήστη «user» και ως κωδικό το κλειδί prk που δώσαμε και στα προηγούμενα βήματα.



Εικόνα 29 - Σύνδεση στην εφαρμογή phpMyAdmin

3. Η εφαρμογή

3.1 Περιγραφή Προβλήματος

Η εφαρμογή επιτρέπει στους χρήστες της να διατηρούν σε ένα κεντρικό σημείο δεδομένα που αφορούν την υγεία τους, ώστε να είναι διαθέσιμα οποιαδήποτε στιγμή αυτά απαιτηθούν. Πιο συγκεκριμένα οι χρήστες, αφού εγγραφούν στην εφαρμογή, αποκτούν μοναδικούς κωδικούς και στη συνέχεια χρησιμοποιώντας τους κωδικούς αυτούς πιστοποιούνται και εισέρχονται στην εφαρμογή. Έπειτα, θα έχουν την δυνατότητα να επεξεργαστούν τα στοιχεία τους και να τα ενημερώσουν, να αποθηκεύσουν τις διάφορες εξετάσεις τους και τέλος να δώσουν πρόσβαση σε συγκεκριμένους ιατρούς στα δεδομένα αυτά.

Παράλληλα, η εφαρμογή επιτρέπει σε ιατρούς να εισέρχονται στην εφαρμογή, αφού πρώτα πιστοποιηθούν και αποκτήσουν κωδικούς, με σκοπό την μελέτη των δεδομένων των χρηστών που τους έχουν επιλέξει. Με τον τρόπο αυτό, οι ιατροί θα έχουν πλήρη πρόσβαση στο ιατρικό ιστορικό των ασθενών τους και θα μπορούν με τον τρόπο αυτόν να σχηματίσουν μια πλήρη κλινική εικόνα αυτών, με σκοπό να επιτελέσουν ορθά το έργο τους. Από την άλλη πλευρά, ο ασθενής απαλλάσσεται από την ανάγκη να θυμάται από μνήμης ή να φέρει μαζί του τον πλήρη φάκελο του ιστορικού του κάθε φορά που επισκέπτεται έναν ιατρό.

Καθίσταται εύκολα λοιπόν αντιληπτή στον αναγνώστη, η σπουδαιότητα μιας τέτοιας εφαρμογής. Τα δεδομένα υγείας που παράγονται στη ζωή του ασθενούς είναι πολλά και η διατήρηση του συνόλου αυτών για μεγάλα χρονικά διαστήματα, είναι ζήτημα υψίστης σημασίας. Η παράληψη έστω και μιας λεπτομέρειας για τον ασθενή, όπως μια αλλεργία, μια αγωγή που μπορεί να λαμβάνει ή μια ασθένεια που ενδεχομένως να έχει νοσήσει στο παρελθόν, μπορεί να οδηγήσει τον ιατρό σε εντελώς λανθασμένες αποφάσεις με απρόβλεπτες συνέπειες για την υγεία του ασθενούς. Επιπλέον, ο ιατρός δεν μπορεί να βασίζεται στην ενημέρωση που θα λάβει από τον ασθενή αναφορικά με το ιστορικό του, αφού είναι πολύ πιθανόν ο ασθενής να μην είναι σε θέση να του γνωστοποιήσει το ιστορικό του λόγω απώλειας αισθήσεων, μειωμένων αντιδράσεων, κλπ. Κατά συνέπεια, θα πρέπει ο ιατρός να μπορεί να έχει απρόσκοπτη πρόσβαση στο σύνολο των ιατρικών δεδομένων του ασθενούς που τον έχει επιλέξει και του έχει δώσει την πρόσβαση.

3.2 Ανάλυση Απαιτήσεων

Στη παράγραφο αυτή, θα περιγραφούν οι απαιτήσεις της εφαρμογής ώστε οι σχεδιαστές της να γνωρίζουν τι πρέπει να κάνουν αλλά και τι θα απαιτηθεί, προκειμένου η εφαρμογή να είναι λειτουργική. Οι απαιτήσεις αυτές ομαδοποιούνται σε λειτουργικές και μη λειτουργικές. Στην κατηγορία των λειτουργικών εντάσσονται οι λειτουργίες της εφαρμογής, ενώ στη κατηγορία των μη λειτουργικών, οι απαιτήσεις της σε πόρους, διαθεσιμότητα, κόστος κλπ.

3.2.1 Λειτουργικές Απαιτήσεις

Η εφαρμογή θα πρέπει να υποστηρίζει τις εξής λειτουργίες:

- Δύο διακριτούς ρόλους χρηστών, τον ιατρό και τον χρήστη. Οι χρήστες επιλέγουν τους ιατρούς που θα έχουν πρόσβαση στα δεδομένα τους και θα εισάγουν στο σύστημα τα δεδομένα αυτά. Οι ιατροί από την άλλη πλευρά, θα έχουν πρόσβαση μόνο στα δεδομένα των χρηστών που τους έχουν επιλέξει.
- Οι χρήστες θα μπορούν να εγγράφονται στο σύστημα και να δημιουργούν ένα ιδιωτικό προφίλ. Για την χρήση του συστήματος θα είναι απαραίτητη η χρήση μοναδικού ονόματος χρήστη και κωδικού πρόσβασης.
- Μετά την επιτυχή εγγραφή τους στο σύστημα, οι χρήστες θα μπορούν να εισάγουν τα προσωπικά τους στοιχεία (όνομα, επώνυμο, βάρος, ηλικία) αλλά και να τα ενημερώνουν, όποτε αυτό κριθεί απαραίτητο.
- Οι χρήστες θα έχουν τη δυνατότητα να εισάγουν στο σύστημα τις διάφορες εξετάσεις τους. Η διαδικασία θα γίνεται με τρόπο απλό στη χρήση, ώστε να διευκολύνεται η λειτουργία του συστήματος.
- Οι χρήστες θα μπορούν να επιλέγουν τους ιατρούς τους από το σύνολο των διαθέσιμων ιατρών. Η επιλογή θα γίνεται με βάση κριτήρια που οι ίδιοι θα επιλέγουν και παράλληλα θα έχουν και τη δυνατότητα να ανανεώσουν τις επιλογές τους στη περίπτωση που αυτό κριθεί

αναγκαίο (στη περίπτωση που για οποιοδήποτε λόγο επιλέξουν να αντικαταστήσουν τον συγκεκριμένο ιατρό με κάποιον άλλο.).

- Παράλληλα με τους χρήστες, και οι ιατροί θα μπορούν να εγγράφονται προκειμένου να χρησιμοποιήσουν το σύστημα και με τη χρήση των κωδικών που θα έχουν αποκτήσει, θα μπορούν να εισάγονται σε αυτό.
- Κατά τη διαδικασία της εγγραφής τους, οι ιατροί εισάγουν προσωπικά στοιχεία βάσει των οποίων θα επιλεγθούν από τους ασθενείς. Τα στοιχεία αυτά περιλαμβάνουν την ειδικότητα και τον τόπο εργασίας τους, ώστε να μπορεί να αποφασίσει ο ασθενής ποιος ιατρός του ταιριάζει καλύτερα.
- Οι ιατροί, εφόσον επιλεγθούν, θα μπορούν να έχουν πλήρη πρόσβαση στο σύνολο των δεδομένων που ο ασθενής έχει εισάγει στο σύστημα. Η διαδικασία θα πρέπει να γίνεται οποιαδήποτε στιγμή απαιτηθεί και με τρόπο που να μην απαιτεί εξειδικευμένες γνώσεις από την πλευρά του ιατρού.

3.2.2 Μη Λειτουργικές Απαιτήσεις

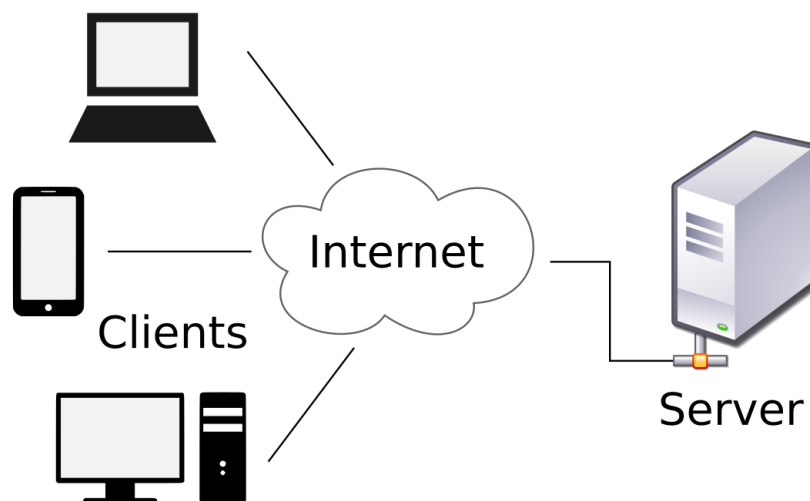
Στις Μη Λειτουργικές Απαιτήσεις, εντάσσονται τα παρακάτω γνωρίσματα:

- Το σύστημα θα πρέπει να βασιστεί σε αξιόπιστες τεχνολογίες ελεύθερου λογισμικού ώστε να μην απαιτεί κόστος για την υλοποίησή του.
- Θα πρέπει να διαθέτει επεξεργαστή μονού πυρήνα, ταχύτητας τουλάχιστον 3GHz, μνήμη 614 MB, 10GB σκληρό δίσκο και λειτουργικό σύστημα βασισμένο σε Linux.
- Θα πρέπει να είναι διαθέσιμο 24/7.
- Για την φιλοξενία του συστήματος, θα χρησιμοποιηθεί η υπολογιστική νέφους ώστε να επωφελείται από το σύνολο των πλεονεκτημάτων που η τεχνολογία αυτή έχει να προσφέρει (άμεση αντιμετώπιση σε περίπτωση αστοχίας υλικού, άμεση αύξηση των διαθέσιμων πόρων όταν αυτό απαιτηθεί, κ.α.).

- Προστασία των χρηστών αλλά και των δεδομένων (αποθήκευση του κωδικού χρήστη στη βάση σε μορφή Hash, προστασία από επιθέσεις τύπου SQL Injection, CSS, κλπ.).
- Πρόσβαση από οποιαδήποτε συσκευή δύναται να ανοίξει φυλλομετρητή.

3.3 Αρχιτεκτονική Συστήματος

Το σύστημα θα αποτελείται από τον εξυπηρετητή και τον πελάτη. Με τον τρόπο αυτό επιτυγχάνεται ένας σαφής διαχωρισμός μεταξύ των δύο, ο οποίος επιτρέπει την επικοινωνία μεταξύ διαφορετικών συστημάτων καθώς και την αδιάκοπη λειτουργία της εφαρμογής σε περιπτώσεις όπως αναβάθμιση των σταθμών. Για την επικοινωνία των δύο διακριτών μερών θα χρησιμοποιείται το πρωτόκολλο HTTP.

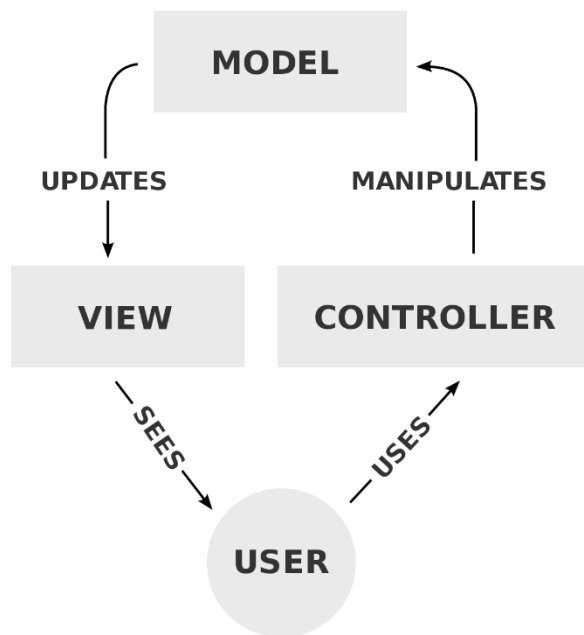


Εικόνα 30 - Το μοντέλο Πελάτη - Διακομιστή (Client Server)
 Πηγή : https://en.wikipedia.org/wiki/Client%E2%80%93server_model

3.3.1 Ο Εξυπηρετητής

Ο εξυπηρετητής αποτελείται από δύο διαφορετικά συστήματα που φιλοξενούνται στο ίδιο εικονικό μηχάνημα, τον Εξυπηρετητή Ιστοσελίδων (Web Server) και το Σύστημα Διαχείρισης Βάσεων

Δεδομένων (DataBase Server). Τον εξυπηρετητή ιστοσελίδων θα εκτελεί μια από τις πλέον καταξιωμένες εφαρμογές ανοιχτού κώδικα, ο Apache2. Αντίστοιχα και ο DataBase Server θα εκτελείτε μέσω της εφαρμογής MySQL. Η γλώσσα προγραμματισμού που θα χρησιμοποιηθεί στο Web Server θα είναι η PHP, μία επίσης ευρέως διαδεδομένη και ανοιχτού κώδικα, γλώσσα προγραμματισμού. Η εφαρμογή θα ακολουθεί την προγραμματιστική τεχνική MVC που επιτρέπει τον σαφή διαχωρισμό μεταξύ των δομών της αρχιτεκτονικής της εφαρμογής DataBase και του Client. Παράλληλα, η εφαρμογή θα πλαισιωθεί με τεχνολογίες που προσφέρουν οι JavaScript, CSS, Ajax, jquery, κλπ.



Εικόνα 31 : Προγραμματιστική Τεχνική MVC
 Πηγή : <https://el.wikipedia.org/wiki/Model-view-controller>

3.3.2 Ο Πελάτης

Ο πελάτης μπορεί να είναι ένας ηλεκτρονικός υπολογιστής, ένα tablet, ή ένα κινητό τηλέφωνο, εκτελώντας την εφαρμογή του φυλλομετρητή ώστε να πραγματοποιεί αιτήματα τύπου GET ή POST μέσω του πρωτοκόλλου HTTP στον εξυπηρετητή. Ακολούθως, θα λαμβάνει την απάντηση και θα την παρουσιάζει στον χρήστη μέσω της ίδιας εφαρμογής.

3.3.3 Η Βάση Δεδομένων

Πίνακας Χρήστες

Ο πίνακας περιέχει τα στοιχεία των ασθενών, όπως το ΑΜΚΑ, το Όνομα, το Επώνυμο, την Ημερομηνία Γέννησης, το Ύψος, το Βάρος, την Διεύθυνση Ηλεκτρονικού Ταχυδρομείου (email), τον Αριθμό Κινητού Τηλεφώνου και τον Κωδικό Πρόσβασης. Από τον κωδικό του χρήστη αποθηκεύεται το Hash του, ώστε οι διαχειριστές της εφαρμογής να μην τον γνωρίζουν και να μην μπορούν να χρησιμοποιήσουν την εφαρμογή με τα στοιχεία των χρηστών.

Πίνακας Εξετάσεις

Ο πίνακας αυτός περιέχει τις εξετάσεις των ασθενών σε μορφή BLOB, μια περιγραφή της εξέτασης και ένα πεδίο με τον κωδικό του ασθενούς στον οποίο ανήκει το συγκεκριμένο αρχείο. Με τον τρόπο αυτό, οι ασθενείς θα μπορούν να αποθηκεύουν τις εξετάσεις τους σε μορφή pdf και ο κάθε ασθενής θα μπορεί να καταχωρήσει απεριόριστο αριθμό εξετάσεων.

Πίνακας Ιατροί

Αντίστοιχα με τους ασθενείς, οι ιατροί αποθηκεύονται σε έναν μεμονωμένο πίνακα. Οι εγγραφές τους περιλαμβάνουν το Όνομα, το Επώνυμο, την Ειδικότητα, τη Διεύθυνση Ηλεκτρονικού Ταχυδρομείου, τον Κωδικό (ομοίως σε μορφή Hash), την Διεύθυνση Εργασίας και τέλος τον Αριθμό Τηλεφώνου. Η ειδικότητα αποθηκεύεται σε μορφή ακέραιου ενώ η πραγματική τιμή διατηρείται σε ξεχωριστό πίνακα για λόγους αποδοτικότερης χρήσης των διαθέσιμων πόρων. Η διατήρηση του ακέραιου που αντιστοιχεί στον κωδικό της ειδικότητας, εξασφαλίζει ότι ο κάθε ιατρός μπορεί να έχει μόνο μία ειδικότητα.

Πίνακας Ειδικότητες

Ο πίνακας περιλαμβάνει τις ειδικότητες των ιατρών. Κατά συνέπεια, κατά την εγγραφή του, ο κάθε ιατρός μπορεί να επιλέξει μόνο μία από τις διαθέσιμες τιμές.

Πίνακας Αντιστοίχισης Ιατρών σε Ασθενείς

Ο πίνακας περιέχει τις συσχετίσεις των ιατρών με τους ασθενείς. Με τον τρόπο αυτό ο κάθε ασθενής μπορεί να έχει περισσότερους από έναν ιατρούς, ανεξαρτήτου ειδικότητας και ο κάθε ιατρός μπορεί να παρακολουθεί περισσότερους από έναν ασθενή.

healthcare.fields	healthcare.doctors	healthcare.users2doctors	healthcare.user	healthcare.exams
id : int(11) name : varchar(20)	id : int(11) name : varchar(20) lname : varchar(20) # fieldId : int(11) email : varchar(50) password : varchar(255) address : varchar(50) phone : varchar(10)	id : int(11) # usersId : int(11) # doctorsId : int(11)	id : int(11) amka : varchar(11) name : varchar(20) lname : varchar(20) birth : date # height : decimal(3,2) # weight : decimal(5,2) email : varchar(50) mobile : varchar(10) password : varchar(256)	id : int(11) # userId : int(11) pdf_file : mediumblob

Εικόνα 32 - Η Βάση Δεδομένων

3.4 Ο κώδικας της Εφαρμογής

3.4.1 Παραμετροποίηση του Web Server

Η εφαρμογή θα αποθηκευτεί στο φάκελο healthcare του εξυπηρετητή και κατά συνέπεια η πλήρης διαδρομή του φακέλου θα είναι /var/www/html/healthcare. Στο φάκελο αυτό, θα δημιουργηθούν οι επιπλέον φάκελοι που θα φιλοξενήσουν τα αρχεία της εφαρμογής και η αρχική σελίδα index.php, η οποία θα αποτελέσει την πύλη εισόδου στην εφαρμογή. Για να επιτευχθεί αυτό, καθώς και να αποκρύπτεται η προβολή των καταλήξεων των αρχείων στη διεύθυνση, θα εφαρμοστεί η τεχνική της δρομολόγησης των διευθύνσεων. Συνεπώς, θα ενεργοποιηθεί το πρόσθετο a2enmod και θα δημιουργηθεί το αρχείο .htaccess που αποτρέπει τη χρήση μη επιτρεπτών αρχείων και ανακατευθύνει όλες τις κλήσεις προς διεκπεραίωση στο αρχείο index.php.

/var/www/html/healthcare/	
Name	Size
 public	
 application	
 index.php	1 KB
 .htaccess	1 KB

Εικόνα 33 - Η δομή των αρχείων της Εφαρμογής

3.4.2 Η Πύλη Εισόδου Index.php

Στόχος της σελίδας αυτής είναι να ενεργοποιήσει τα αντικείμενα που είναι απαραίτητα για την διεκπεραίωση των κλήσεων. Αρχικά, θα φορτώσει το αρχείο config.php, το οποίο ορίζει ότι πρέπει να εμφανίζονται τα διαγνωστικά μηνύματα της εφαρμογής, η διεύθυνση της εφαρμογής και τα στοιχεία για τη σύνδεση στη βάση δεδομένων. Ακολούθως, θα φορτώνει το αρχείο Application.php, το οποίο αποτελεί το δρομολογητή της εφαρμογής. Το αντικείμενο αυτό, διαβάζει τη διεύθυνση που εισάγει ο χρήστης (το URL) και εξάγει από αυτό, τα αρχεία που θα πρέπει να ενεργοποιηθούν. Σε περίπτωση που η διεύθυνση περιέχει κάτι το οποίο δεν υπάρχει, θα εμφανίζει την αρχική σελίδα μέσω του controller Home.php. Με τον τρόπο αυτό, αν ο χρήστης εισάγει στην διεύθυνση το «home» (ή οτιδήποτε άλλο), η εφαρμογή θα καλέσει το αντικείμενο home.php και θα εκτελέσει την προκαθορισμένη (default) συνάρτηση του index() προβάλλοντας την αρχική σελίδα στο χρήστη. Τέλος, ενεργοποιείται ο βασικός διαχειριστής (Controller) της εφαρμογής, από τον οποίο κληρονομούν τα διάφορα αντικείμενα. Ο Controller αυτός περιέχει τη σύνδεση με τη βάση δεδομένων, ενεργοποιεί το Session για την αποθήκευση δεδομένων μεταξύ των HTTP κλήσεων και φορτώνει το μοντέλο που θα του ζητηθεί. Συνεπώς, δημιουργείτε ένα αντικείμενο της κλάσης Application.

```
error_reporting(E_ALL);
ini_set("display_errors", 1);

define('URL', 'http://34.65.141.100/healthcare/');

define('DB_TYPE', 'mysql');
define('DB_HOST', '127.0.0.1');
define('DB_NAME', 'healthcare');
define('DB_USER', 'healthcare');
define('DB_PASS', 'HealthCare2021!');
```

Εικόνα 34 - config.php

```
require 'application/configs/config.php';
require 'application/libs/Application.php';
require 'application/libs/Controller.php';

$app = new Application();
```

Εικόνα 35 - index.php

3.4.3 Η Αρχική Σελίδα Home

Πληκτρολογώντας ο χρήστης τη διεύθυνση του Web Server, το σύστημα εμφανίζει την ιστοσελίδα της ακόλουθης εικόνας.

Εφαρμογή Ιατρικού Φακέλου

Παρακαλώ εισάγετε τους κωδικούς σας

Όνομα Χρήστη

Κωδικός Πρόσβασης

Είσοδος

Εγγραφή

Εικόνα 36 - Η αρχική σελίδα της εφαρμογής

Η σελίδα είναι προϊόν της κλάσης Home που βρίσκεται στη διαδρομή /healthcare/application/controllers. Η κλάση ελέγχει τη μεταβλητή role του Session και αν δεν βρει τιμή σημαίνει ότι κανείς δεν έχει περάσει τη διαδικασία πιστοποίησης, οπότε εμφανίζει τη σελίδα της παραπάνω εικόνας. Σε περίπτωση που κάποιος έχει ταυτοποιηθεί και για κάποιο λόγο βρέθηκε στη σελίδα αυτή, η μεταβλητή role περιέχει το ρόλο του (ιατρός ή ασθενής) και ανάλογα εμφανίζει το κατάλληλο περιεχόμενο.

Το αρχείο της σελίδας που θα εμφανίσει η κλάση μέσω της εντολής require, βρίσκεται στη διαδρομή application/views/login/index.php. Αποτελείται από δύο πεδία στα οποία εισάγει ο χρήστης τα στοιχεία του και ακολούθως ένα κουμπί 'Είσοδος' προκειμένου να εισαχθεί στην εφαρμογή. Στην περίπτωση που δεν διαθέτει κωδικούς, επιλέγει του κουμπί 'Εγγραφή'. Για τη μορφοποίηση του περιεχομένου, χρησιμοποιείται η γλώσσα προγραμματισμού CSS.

3.4.4 Εγγραφή Χρήστη

Η επιλογή του κουμπιού 'Εγγραφή', καλεί τη συνάρτηση register του αντικειμένου Login. Η συνάρτηση αυτή εμφανίζει στο χρήστη το αρχείο registerForm.php από τη διαδρομή application/register/. Στη φόρμα αυτή, ο χρήστης θα εισάγει τα στοιχεία του και θα επιλέγει 'Εγγραφή' ώστε να ολοκληρωθεί η διαδικασία. Ο διαχωρισμός των ρόλων μεταξύ ασθενή και ιατρού, γίνεται από την επιλογή του Radio Button της 'Ιδιότητας' μέσω JavaScript ώστε να μην εμπλέκεται ο Server εκ νέου. Το αρχείο που περιέχει το script, είναι στη διαδρομή public/jss/register.js και ενεργοποιείται με την αλλαγή του Radio Button. Κατά συνέπεια, όταν ο χρήστης προβεί σε κάποια αλλαγή, το script γράφει στη σελίδα τα δεδομένα που αφορούν την συγκεκριμένη επιλογή.

Εφαρμογή Ιατρικού Φακέλου

Εγγραφή Νέου Χρήστη

Παρακαλώ εισάγετε τα απαραίτητα στοιχεία

Ιδιότητα

Ασθενής Γιατρός

Email:

Εισάγετε email σας

Κωδικός Πρόσβασης:

Εισάγετε κωδικό πρόσβασης

Όνομα:

Εισάγετε το όνομα σας

Επώνυμο:

Εισάγετε το επώνυμο σας

ΑΜΚΑ:

Εισάγετε το ΑΜΚΑ σας

Κιν. Τηλέφωνο:

Εισάγετε τον αριθμό του κινητού σας

Ημερομηνία Γέννησης:

ηη/μμ/εεεε



Ύψος:

Εισάγετε το ύψος σας

Βάρος:

Εισάγετε το βάρος σας

Εγγραφή

Εικόνα 37 - Φόρμα Εγγραφής Χρήστη

Εφαρμογή Ιατρικού Φακέλου

Εγγραφή Νέου Γιατρού

Παρακαλώ εισάγετε τα απαραίτητα στοιχεία

Ιδιότητα

Ασθενής Γιατρός

Email:

Εισάγετε email σας

Κωδικός Πρόσβασης:

Εισάγετε κωδικό πρόσβασης

Όνομα:

Εισάγετε το όνομα σας

Επώνυμο:

Εισάγετε το επώνυμο σας

Πεδίο:

Επιλέξτε Τιμή

Κιν. Τηλέφωνο:

Εισάγετε τον αριθμό του κινητού σας

Διεύθυνση:

Εισάγετε οδό, αριθμό, περιοχή

Εγγραφή

Εικόνα 38 - Φόρμα Εγγραφής Ιατρού

Όταν ο χρήστης συμπληρώσει όλα τα απαιτούμενα πεδία και επιλέξει το κουμπί 'Εγγραφή', η φόρμα μέσω της μεθόδου POST θα καλέσει τη συνάρτηση subRegister του αντικειμένου Login. Η συνάρτηση θα ελέγξει με τη σειρά της τη μεταβλητή role που της απέστειλε η φόρμα και θα αποφασίσει εάν πρόκειται για ασθενή ή ιατρό. Κατόπιν, θα φορτώσει το αρμόδιο μοντέλο 'userModel' και θα εκτελέσει την κατάλληλη συνάρτηση, παρέχοντάς της τα δεδομένα που απέστειλε η φόρμα. Έπειτα, θα ενημερώσει τον χρήστη για την επιτυχή εγγραφή του και θα του διαθέσει έναν σύνδεσμο για να επιστρέψει στη σελίδα ταυτοποίησης.

Το αντικείμενο UserModel από τη πλευρά του, θα πάρει τα δεδομένα της φόρμας ως ορίσματα της συνάρτησης regPerson, θα αφαιρέσει μέσω της συνάρτησης strip_tags τους μη επιτρεπτούς χαρακτήρες και θα τα εισάγει στον κατάλληλο πίνακα της βάσης δεδομένων. Στις τιμές των δεκαδικών ψηφίων, θα αντικαταστήσει το κόμμα με τελεία ενώ στη περίπτωση του κωδικού θα αποθηκεύσει το hash του.

3.4.5 Είσοδος Χρήστη

Μόλις ο χρήστης εισάγει τους κωδικούς του και επιλέξει 'Είσοδος', υποβάλλεται η φόρμα και καλείται η συνάρτηση index του αντικειμένου Login. Η συνάρτηση τώρα, φορτώνει το αντικείμενο userModel και καλεί τη συνάρτηση του login, δίνοντάς του ως ορίσματα το email και τον κωδικό που πληκτρολόγησε νωρίτερα ο χρήστης.

```
public function index() {  
    $session = $this->startSession();  
    $session->init();  
  
    $usersModel = $this->loadModel('userModel');  
    $person = $usersModel->login($_POST["email"], $_POST["psw"]);  
}
```

Εικόνα 39 - Κλήση της συνάρτησης login του μοντέλου userModel

Το μοντέλο από την πλευρά του, αρχικά ζητά από τη βάση να του επιστρέψει εάν υπάρχει, εγγραφή που να αντιστοιχεί στο email που έλαβε. Εάν βρει υπάρχουσα εγγραφή, ελέγχει το κατά πόσον ταυτίζονται το password που έδωσε ο χρήστης με το password της εγγραφής. Αν υπάρξει επιτυχής αντιστοίχιση, επιστρέφει το αντίστοιχο αντικείμενο (ασθενής ή ιατρός) ενώ σε διαφορετική περίπτωση επιστρέφει 'false'.

Στη συνέχεια το αντικείμενο Login, ελέγχει την τιμή που επέστρεψε η συνάρτηση. Στη περίπτωση που το αντικείμενο είναι τύπου ιατρός ή ασθενής, αποθηκεύει στο Session τις αντίστοιχες τιμές για να είναι διαθέσιμες στη συνέχεια και καλεί τη συνάρτηση showUser του αντικειμένου Home. Σε διαφορετική περίπτωση ενημερώνει το χρήστη για λάθος.

Μετά την επιτυχή είσοδο του χρήστη στο σύστημα, εμφανίζεται η φόρμα ενημέρωσης στοιχείων ώστε να μπορεί να επικαιροποιήσει τα στοιχεία του.

```
if(is_a($person, 'User')){ //It's a patient

    $session->put('role', 'patient');
    $session->put('id', $person->getId());
    $session->put('amka', $person->getAmka());
    $session->put('name', $person->getName());
    $session->put('lname', $person->getLname());
    $session->put('birth', $person->getBirth());
    $session->put('height', $person->getHeight());
    $session->put('weight', $person->getWeight());
    $session->put('email', $person->getEmail());
    $session->put('mobile', $person->getMobile());

    header("Location: ".URL."Home/showUser");
}
elseif(is_a($person, 'Doctor')){ //It's a doctor
```

Εικόνα 40 - Αποθήκευση των τιμών του αντικείμενου στο Session

```
public function showUser()
{
    $session = $this->startSession();
    $session->init();

    $role = $session->get('role');
    $id = $session->get('id');
    $amka = $session->get('amka');
    $name = $session->get('name');
    $lname = $session->get('lname');
    $birth = $session->get('birth');
    $height = $session->get('height');
    $weight = $session->get('weight');
    $email = $session->get('email');
    $mobile = $session->get('mobile');

    require 'application/views/templates/header.php';
    require 'application/views/person/showData.php';
    require 'application/views/templates/footer.php';
}
```

Εικόνα 41 - Η showUser της Home

3.4.6 Ενημέρωση Στοιχείων Χρήστη

Οι τιμές του Session εγγράφονται στα input type αντικείμενα της φόρμας του αρχείου showData, ώστε να είναι διαθέσιμα για επαναποστολή στον εξυπηρετητή, προκειμένου αυτός να ενημερώσει τη βάση δεδομένων με τα νέα στοιχεία που εισήγαγε ο χρήστης.

```
<div id = "formContent">
  <div class="formItem">
    <label for="email" class="formLabel" ><b>Email:</b></label>
    <input type="text" class="formInput" placeholder="Εισάγετε email σας" name="email" value="<?php echo $email ?>" required>
  </div>
  <div class="formItem">
    <label for="psw" class="formLabel"><b>Κωδικός Πρόσβασης:</b></label>
    <input type="password" class="formInput" placeholder="Εισάγετε νέο κωδικό πρόσβασης" name="psw" required>
  </div>

  <div class="formItem">
    <label for="uname" class="formLabel" ><b>Όνομα: </b></label>
    <input type="text" class="formInput" placeholder="Εισάγετε το όνομα σας" name="uname" value="<?php echo $name ?>" required>
  </div>
</div>
```

Εικόνα 42 - Οι τιμές του Session γράφονται στα αντίστοιχα inputs

Με τον τρόπο αυτό, ο χρήστης έχει τη δυνατότητα να ενημερώσει τα στοιχεία του.

Εφαρμογή Ιατρικού Φακέλου

Καλώς όρισε Εμμανουήλ Κωνσταντίνου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να ενημερώσετε τα στοιχεία σας. Συμπληρώστε τις νέες τιμές και πατήστε 'Υποβολή'. Για τη διαχείριση των ιατρών σας επιλέξτε 'Ιατροί' ενώ για τις εξετάσεις σας επιλέξτε 'Εξετάσεις'.

Email:

Κωδικός Πρόσβασης:

Όνομα:

Επώνυμο:

ΑΜΚΑ:

Κιν. Τηλέφωνο:

Ημερομηνία Γέννησης (σε μορφή Έτος - Μήνας - Μέρα):

Ύψος:

Βάρος:

Εικόνα 43 - Φόρμα ενημέρωσης των στοιχείων του χρήστη

Επιλέγοντας ενημέρωση, η φόρμα καλεί τη συνάρτηση `updateUser` του αντικειμένου `Login` αποστέλλοντάς της ό,τι έχει πληκτρολογήσει ο χρήστης. Όλα τα πεδία είναι απαραίτητα να συμπληρωθούν προτού η φόρμα υποβληθεί. Η `updateUser`, μέσω της αντίστοιχης συνάρτησης του αντικειμένου `userModel`, ενημερώνει την εγγραφή του χρήστη στη βάση, εισάγει τα νέα πεδία στο `Session` και εμφανίζει εκ νέου τη φόρμα ενημερωμένη με τα νέα στοιχεία.

```

public function updateUser(){
    $session = $this->startSession();
    $session->init();

    $usersModel = $this->loadModel('userModel');
    $usersModel->updateUser($_POST["id"],$_POST["email"],$_POST["paw"],$_POST["uname"],$_POST["lname"],$_POST["anks"],$_POST["mobile"],$_POST["birth"],$_POST["height"],$_POST["weight"]);

    $session->put('ankle', $_POST["ankle"]);
    $session->put('name', $_POST["uname"]);
    $session->put('lname', $_POST["lname"]);
    $session->put('birth', $_POST["birth"]);
    $session->put('height', $_POST["height"]);
    $session->put('weight', $_POST["weight"]);
    $session->put('email', $_POST["email"]);
    $session->put('mobile', $_POST["mobile"]);

    header("Location:".URL."/Home/showUser");
}

```

Εικόνα 44 - Ενημέρωση στοιχείων του χρήστη

3.4.7 Επιλογή Ιατρών

Εφαρμογή Ιατρικού Φακέλου

Καλώς όρισες Ζωή Γεωργίου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να ενημερώσετε τους ιατρούς σας. Συμπληρώστε τις νέες τιμές και πατήστε 'Υποβολή'.

Δέν εχετε επιλέξει Ιατρούς. Παρακαλώ Επιλέξτε:

Επιλέξτε Πεδίο:

Ενημέρωση Στοιχείων

Εξετάσεις

Εικόνα 45 - Φόρμα επιλογής Ιατρών

Επιλέγοντας το κουμπί 'Ιατροί' από την αρχική σελίδα του χρήστη, ενεργοποιείται το αντικείμενο PatientControler και εκτελείται η συνάρτηση doctors. Αυτή διαβάζει από το Session τις τιμές του χρήστη, ζητά από το μοντέλο να επιστρέψει τους ιατρούς τους οποίους ο χρήστης έχει επιλέξει και κατόπιν εμφανίζει τη σελίδα showDoctors.php.

Το μοντέλο για να επιλέξει τους ιατρούς, πραγματοποιεί ένα Join στους πίνακες user2doctors και doctors για να πάρει τα στοιχεία των ιατρών που ανήκουν στον συγκεκριμένο ασθενή και παράλληλα

ένα ακόμα Join στους πίνακες που περιλαμβάνουν τους γιατρούς και τις ειδικότητες, για να πάρει την ειδικότητα του εκάστοτε ιατρού. Στη περίπτωση που υπάρχουν διαθέσιμες εγγραφές, αυτές επιστρέφονται στη μεταβλητή msg, ενώ σε διαφορετική περίπτωση επιστρέφει false. Τη μεταβλητή αυτή αργότερα, ελέγχει η σελίδα showDoctors και αν δεν είναι false, γράφει τις εγγραφές που περιέχει. Σε διαφορετική περίπτωση εκτυπώνει το μήνυμα της προηγούμενης εικόνας αφού ο χρήστης δεν έχει ακόμα ορίσει ιατρούς.

Τέλος, μέσω των κουμπιών 'Ενημέρωση Στοιχείων' και 'Εξετάσεις' μεταβαίνει στις αντίστοιχες σελίδες.

Εφαρμογή Ιατρικού Φακέλου

Καλώς όρισε Ζωή Γεωργίου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να ενημερώσετε τους ιατρούς σας. Συμπληρώστε τις νέες τιμές και πατήστε 'Υποβολή'.

Δεν έχετε επιλέξει Ιατρούς. Παρακαλώ Επιλέξτε:

Επιλέξτε Πεδίο:

Επιλέξτε Τιμή
Επιλέξτε Τιμή
Καρδιολογία
Παθολογία - Γενική Ιατρική
Παιδιατρική
Δερματολογία
Χειρουργική
Γυναικολογία
Νευρολογία
Ψυχιατρική

Εξετάσεις

Εικόνα 46 - Επιλογή Ιατρών από τη λίστα

Μόλις ο χρήστης επιλέξει ένα ιατρό από το drop down list, η javascript πραγματοποιεί μια ασύγχρονη κλήση μέσω ajax στη συνάρτηση getDoctors του αντικειμένου Login, αποστέλλοντας της το id του κάθε χρήστη και αντλεί από αυτήν τους ιατρούς που ο χρήστης έχει ήδη ορίσει. Αμέσως μετά, μόλις λάβει τις απαντήσεις σε μορφή JSON, αντλεί τα στοιχεία των ιατρών και τα γράφει στο συγκεκριμένο

σημείο της σελίδας σε μορφή φόρμας (ξεχωριστής για κάθε εγγραφή, για να διακρίνονται τα ξεχωριστά id των ιατρών), ώστε να μπορούν να αποσταλούν στη συνέχεια στον εξυπηρετητή για να αποθηκευτούν στη βάση δεδομένων. Τέλος, μετατρέπει το div αλλά και τον πίνακα που περιέχει τις συγκεκριμένες εγγραφές σε ορατό, για να μπορεί να τις δει ο χρήστης.

Εφαρμογή Ιατρικού Φακέλου

Καλώς όρισε Ζωή Γεωργίου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να ενημερώσετε τους ιατρούς σας. Συμπληρώστε τις νέες τιμές και πατήστε 'Υποβολή'.

Δεν έχετε επιλέξει Ιατρούς. Παρακαλώ Επιλέξτε:

Επιλέξτε Πεδίο:

Νευρολογία

Λίστα διαθέσιμων ιατρών στη κατηγορία:

Όνομα	Επώνυμο	Διεύθυνση	Τηλέφωνο	
Αθανάσιος	Παπαδόπουλος	Αντιγόνης 24, Καλλιθέα	6978452547	Ενημέρωση
Ευάγγελος	Κατσαρός	Καλαμών 34, Τρίκαλα	6954823540	Ενημέρωση

Ενημέρωση Στοιχείων

Εξετάσεις

Εικόνα 47 - Προβολή διαθέσιμων για επιλογή Ιατρών ειδικότητας Νευρολογίας

Το περιεχόμενο του πίνακα δημιουργείται εκ νέου κάθε φορά που ο χρήστης επιλέγει μια διαφορετική ειδικότητα. Πιο συγκεκριμένα, όταν αλλάξει η ειδικότητα, αφαιρούνται οι προηγούμενες εγγραφές και προστίθενται οι νέες.

Εφαρμογή Ιατρικού Φακέλου

Καλώς όριστες Ζωή Γεωργίου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να ενημερώσετε τους ιατρούς σας. Συμπληρώστε τις νέες τιμές και πατήστε 'Υποβολή'.

Δεν έχετε επιλέξει Ιατρούς. Παρακαλώ Επιλέξτε:

Επιλέξτε Πεδίο:

Δερματολογία

Λίστα διαθέσιμων ιατρών στη κατηγορία:

Όνομα	Επώνυμο	Διεύθυνση	Τηλέφωνο	
Σοφία	Παρασκευοπούλου	Ξάνθου 23, Ιωάννινα	6940254358	Ενημέρωση

Ενημέρωση Στοιχείων

Εξετάσεις

Εικόνα 48 - Νέες εγγραφές του πίνακα

Από τη φόρμα αυτή ο χρήστης μπορεί να επιλέξει τον ιατρό που τον εξυπηρετεί και ακολούθως ο ιατρός αυτός, να προστεθεί στη λίστα του. Το πάτημα του κουμπιού 'Ενημέρωση' καλεί τη συνάρτηση `updateDoctor` του αντικειμένου `PatientControler`, αποστέλλοντάς του το `id` του χρήστη και το `id` του ιατρού που επέλεξε, ώστε να ξέρει το σύστημα ποιο χρήστη και ποιο ιατρό να προσθέσει στο πίνακα `users2doctors`. Μετά το πέρας της διαδικασίας, καλείται εκ νέου η `doctors` του `PatientControler`, ώστε η σελίδα να αποστέλλεται εκ νέου στο χρήστη ενημερωμένη.

Εφαρμογή Ιατρικού Φακέλου

Καλώς όριστες Ζωή Γεωργίου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να ενημερώσετε τους ιατρούς σας. Συμπληρώστε τις νέες τιμές και πατήστε 'Υποβολή'.

Λίστα επιλεγμένων ιατρών:

Όνομα	Επώνυμο	Email	Διεύθυνση	Τηλέφωνο	Ειδικότητα	
Σοφία	Παρασκευοπούλου	sofia@skincare.com	Ξάνθου 23, Ιωάννινα	6940254358	Δερματολογία	Διαγραφή

Επιλέξτε Πεδίο:

Επιλέξτε Τιμή

Ενημέρωση Στοιχείων

Εξετάσεις

Εικόνα 49 - Η ίδια σελίδα μετά την επιλογή του κουμπιού Ενημέρωση της πρώτης εγγραφής του πίνακα

Τώρα η μεταβλητή msg δεν είναι false, περιέχει τιμές και η σελίδα εμφανίζει τις εγγραφές. Οι εγγραφές γίνονται πάλι σε μορφή φόρμας ώστε ο χρήστης να μπορεί, μέσω του κουμπιού 'Διαγραφή' να διαγράψει το συγκεκριμένο ιατρό από τη λίστα του. Παράλληλα, μπορεί να συνεχίσει να προσθέτει ιατρούς μέσω του drop down list.

Εφαρμογή Ιατρικού Φακέλου

Καλώς όριστες Ζωή Γεωργίου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να ενημερώσετε τους ιατρούς σας. Συμπληρώστε τις νέες τιμές και πατήστε 'Υποβολή'.

Λίστα επιλεγμένων ιατρών:

Όνομα	Επώνυμο	Email	Διεύθυνση	Τηλέφωνο	Ειδικότητα	
Αλεξία	Κωνσταντίνου	konstantinou alex@doctor.com	Αριστοτέλους 19, Θεσσαλονίκη	6942530941	Καρδιολογία	Διαγραφή
Μαρία	Αλεξίου	malexiu@hotmail.com	Κλαυθμώνος 33, Αθήνα	6948210352	Παθολογία - Γενική Παθολογία	Διαγραφή
Κατερίνα	Γεωργίου	akg_georgiou@yahoo.com	Πραξιτέλους 42, Παγκράτι	6958425685	Παιδιατρική	Διαγραφή
Αθανάσιος	Παπαδόπουλος	thanasias@bing.com	Αντιγόνης 24, Καλλιθέα	6978452547	Νευρολογία	Διαγραφή
Σοφία	Κυπραίου	sof_kyp@docs_anytime.com	Δρυάδων 23, Τρίκαλα	6938425682	Ψυχιατρική	Διαγραφή

Επιλέξτε Πεδίο:

Επιλέξτε Τιμή

Ενημέρωση Στοιχείων

Εξετάσεις

Εικόνα 50 - Προσθήκη Ιατρών

Εφαρμογή Ιατρικού Φακέλου

Καλώς όρισες Ζωή Γεωργίου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να ενημερώσετε τους ιατρούς σας. Συμπληρώστε τις νέες τιμές και πατήστε 'Υποβολή'.

Λίστα επιλεγμένων ιατρών:

Όνομα	Επώνυμο	Email	Διεύθυνση	Τηλέφωνο	Ειδικότητα	
Κατερίνα	Γεωργίου	akg_georgiou@yahoo.com	Πραξιτέλους 42, Παγκράτι	6958425685	Παιδιατρική	Διαγραφή
Αθανάσιος	Παπαδόπουλος	thanasis@bing.com	Αντιγόνης 24, Καλλιθέα	6978452547	Νευρολογία	Διαγραφή

Επιλέξτε Πεδίο:

Επιλέξτε Τιμή

Ενημέρωση Στοιχείων

Εξετάσεις

Εικόνα 51 - Διαγραφή Ιατρών και προσθήκη νέων

Στη περίπτωση που όλοι οι ιατροί αφαιρεθούν, το σύστημα εμφανίζει το μήνυμα της αρχικής εικόνας ενημερώνοντας το χρήστη ότι δεν έχει ορίσει ιατρούς.

3.4.8 Εξετάσεις

Εφαρμογή Ιατρικού Φακέλου

Παρακαλώ εισάγετε τις εξετάσεις σας σε μοφή PDF

Περιγραφή

Εισάγετε μία περιγραφή

Επιλέξτε αρχείο. Μέγιστο μέγεθος: 2MB

Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχείο.

Υποβολή Αρχείου

Ενημέρωση Στοιχείων

Ιατροί

Εικόνα 52 - Επιλέγοντας το κουμπί 'Εξετάσεις' ο χρήστης μεταβαίνει στην ομώνυμη φόρμα

Η φόρμα αυτή προβάλλεται από τη συνάρτηση exams, επίσης του αντικειμένου PatientControler. Η υποβολή της, ενεργοποιεί τη συνάρτηση updateExams του ίδιου αντικειμένου, το οποίο αφού κάνει τους απαραίτητους ελέγχους και τη μεταφορά του αρχείου στον εξυπηρετητή, το εισάγει στη βάση σε μορφή BLOB μέσω της συνάρτησης insertExams του μοντέλου userModel.

`SELECT * FROM `exams` ORDER BY `userId` ASC`

Δημιουργία προφίλ [Επεξεργασία εσωτερικά] [Επε

Εμφάνιση όλων | Αριθμός εγγραφών: Φιλτράρισμα εγγραφών: | | | | | id | userId | 1 | details | pdf_file |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | | | 13 | | 0 | Σκολίωση | [BLOB - 306,6 KB] |
| | | | | 17 | | 0 | Ορμονικές | [BLOB - 128,0 KB] |
| | | | | 9 | | 3 | Εξετάσεις Σακχάρου | [BLOB - 70,9 KB] |
| | | | | 10 | | 8 | Φωτογραφίες Σκολίωσης | [BLOB - 306,6 KB] |
| | | | | 8 | | 13 | MRI | [BLOB - 469,1 KB] |

Εικόνα 53 - Εγγραφές του πίνακα exams μετά την εισαγωγή των εξετάσεων του χρήστη

3.4.9 Εγγραφή Ιατρού

Όπως αναφέρθηκε στη παράγραφο 2.4.4, η εγγραφή του ιατρού στο σύστημα γίνεται από την ίδια φόρμα με αυτή που διεκπεραιώνει την εγγραφή του χρήστη. Πιο συγκεκριμένα, η επιλογή του κουμπιού ‘Εγγραφή’ ενεργοποιεί τη συνάρτηση register του αντικειμένου Login, το οποίο εμφανίζει στο χρήστη τη φόρμα registerForm.php. Από τη φόρμα αυτή, ο χρήστης αρχικά θα επιλέξει το ρόλο του, που είναι είτε ασθενής είτε ιατρός. Ανάλογα με την επιλογή, το περιεχόμενο της φόρμας προσαρμόζεται, καθώς το αρχείο register.js περιέχει ένα Javascript το οποίο ανιχνεύει την αλλαγή του συγκεκριμένου Radio Button και ενημερώνει τα περιεχόμενα της φόρμας. Με τον τρόπο αυτό, επιτυγχάνεται μια αποδοτικότερη διαχείριση των πόρων, αφού υπάρχει ένα αρχείο και για τους δύο ρόλους και ο server δεν εμπλέκεται κατά την εναλλαγή.

```

1 function getValue(radio) {
2     if(radio.value=='person'){
3         //alert("Άτομο");
4         document.getElementById("title").innerHTML = "Εγγραφή Νέου Χρήστη";
5         document.getElementById("formContent").innerHTML = `
6
7             <div class="formItem">
8                 <label for="email" class="formLabel" ><b>Email:</b></label>
9                 <input type="text" class="formInput" placeholder="Εισάγετε email σας" name="email" >
10            </div>
11            <div class="formItem">
12                <label for="psw" class="formLabel"><b>Κωδικός Πρόσβασης:</b></label>
13                <input type="password" class="formInput" placeholder="Εισάγετε κωδικό πρόσβασης" name="psw" >
14            </div>
15

```

Εικόνα 54 - Τμήμα κώδικα Javascript. Στη περίπτωση που η τιμή του radio είναι person τα αντικείμενα title και formContent της φόρμας παίρνουν τις αντίστοιχες τιμές

```

51     }
52     else {
53         //alert("Γιατρός");
54         document.getElementById("title").innerHTML = "Εγγραφή Νέου Γιατρού";
55         document.getElementById("formContent").innerHTML = `
56
57             <div class="formItem">
58                 <label for="email" class="formLabel"><b>Email:</b></label>
59                 <input type="text" class="formInput" placeholder="Εισάγετε email σας" name="email" >
60            </div>
61            <div class="formItem">
62                <label for="psw" class="formLabel"><b>Κωδικός Πρόσβασης:</b></label>
63                <input type="password" class="formInput" placeholder="Εισάγετε κωδικό πρόσβασης" name="psw" >
64            </div>
65

```

Εικόνα 55 - Στη περίπτωση που η τιμή του radio δεν είναι person παίρνουν τις τιμές του ιατρού

Αφού ο χρήστης ολοκληρώσει την εισαγωγή των στοιχείων, επιλέγει το κουμπί 'Εγγραφή' και οι τιμές, μέσω της μεθόδου POST του HTTP μεταφέρονται στο Server, ο οποίος τα μεταφέρει στη βάση.

3.4.10 Είσοδος Ιατρού

Η διαδικασία εισόδου του ιατρού στο σύστημα είναι ίδια με αυτή του χρήστη. Συγκεκριμένα, ο ιατρός εισάγει τους κωδικούς του στο σύστημα και επιλέγει 'Είσοδος'. Τα στοιχεία τα λαμβάνει η συνάρτηση index του αντικειμένου Login, το οποίο μέσω του μοντέλου επικοινωνεί με την βάση δεδομένων ώστε να γίνει έλεγχος αν υπάρχουν οι αντίστοιχες εγγραφές, αρχικά αναζητώντας το email στο πίνακα των χρηστών. Εάν δεν βρεθεί εγγραφή, αναζητά και στον πίνακα των ιατρών. Η παραδοχή είναι ότι δεν είναι δυνατόν να υπάρχουν δύο άτομα με το ίδιο ακριβώς email.

Αφού ολοκληρωθεί η ταυτοποίηση του ιατρού, η index του Login αποθηκεύει στο Session τα δεδομένα του ιατρού και ύστερα εμφανίζει τη φόρμα ενημέρωσης στοιχείων ιατρού, μέσω της συνάρτησης showDoctor του αντικειμένου Home.

```
elseif(is_a($person, 'Doctor')){ //It's a doctor

    $session->put('role', 'doctor');
    $session->put('id', $person->getId());
    $session->put('name', $person->getName());
    $session->put('lname', $person->getLName());
    $session->put('field', $person->getFieldId());
    $session->put('phone', $person->getPhone());
    $session->put('address', $person->getAddress());
    $session->put('email', $person->getEmail());

    header("Location:".URL."Home/showDoctor");
```

Εικόνα 56 - Τμήμα κώδικα συνάρτησης index του αντικειμένου Login. Το μοντέλο έχει δημιουργήσει ένα αντικείμενο Person του οποίου οι τιμές αποθηκεύονται στο Session για μελλοντική χρήση

3.4.11 Ενημέρωση στοιχείων Ιατρού

Όπως και στη περίπτωση του χρήστη, έτσι και στην περίπτωση του ιατρού υπάρχει η δυνατότητα να ενημερωθεί το σύνολο των στοιχείων του ιατρού μέσω της φόρμας ενημέρωσης στοιχείων. Η φόρμα αυτή είναι η πρώτη σε σειρά που εμφανίζεται μετά την επιτυχή είσοδο του ιατρού στο σύστημα.

Εφαρμογή Ιατρικού Φακέλου

Καλώς όρισεσ Αλεξία Κωνσταντίνου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να ενημερώσετε τα στοιχεία σας. Συμπληρώστε τις νέες τιμές και πατήστε 'Υποβολή'.

Email:

konstantinou alexi@doctor.com

Κωδικός Πρόσβασης:

Εισάγετε νέο κωδικό πρόσβασης

Όνομα:

Αλεξία

Επώνυμο:

Κωνσταντίνου

Πεδίο:

Επιλέξτε Τιμή

Κιν. Τηλέφωνο:

6942530941

Διεύθυνση:

Αριστοτέλους 19, Θεσσαλονίκη

Ενημέρωση

Εξετάσεις

Εικόνα 57 - Ενημέρωση στοιχείων ιατρού

Ο χρήστησ εισάγει τις νέες τιμές και ακολούθως επιλέγει 'Ενημέρωση', ώστε αυτές να ενημερώσουν τις ήδη υπάρχουσες στο DataBase Server.

3.4.12 Προβολή Εξετάσεων Ασθενούς

Η λειτουργικότητα αυτή είναι προσβάσιμη μέσω του κουμπιού 'Εξετάσεις'. Πιο συγκεκριμένα, το κουμπί αυτό, καλεί τη συνάρτηση exams του αντικειμένου DocController.

```
public function exams(){
    $session = $this->startSession();
    $session->init();

    $docsId = $session->get('id');
    $name = $session->get('name');
    $lname = $session->get('lname');

    $usersModel = $this->loadModel('userModel');
    $msg = $usersModel->getDoctorsPatients($docsId);

    require 'application/views/templates/header.php';
    require 'application/views/doctor/showPatients.php';
    require 'application/views/templates/footer.php';
}
```

Εικόνα 58 - Η συνάρτηση exams του αντικειμένου DocController

Η συνάρτηση ανακτά από το Session τις τιμές id, name και lname του ιατρού, ώστε να τις γράψει στο μήνυμα καλωσορίσματος και μέσω της συνάρτησης getDoctorsPatients του μοντέλου, λαμβάνει τους ασθενείς του συγκεκριμένου ιατρού. Στη συνέχεια, η εφαρμογή εμφανίζει τη σελίδα showPatients η οποία θα παρουσιάσει στον ιατρό τις συγκεκριμένες πληροφορίες μέσω της μεταβλητής msg.

```
public function getDoctorsPatients($id){
    $sql = "select user.id, user.amka, user.name, user.lname, user.email, user.mobile from user
    inner join users2doctors on user.id = users2doctors.usersid where users2doctors.doctorsId=".$id;
    $query = $this->db->prepare($sql);
    $result = $query->execute();
    $num = $query->rowCount();
    if($num>0){
        $rows = Array();

        while($row = $query->fetch(PDO::FETCH_ASSOC)){
            array_push($rows,$row);
        }
        return $rows;
    }
    else
        return false;
}
```

Εικόνα 59 - Η συνάρτηση getDoctorsPatients που παίρνει ως όρισμα το id του ιατρού και μέσω ενός join των πινάκων users και users2doctors επιστρέφει τους χρήστες που έχουν ορίσει το συγκεκριμένο ιατρό

Η showPatiets για κάθε εγγραφή του πίνακα, ο οποίος είναι αποθηκευμένος στη μεταβλητή msg και αντιστοιχεί σε ένα ασθενή, θα δημιουργήσει μια εγγραφή στο πίνακα προβολής ασθενών. Στη περίπτωση που ο ιατρός δεν έχει επιλεγεί από κάποιον ασθενή, η σελίδα εμφανίζει το αντίστοιχο μήνυμα.

```

<?php
if($msg) {
    echo '<p>Ποια Ασθενών:</p>';
    echo '<table border="1" align="center" >';
    echo '<thead><tr><th>ΑΜΒΟ</th><th>Όνομα</th><th>Επώνυμο</th><th>Email</th><th>Τηλέφωνο</th></tr></thead>';
    echo '<tbody>';
    $patientId = '';
    foreach ($msg as $patient){
        echo '<tr>';
        foreach ($patient as $key => $record){
            if($key=='id'){
                $patientId = $record;
            }
            else
                echo '<td>'.$record.'</td>';
        }

        echo '<td><div class="formItem"><button class="formBtn" onclick="showPatient('.$docsId.', '.$patientId.')"> Προβολή Στοιχείων </button></div></td>';
        echo '</tr>';
    }
    echo '</tbody>';
    echo '</table>';
}
else
    echo '<p>Δεν έχει οριστεί απο ασθενής</p>';
}
}

```

Εικόνα 60 - Προβολή στοιχείων ασθενούς

	id	name	lname	fieldId	email	password
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	1	Μαρία	Αλεξίου	2	malexiou@hotmail.com	\$2y\$10\$82RL4haab4TSYRKtFWQ4Aevbj/EHPmi
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	2	Κατερίνα	Γεωργίου	3	akg_georgiou@yahoo.com	\$2y\$10\$0sFzdoKYb9jJDspHULIXEe/UI3v2cUBWf
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	3	Αθανάσιος	Παπαδόπουλος	7	thanasis@bing.com	\$2y\$10\$WSPt1byixkmWab0Y.Oi6YOHYPeNNfpc
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	4	Σοφία	Κυπραίου	8	sof_kyp@docs_anytime.com	\$2y\$10\$XYjNiL0Jyye50U6brEzZ5uhLHKOm3e0c
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	5	Αλεξία	Κωνσταντίνου	1	konstantinou_alex@doctor.com	\$2y\$10\$c2cbKbCn7VccSJ2/iS8N/.FpxsMLi9NJdJ

Εικόνα 61 - Τμήμα του πίνακα ιατρών. Ο ιατρός Αλεξία Κωνσταντίνου έχει λάβει την id τιμή 5

			id	usersId	doctorsId				
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	1	3	1
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	49	8	2
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	3	13	3
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	64	22	3
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	4	13	4
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	65	13	5
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	66	8	5
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	63	22	9
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	52	20	10
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	50	8	11
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	51	18	11

Εικόνα 62 - Τμήμα του πίνακα users2doctors. Ο ιατρός με id 5 έχει οριστεί από τους ασθενής με id 13 και 8

				id	amka	name	lname	birth	height	weight	email			
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	3	15059811286	Ιωάννης	Τσιτσίκας	1998-05-15	1.92	92.00	itsitsikas@ote.gr
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	8	05020565845	Πέτρος	Αγγέλου	2008-05-02	1.54	51.90	petran@yahoo.com
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	13	05207264852	Κωνσταντίνος	Κωστόπουλος	1972-05-20	1.95	124.65	kostas@gmail.com
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	16	25107900842	Ελένη	Αντωνίου	1979-10-25	1.75	54.00	eantoniou@windows

Εικόνα 63 - Τμήμα του πίνακα users. Διακρίνονται οι χρήστες με id 13 και 8.

Εφαρμογή Ιατρικού Φακέλου

Καλώς όρισεσ Αλεξία Κωνσταντίνου

Ενημέρωση Στοιχείων

Μέσω της φόρμασ αυτήσ έχετε τη δυνατότητα να χειριστείτε τουσ ασθενείσ σας.

Λίστα Ασθενών:

ΑΜΚΑ	Όνομα	Επώνυμο	Email	Τηλέφωνο	
05207264852	Κωνσταντίνος	Κωστόπουλος	kostas@gmail.com	6974821535	Προβολή Στοιχείων
05020565845	Πέτροσ	Αγγέλου	petran@yahoo.com	6954824658	Προβολή Στοιχείων

Ενημέρωση Στοιχείων

Εικόνα 64 - Προβολή ασθενών ιατρού

Προκειμένου ο ιατρός να δει τα στοιχεία του κάθε ασθενούς ξεχωριστά, επιλέγει το κουμπί 'Προβολή Στοιχείων' του συγκεκριμένου ασθενούς. Το κουμπί ενεργοποιεί τη συνάρτηση showPatient του αρχείου showPatients το οποίο περιέχει το Javascript, το οποίο αναλαμβάνει να εμφανίσει τα στοιχεία του ασθενούς.

Το συγκεκριμένο αρχείο, όπως και στη περίπτωση του ασθενούς, επικοινωνεί με τον Server και συγκεκριμένα με τη συνάρτηση getPatients του αντικειμένου DocController. Πιο συγκεκριμένα, πραγματοποιεί μια ασύγχρονη κλήση μέσω Ajax και αποστέλλει το αναγνωριστικό id του χρήστη του οποίου το κουμπί επέλεξε ο ιατρός.

```

function showPatient(docsId,userId) {
  //alert('DoctorsId: '+docsId+', UsersId :'+userId);

  $.ajax({
    url: "http://192.168.7.65/healthcare/DocControler/getPatients",
    method: "POST",
    data: {userId:userId},
    dataType: "JSON",
    success: function(data)
    {

```

Εικόνα 65 - Ασύγχρονη κλήση για τη λήψη των στοιχείων του ασθενούς

Η συνάρτηση λαμβάνει τις τιμές και ζητά από το μοντέλο, μέσω της συνάρτησης `getPatientDetails`, να της επιστρέψει τις λεπτομέρειες του συγκεκριμένου ασθενούς. Ακολούθως, εκτυπώνει τα δεδομένα σε μορφή JSON, ώστε να τα λάβει και να τα παρουσιάσει η client εφαρμογή.

```

public function getPatientsDetails($id) {

  $sql = 'select * from user where id =' . $id;
  $query = $this->db->prepare($sql);
  $result = $query->execute();
  $examsArray = Array();
  $user = $query->fetch(PDO::FETCH_ASSOC);
  $examsArray = Array();
  $examRow = Array();
  $exSql = "select id,userId,details: from exams where userId =$id";
  $exQuery = $this->db->prepare($exSql);
  $exResult = $exQuery->execute();

  while($examRow = $exQuery->fetch(PDO::FETCH_ASSOC)) {

    array_push($examsArray,$examRow);
  }

  $user["exams"] = $examsArray;

  return $user;
}

```

Εικόνα 66 - Η συνάρτηση `getPatientsDetails` του μοντέλου

Η συνάρτηση λαμβάνει ως όρισμα το μοναδικό αναγνωριστικό του ασθενούς και ανακτά από τον πίνακα `users` τις τιμές του, οι οποίες αποθηκεύονται στον πίνακα `user`. Έπειτα, ζητά από τον πίνακα των εξετάσεων πληροφορίες για τις εξετάσεις που έχει κάνει. Οι πληροφορίες αυτές αποθηκεύονται στον πίνακα `examsArray`, ο οποίος αντιστοιχείται στο πεδίο `exams` του πίνακα `user`. Με τον τρόπο αυτό, ο πίνακας `user` συγκεντρώνει όλη την απαιτούμενη πληροφορία.

Το script μετά την ολοκλήρωση της κλήσης, εμφανίζει στη σελίδα το συγκεκριμένο div και γράφει σε ένα πίνακα τις εγγραφές του JSON, αφού πρώτα διαγράψει τις παλαιότερες αν υπάρχουν. Με τον τρόπο αυτό το περιεχόμενο της σελίδας ενημερώνεται με κάθε πάτημα του κουμπιού 'Προβολή Στοιχείων', ανακτώντας κάθε φορά μόνο τα απαραίτητα στοιχεία επιτυγχάνοντας με τον τρόπο αυτό την βέλτιστη διαχείριση των διαθέσιμων πόρων.

```
success: function(data)
{
  $('#fieldDoctors').css("display", "block");
  $('#records_table').css("visibility", "visible");
  $('#tbodyid').empty();
  var trHTML = '';

  trHTML += '<tr><td>ΑΜΚΑ </td><td>' + data.maka + '</td></tr>';
  trHTML += '<tr><td>Όνομα </td><td>' + data.name + '</td></tr>';
  trHTML += '<tr><td>Επώνυμο </td><td>' + data.lname + '</td></tr>';
}
```

Εικόνα 67 - Τμήμα κώδικα της showPatient του script showPatients

Εφαρμογή Ιατρικού Φακέλου

Καλώς όριστες Αλεξία Κωνσταντίνου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να χειριστείτε τους ασθενείς σας.

Λίστα Ασθενών:

ΑΜΚΑ	Όνομα	Επώνυμο	Email	Τηλέφωνο	
05207264852	Κωνσταντίνος	Κωστόπουλος	kostas@gmail.com	6974821535	Προβολή Στοιχείων
05020565845	Πέτρος	Αγγέλου	petran@yahoo.com	6954824658	Προβολή Στοιχείων

Στοιχεία Ασθενούς:

ΑΜΚΑ	05207264852
Όνομα	Κωνσταντίνος
Επώνυμο	Κωστόπουλος
Ημερομηνία Γέννησης	1972-05-20
Υψος	1.95
Βάρος	124.65
Διεύθυνση Ηλ. Ταχ.	kostas@gmail.com
Κινητό Τηλ.	6974821535
Περιγραφή	Αργείο
MRI	Λήψη Αρχείου

[Ενημέρωση Στοιχείων](#)

Εικόνα 68 - Προβολή των στοιχείων του συγκεκριμένου ασθενούς. Οι εξετάσεις του αντιστοιχούν στις εγγραφές που περιέχουν περιγραφή και αρχείο

Αν ο ιατρός επιλέξει άλλον ασθενή το συγκεκριμένο div της σελίδας ανανεώνεται με το νέο περιεχόμενο.

Εφαρμογή Ιατρικού Φακέλου

Καλώς όριστες Αλεξία Κωνσταντίνου

Ενημέρωση Στοιχείων

Μέσω της φόρμας αυτής έχετε τη δυνατότητα να χειριστείτε τους ασθενείς σας.

Λίστα Ασθενών:

ΑΜΚΑ	Όνομα	Επώνυμο	Email	Τηλέφωνο	
05207264852	Κωνσταντίνος	Κωστόπουλος	kostas@gmail.com	6974821535	Προβολή Στοιχείων
05020565845	Πέτρος	Αγγέλου	petran@yahoo.com	6954824658	Προβολή Στοιχείων

Στοιχεία Ασθενούς:

ΑΜΚΑ	05020565845
Όνομα	Πέτρος
Επώνυμο	Αγγέλου
Ημερομηνία Γέννησης	2008-05-02
Ύψος	1.54
Βάρος	51.90
Διεύθυνση Ηλ. Ταχ	petran@yahoo.com
Κινητό Τηλ.	6954824658
Περιγραφή	Αρχείο
Φωτογραφίες Σκολίωσης	Λήψη Αρχείου

Ενημέρωση Στοιχείων

Εικόνα 69 - Προβολή στοιχείων έτερου ασθενούς

			id	userId	1	details	pdf_file			
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	13	0	Σκολίωση	[BLOB - 306,6 KB]
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	17	0	Ormonikes	[BLOB - 128,0 KB]
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	9	3	Εξετάσεις Σακχάρου	[BLOB - 70,9 KB]
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	10	8	Φωτογραφίες Σκολίωσης	[BLOB - 306,6 KB]
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	8	13	MRI	[BLOB - 469,1 KB]
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	11	16	Cholesterol_Exams	[BLOB - 121,4 KB]
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	12	17	Χοληστερίνη	[BLOB - 121,4 KB]
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	14	20	Test_Pap_2020	[BLOB - 253,9 KB]
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	15	21	Υπέρταση	[BLOB - 159,0 KB]
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	7	22	MRI_2020	[BLOB - 469,1 KB]
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή	16	22	Akmi	[BLOB - 199,7 KB]

Εικόνα 70 - Τμήμα του πίνακα Exams. Διακρίνονται οι περιγραφές των εξετάσεων του χρήστη 22 ο οποίος αντιστοιχεί στη Ζωή Γεωργίου της προηγούμενης εικόνας

Τα αρχεία pdf των εξετάσεων, λόγω του ότι αναμένεται να είναι πολλά σε αριθμό, αποφασίστηκε να μην γράφονται όλα στη σελίδα, αλλά να επιλέγει ο κάθε ιατρός το αρχείο που θέλει να δει και κατόπιν αυτό να ανακτάται από τον εξυπηρετητή. Για το λόγο αυτό, στις εγγραφές των εξετάσεων αναγράφεται η περιγραφή και ακολουθεί το κουμπί 'Λήψη Αρχείου'. Το πάτημα του κουμπιού αποστέλλει μια φόρμα στη συνάρτηση getPdf του αντικειμένου DocController με το αναγνωριστικό της συγκεκριμένης εξέτασης.

```

for(i=0;i<data.exams.length;i++){
    varHTML += '<tr><td>' + data.exams[i].details + '</td><td><div class="formItem"><form class="form" action="http://192.168.7.65/healthcare/DocController/getPdf" method="post">
    <input type="hidden" id="pdfId" name="pdfId" value="'+data.exams[i].id +' "><button class="formBtn" type="submit"> Λήψη Αρχείου </button></form></td></tr>';
}

```

Εικόνα 71 - Τμήμα κώδικα της showPatient του script showPatents το οποίο αναγράφει στο πίνακα τη περιγραφή της κάθε εξέτασης και μια φόρμα για την ανάκτηση του αρχείου

Η getPdf από την πλευρά της, λαμβάνει το αναγνωριστικό id της εξέτασης και ζητάει από το μοντέλο το BLOB περιεχόμενο του πίνακα. Το περιεχόμενο θα σταλεί σαν αρχείο στο client και κατά συνέπεια ορίζονται κατάλληλα τα http Headers και εκτυπώνεται η πληροφορία.

```
public function getPdf(){
    $usersModel = $this->loadModel('userModel');
    $result = $usersModel->getPdf($_POST["pdfId"]);
    header("Content-type: application/pdf");
    header('Content-disposition: inline; filename="examps.pdf");
    header('Content-Transfer-Encoding: binary');
    header('Accept-Ranges: bytes');
    echo $result["pdf_file"];
}
```

Εικόνα 72 - Η συνάρτηση getPdf του DocControler

Συνεπώς, μόλις ο ιατρός επιλέξει το κουμπί 'Λήψη Αρχείου', το συγκεκριμένο αρχείο μεταφέρεται στον υπολογιστή του και είναι έτοιμο για χρήση.

4. Επίλογος - Συμπεράσματα

Στόχος της παρούσας πτυχιακής εργασίας ήταν να παρουσιάσει την ιδέα του Ηλεκτρονικού Φακέλου Ασθενούς καθώς και τις δυνατότητες που έχει να προσφέρει η τεχνολογία αυτή. Για το σκοπό αυτό αναλύθηκαν οι λόγοι που οδήγησαν στη δημιουργία του συστήματος αυτού, τα χαρακτηριστικά του, καθώς και τα πλεονεκτήματα αλλά και τα σημεία που χρήζουν την προσοχή των σχεδιαστών. Ακολούθως, έγινε μια αναφορά στο νομικό πλαίσιο της Ελλάδος που ορίζει τα πλαίσια εντός των οποίων μπορεί να κινηθεί το σύστημα, και στη συνέχεια παρουσιάστηκαν υλοποιήσεις του τόσο στην Ελλάδα όσο και στο εξωτερικό.

Στο επόμενο κεφάλαιο έγινε μια εισαγωγή στην υπολογιστική νέφους, η οποία θα φιλοξενούσε την εφαρμογή που θα υλοποιούνταν στη συνέχεια. Ακολούθως, παρουσιάστηκαν αναλυτικά τα βήματα ανάπτυξης της εφαρμογής.

Η συγγραφέας εκτιμά ότι μέσω της παραπάνω παρουσίασης, ο αναγνώστης θα αποκτήσει μια αντιπροσωπευτική εικόνα του τι ακριβώς αφορούν οι τεχνολογίες που παρουσιάζονται και με αυτόν τον τρόπο θα σχηματίσει τη δική του, εμπειριστατωμένη πλέον άποψη για το αν και κατά πόσο είναι χρήσιμες και απαραίτητες ενδεχομένως, για τη σημερινή κοινωνία. Επιπλέον, μέσα από την παρουσίαση της υλοποίησης της εφαρμογής θα αποκτήσει αρκετή εμπειρία, ώστε να είναι σε θέση να αναπτύξει τα δικά του συστήματα όταν και όποτε αυτό απαιτηθεί.

4.1 Σημεία που δυσκόλεψαν την συγγραφέα

Αρκετά ήταν ωστόσο τα σημεία που δυσκόλεψαν την συγγραφέα και τα περισσότερα εξ αυτών εντοπίζονται στην υλοποίηση της εφαρμογής. Οι αρχιτεκτονικές αλλά και οι τεχνολογίες που απαιτούνται προκειμένου να λειτουργήσει ένα σύστημα σαν και αυτό που περιγράφεται είναι πολλές και διαφορετικές, και η διαδικασία του συγχρονισμού και της συνεργασίας όλων αυτών των διαφορετικών οντοτήτων είναι μια ιδιαίτερα κοπιαστική εργασία. Απαιτείται γνώση πολλών και διαφορετικών επιπέδων, αλλά και εμπειρία και υπομονή ώστε να επιτευχθεί η υλοποίηση ενός συστήματος σαν και αυτό που περιγράφεται. Ευτυχώς, μέγας αρωγός στη προσπάθεια αυτή είναι η παγκόσμια κοινότητα, η οποία μέσω του ίντερνετ προσφέρει την απαιτούμενη γνώση αλλά και λύσεις για τα διάφορα προβλήματα που εμφανίζονται κατά την ανάπτυξη της εφαρμογής.

4.2 Μελλοντικές προσθήκες

Όπως συχνά συμβαίνει στην επιστήμη της πληροφορικής, μοναδικό όριο στις προσθήκες που μπορούν να γίνουν στο υπάρχον σύστημα είναι η φαντασία. Πράγματι, είναι αναρίθμητες οι προσθήκες που θα μπορούσε να σκεφτεί κανείς και περιλαμβάνουν λειτουργίες όπως αποθήκευση περισσότερων λεπτομερειών για το κάθε χρήστη (εμβόλια, αλλεργίες) αλλά και τεχνικές μηχανικής μάθησης που θα δέχονται ως είσοδο τα συμπτώματα του ασθενούς και με βάση τη συμπεριφορά άλλων ασθενών με παρόμοια συμπτώματα, θα μπορούν να εξάγουν ασφαλή συμπεράσματα τόσο για την πιθανή ασθένεια όσο και για τη θεραπεία της. Φυσικά, σε όλη αυτή διαδικασία θα πρέπει να λαμβάνεται ιδιαίτερη μέριμνα για τη προστασία των ευαίσθητων προσωπικών δεδομένων αλλά και της ιδιωτικότητας του ατόμου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [01] Χουλιάρας Δ. (2006). *Πρότυπα Μοντέλα Αναφοράς, Αναπαράσταση Γνωστικής Πληροφορίας Σχεδιαστικοί Περιορισμοί και Προδιαγραφές*. Πανεπιστήμιο Πατρών, Τμήμα Ιατρικής, Εθνικό Μετσόβιο Πολυτεχνείο, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ, Τμήμα Μηχανολόγων Μηχανικών.
<https://docplayer.gr/11867124-Protypa-montela-anaforas-anaparastasi-gnostikis-pliroforias-shediastikoi-periorismoi-kai-prodiagrafes.html>
- [02] Μαντάς Ι. & Μπλέτσα Β. (2012). Μία θεωρητική προσέγγιση του ηλεκτρονικού φακέλου ασθενούς. *Περιεχειρητική Νοσηλευτική*. Σύλλογος Διπλωματούχων Νοσηλευτών Χειρουργείου (ΣΥ.Δ.ΝΟ.Χ.), 42-56.
https://www.spnj.gr/articlefiles/volume1_issue2/122.pdf
- [03] Hunter, David. (2003). *Public Health Policy*. Ηνωμένο Βασίλειο: Polity Press.
https://www.researchgate.net/profile/David-Hunter-3/publication/30052803_Public_Health_Policy/links/56efe21208ae01ae3e70e230/Public-Health-Policy.pdf
- [04] Νόμος 4486/2017, άρθρο 21, Ατομικός Ηλεκτρονικός Φάκελος Υγείας, Εφημερίδα της Κυβερνήσεως της Ελληνικής Δημοκρατίας (ΦΕΚ Α' 115/07.08.2017)
<https://www.kodiko.gr/nomothesia/document/274963/nomos-4486-2017>
- [05] Εγχειρίδιο οδηγιών (ΠΦΥ). <https://ehealth.gov.gr/p-rv/p>
- [06] Microsoft HealthVault. https://en.wikipedia.org/wiki/Microsoft_HealthVault
- [07] Mell, P. and Grance, T. (2011), The NIST Definition of Cloud Computing, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], <https://doi.org/10.6028/NIST.SP.800-145>
- [08] Liu, F. , Tong, J. , Mao, J. , Bohn, R. , Messina, J. , Badger, M. and Leaf, D. (2011), NIST Cloud Computing Reference Architecture, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, <https://doi.org/10.6028/NIST.SP.500-292>

- [09] Google Cloud Platform. <https://cloud.google.com/>
- [10] XAMPP. <https://www.apachefriends.org/docs/hosting-xampp-on-google.html>
- [11] PuTTY. <https://www.putty.org/>
- [12] FileZilla. <https://filezilla-project.org>

