Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Σχεδιασμός και Ανάπτυξη τμήματος Δικτυακού ΠληροφοριακούΣυστήματος Νοσοκομείων** **Design and Develop a part of a Web Based Hospital Information System**
Ονοματεπώνυμο Φοιτητή	**Χερολίδης Διονύσιος**
Πατρώνυμο	**Σταύρος**
Αριθμός Μητρώου	**ΜΠΣΠ/10055**
Υπεύθυνος Καθηγητής	**Χρήστος Δουληγέρης, Καθηγητής**
Επιβλέπων Συνεργάτης	**Δρ. Σαράντης Μητρόπουλος**

Ημερομηνία Παράδοσης **Απρίλιος2014**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Χρήστος Δουληγέρης Καθηγητής	ΧαράλαμποςΚωνσταντόπουλος Επίκουρος Καθηγητής	Παναγιώτης Κοτζανικολάου Λέκτορας

Ευχαριστίες

Η εργασία αυτή αναπτύχθηκε στο πλαίσιο μεταπτυχιακής μου διατριβής του Π.Μ.Σ. Προηγμένα Συστήματα Πληροφορικής. Βασίστηκε σε εργασία του μαθήματος "Πληροφοριακά συστήματα στο διαδίκτυο" του καθηγητή του τμήματος Δρ.Σαράντη Μητρόπουλου τον οποίο και ευχαριστώ για την επίβλεψη της παρούσας εργασίας και την βοήθεια που παρείχε σε τεχνολογικές και θεωρητικές γνώσεις.

Επίσης θα ήθελα να ευχαριστήσω πολύ και την αδελφή μου Δρ. Ελένη Χερολίδου ειδικευόμενη ιατρό στον ευαγγελισμό και την μητέρα μου Δρ. Αθηνά Παναγιώτοβα ιδιωτική ιατρό για τις γνώσεις που μου παρείχαν πάνω σε θέματα υγείας και παρόμοιων ηλεκτρονικών συστημάτων νοσοκομείων σαν και αυτό που δημιουργήσαμε κατά την διάρκεια της μεταπτυχιακής διατριβής.

Περίληψη. Η παρούσα μεταπτυχιακή διατριβή παρουσιάζει ένα Νοσοκομειακό Πληροφοριακό Σύστημα που θα μπορούσε να χρησιμοποιηθεί σε ιδιωτικά και δημόσια νοσοκομεία, κλινικές και ιατρεία. Αναλύουμε την χρησιμότητα αυτών των συστημάτων, τα οφέλη που μας προσφέρουν και την ανάγκη ύπαρξής τους σε κάθε οργανισμό. Αντίστοιχα, θα περιγράψουμετις λειτουργικές δυνατότητες που

έχει η προτεινόμενη εφαρμογήδίνοντας έμφασηστις τεχνολογίες που χρησιμοποιήθηκαν (Java, Spring, SpringSecurity, Hibernate κτλ.) για την υλοποίησή της. Μελετούμε και το επίπεδο ασφάλειας τις εφαρμογής καθώς έχουμε να κάνουμε με προσωπικά δεδομένα ασθενών τα οποία είναι απόρρητα και πρέπει να δώσουμε προσοχή στο ποιος και πώς τα διαχειρίζεται. Τελικά, θα δούμε ότι είναι εφικτή η δημιουργία ενός τέτοιου συστήματος με μικρό οικονομικό κόστος.

Abstract. This dissertation presents a Hospital Information System which could be used in private and public hospitals, clinics and dispensaries. We analyze the utility of these systems,the benefits which areoffered and the need of their usage in every single organization.Respectively,we describe the functional capabilities of the application, focusing on the technology used (Java, Spring, Spring Security, Hibernate, etc.) for its implementation.We will also study the security level of the proposedsystem as we have to manage private data of patients that are classified and we have to be careful with who and how anyone is managing them.Eventually we will see that the establishment of such a system is achievable/feasible without spending large amount of money.

Περιεχόμενα

Πίνακας εικόνων

Λίστα πινάκων

Εισαγωγή

Πληροφοριακό σύστημα (Information System ή IS) ονομάζεται "ένα σύνολο διαδικασιών, ανθρώπινου δυναμικού και αυτοματοποιημένων υπολογιστικών συστημάτων, που προορίζονται για τη συλλογή, εγγραφή, ανάκτηση, επεξεργασία, αποθήκευση και ανάλυση πληροφοριών"[1].Η χρήση ενός τέτοιου συστήματος σε συνδυασμό με την ασφάλεια και την ταχύτητα της λειτουργίας του, όχι μόνο συνίσταται αλλά μπορούμε να πούμε ότι επιβάλλεται σε οποιαδήποτε σύγχρονη και σοβαρή εταιρία ή οργανισμόπου σέβεται τους εργαζόμενους και τους πελάτες και θέλει τα μέγιστα και ποιοτικότερα αποτελέσματα.

Στην ίδια κατηγορία ανήκουν και τα νοσοκομεία και γενικότερα κάθε είδους κέντρα υγείας που παρέχουν υπηρεσίες υγείας τα οποία μπορούν να επωφεληθούν χρησιμοποιώντας πληροφοριακά συστήματα. Με την χρήση τους θα μπορέσουμε να οργανώσουμε τον μεγάλο όγκο δεδομένων που έχουμε για τους υπαλλήλους, τους ασθενείς, τα εργαστηριακά τεστ, τα φάρμακα και γενικότερα όποιες πληροφορίες αφορούν την χρήση του Νοσοκομείου.Έτσι θα πετύχουμε και οργανωτικά-λειτουργικά οφέλη και ποιοτικότερη και ταχύτερη εξυπηρέτηση καθώς θα επιτύχουμε μία καλύτερη συνεργασία του προσωπικού.Φυσικά δεν μπορούμε να παραλείψουμε ότι η συγκέντρωση μεγάλου όγκου ιατρικών δεδομένων που είναι εύκολα προσβάσιμος από ιατρούς και επιστήμονες μπορεί επίσης να ωφελήσει σε μελλοντικές ιατρικές μελέτες και έρευνες.

Τέτοιου είδους συστήματα που χρησιμοποιούνται στα νοσοκομεία ονομάζονται 'Πληροφοριακά Συστήματα Νοσοκομείου (ΠΣΝ)' (Hospital Information System 'HIS').Στην παρούσα διατριβήθα μελετήσουμε την κατασκευή ενός παρόμοιου συστήματος και το πώς μπορούν να μας χρησιμεύσουν οι νέες τεχνολογίες. Ένα ΠΣΝμπορεί να περιλαμβάνει ηλεκτρονικές υπηρεσίες όπως το φάκελο υγείας, τη συνταγογράφηση, τη διαχείριση φαρμακευτικού υλικού, την οικονομική διαχείριση (λογιστικά), τη διαχείριση προσωπικού, την καταγραφή εργαστηριακών εξετάσεων και γενικότερα οτιδήποτε άλλο έχει να κάνει με την λειτουργία ενός νοσοκομείου.Η μορφή ενός τέτοιου συστήματος μπορεί να είναι είτε σαν εκτελέσιμο πρόγραμμα το οποίο θα μας περιορίσει σε συγκεκριμένο λειτουργικό σύστημα (πχ, Windows,Linux, MAC κτλ.) είτε σαν διαδικτυακή εφαρμογή (WebApplication) η οποία θα μπορεί να εκτελεστεί σε οποιαδήποτε συσκευή που να μπορεί να συνδεθεί σε τοπικό δίκτυο ή στο διαδίκτυο και διαθέτει πρόγραμμα περιήγησης (webbrowser). Την περίπτωση αυτή της διαδικτυακήςεφαρμογής θα μελετήσουμε στην παρούσα διατριβή για να επωφεληθούμε από την ποικιλία των εναλλακτικών μεθόδων σύνδεσης που μας δίνει. Για παράδειγμα, σε ένα νοσοκομείο θα μπορούν να έχουν πρόσβαση στην εφαρμογή όλοι οι υπάλληλοι από τα γραφεία τους μέσω υπολογιστών, από του υπόλοιπους χώρους του νοσοκομείου μέσω κάποιουυπολογιστή ταμπλέτας[2]και από οποιοδήποτε σημείο στον κόσμο που έχει πρόσβαση στο διαδίκτυο μέσο κινητού ή άλλου υπολογιστή.

Για να προσεγγίσουμε ένα ικανοποιητικό αποτέλεσμα στην εργασία αυτή αναλύσαμε και μελετήσαμε τις ανάγκες ενός νοσοκομείου τις οποίες και θα μπορούσαμε να καλύψουμε ή να διευκολύνουμε τουλάχιστον, με την χρήση ενός Πληροφοριακού Συστήματος Νοσοκομείου. Για να συγκεντρωθούν αυτές οι πληροφορίες υπήρξε συνεχόμενη επικοινωνία με ιατρούς που δουλεύουν ή δούλευαν σε νοσοκομεία είτε χρησιμοποιώντας παρόμοια πληροφοριακά συστήματα είτε όχι. Αφού δημιουργήθηκε ένα πλάνο με τις λειτουργίες που πρέπει να έχει το σύστημά μας ξεκίνησε η τμηματική υλοποίηση της εφαρμογής και ταυτόχρονα γινόταν έλεγχος σωστής λειτουργίας των τμημάτων που υλοποιήθηκαν. Για τον έλεγχο γινόταν χρήση της εφαρμογήςκαι των λειτουργιών που υλοποιούνταν και από την πλευρά του προγραμματιστή αλλά και από ιατρούςγια πιο έγκυρα αποτελέσματα.

Σκοπός τις παρούσας εργασίας δεν είναι να παρουσιάσουμε ένα ολοκληρωμένο Πληροφοριακό Σύστημα Νοσοκομείου το οποίο είναι καλύτερο ή πληρέστερο από άλλα που υπάρχουν ήδη στην αγορά. Αυτό θα ήταν αδύνατο καθώς τα συστήματα που υπάρχουν έχουν αναπτυχθεί από μεγάλες πληροφοριακές εταιρίες του χώρου που έχουν ξοδέψει τεράστια ποσά χρημάτων και χρησιμοποιούν πολλούς εξειδικευμένους προγραμματιστές με μεγάλη εμπειρία. Χαρακτηριστικό είναι ότι το κράτος μας έχει ξοδέψει πολλά εκατομμύρια ευρώ τα τελευταία χρόνια για να αναπτυχθούν τέτοια συστήματα και έχει αναθέσει σε πολλές και διάφορες εταιρίες πληροφορικής την υλοποίησή τους. Παρά ταύτα τα συστήματα αυτά παρουσιάζουν πολλά προβλήματα και υπάρχουν δυσκολίες προσαρμογής από τους υπαλλήλουςλόγω πολυπλοκότητας και μειωμένης γνώσης χρήσης ηλεκτρονικών υπολογιστών και γενικότερα σύγχρονων τεχνολογιών. Εμείς θα επιχειρήσουμε να δημιουργήσουμε ένα (ΠΣΝ)

χρησιμοποιώντας σύγχρονες τεχνολογίες οι οποίες είναι δωρεάν και θα δείξουμε ότι ένα τέτοιο σύστημα δεν χρειάζεται τεράστια δαπάνη για να υλοποιηθεί.

1. Χώρος του προβλήματος

1.1 Διαχείριση Νοσοκομείου

Πριν από μερικές δεκαετίες όλα τα νοσοκομεία χρησιμοποιούσαν τις παραδοσιακές χειρόγραφες μεθόδους για να αποθηκεύσουν τους φακέλους των ασθενών και γενικότερα οποιαδήποτε πληροφορία αφορούσε το νοσοκομείο. Αυτό είχε σαν αποτέλεσμα να απαιτούνται μεγάλοι χώροι για την αποθήκευση και την φύλαξή τους οι οποίοι ποτέ δεν ήταναρκετοί μιας και τα δεδομένα που πρέπει να αποθηκευτούν αυξάνονται συνεχώς με τον χρόνο.

Εικόνα 1 Φάκελοι ασθενών σε Νοσοκομείο.

Για αυτό τον λόγω έπρεπε να βρεθεί μία πιο πρακτική λύση,η οποία με τη ραγδαία εξέλιξη της πληροφορικής έγινε εφικτή. Ο όρος «Ιατρική Πληροφορική» έχει κάνει την εμφάνισή του περίπου πριν από 40 χρόνια. Στην αρχή ο όρος αυτός κάλυπτε όλους τους χώρους της Υγείας αν και η χρήση των υπολογιστών περιορίζονταν μόνο στην Ιατρική επιστήμη [3]. Στη συνέχεια όμως η χρήση των υπολογιστών επεκτάθηκε στους Επαγγελματίες Υγείας περικλείοντας όλες τις μορφές χρήσης, από τις τελείως θεωρητικές ως τις εφαρμοσμένες. Έτσι άρχισαν να δημιουργούνται τα Πληροφοριακά Συστήματα Νοσοκομείου (ΠΣΝ)(Hospital Information System 'HIS'). Ως Πληροφοριακό Σύστημα Νοσοκομείου (ΠΣΝ) χαρακτηρίζουμε εκείνο το υπολογιστικό σύστημα το οποίο φροντίζει για τη συνύπαρξη και την επικοινωνία της εξωτερικής και της εσωτερικής ροής των πληροφοριών σε ένα νοσοκομείο, καθώς και για τον κοινό τρόπο λειτουργίας των εφαρμογών (λογισμικό) που χρησιμοποιούνται μέσα στο νοσοκομείο. Σκοπός των πληροφοριακών υποδομών και των συστημάτων είναι να αναγνωρίσουν, να προετοιμάσουν, να ελέγξουν και να αξιοποιήσουν υπηρεσίες ηλεκτρονικής υγείας (η-υγεία). Στο χώρο της υγείας, το 80% των σχεδιασμένων ηλεκτρονικών υπηρεσιών έχουν εκχωρηθεί για εξάσκηση, εκπαίδευση και κοινωνικές αλληλεπιδράσεις των ασθενών, ενώ μόνο μερικές (20%) ασχολούνται με την πρόληψη και διαχείριση της υγείας.

Ο τελικός στόχος ενός ΠΣΝ είναι να συλλέγει, αποθηκεύει, επεξεργάζεται και ανακτά πληροφορίες, με τη χρήση Η/Υ και επικοινωνιακού εξοπλισμού, σχετικά με την περίθαλψη των ασθενών και όλες τις διοικητικές λειτουργίες για να ικανοποιήσει τελικά τις λειτουργικές ανάγκες όλων των εξουσιοδοτημένων χρηστών. Υπάρχουν πάρα πολλάέργα που καθιερώνουν στρατηγικές απαιτήσεων ασθενών, μελετών αξιολόγησης σε πιλοτικές φάσεις, επιχειρηματικά μοντέλα, κτλ., τα οποία συγκλίνουν στα επόμενα συμπεράσματα**Error! Reference source not found.**:

- Η επιλογή των κατάλληλων ανθρώπων που θα εμπλακούν σε κάθε πραγματοποίηση του έργου
- Η δέσμευση της διαχείρισης των οργανισμών που εμπλέκονται στο έργο
- Η ευκολία χρήσης και αξιοπιστίας του συστήματος καθώς και η διαθεσιμότητα της υπηρεσίας

Είναι σημαντικό να κατανοήσουμε ότι αυτά τα συμπεράσματα έχουν την ίδια βαρύτητα, και ότι η παράβλεψη ενός και μόνου από αυτά είναι σε θέση να εμποδίσει την επιτυχία του έργου.

Η ανάπτυξη των Πληροφοριακών Συστημάτων Υγείας μπορείνα ωφελήσει σε μεγάλο βαθμό στην καλύτερη οργάνωση, διαχείριση και διοίκηση του νοσοκομείου. Έτσι θα συμβάλει στη μείωση του υπέρογκου κόστους κατά τη νοσηλεία των ασθενών, μια και υπάρχει η δυνατότητα ελέγχου των υλικών, εξετάσεων κ.λπ., και στη βελτίωση της παραγωγικότητας σε τομείς όπως η τιμολόγηση και η αρχειοθέτηση, η μείωση των ιατρονοσηλευτικών λαθών, ο περιορισμός των αδικαιολόγητων θεραπειών, αλλά και η βελτίωση της ποιότητας της υγειονομικής περίθαλψης [5]. Αυτό φυσικά θα έχει σαν αποτέλεσμα τον περιορισμό των δαπανών και την αύξηση των κερδών κάτι που στην εποχή που ζούμε είναι εξαιρετικά σημαντικό.

Ένα πληροφοριακό σύστημα μπορεί να παρέχει όλες τις παρακάτω υπηρεσίες ή κάποιες από αυτές:

- Πρόσβαση από παντού (Νοσοκομείο, σπίτι, διαδίκτυο).
- Πρόσβαση όλο το 24ωρο κάθε μέρα του χρόνου.
- Πρόσβαση σε όλους στα τμήματα που τους αφορούν (Διευθυντής, γιατροί, νοσοκόμοι, φαρμακοποιοί, εργαστηριακοί ιατροί ασθενείς κτλ.).
- Υπηρεσία ενημέρωσης ασθενών για αποτελέσματα εξετάσεων, μελλοντικές εξετάσεις, ραντεβού κτλ (ηλεκτρονικά μέσω του Portal, με e-mail ή και με κινητό τηλέφωνο).
- Ιατρικό και νοσηλευτικό υποσύστημα που διαχειρίζεται τον Ιατρικό φάκελο και τα Ιατρικά και Νοσηλευτικά πρωτόκολλα.
- Υποσύστημα εργαστηρίων Laboratory Information System (LIS) που διαχειρίζεται τα αποτελέσματα των αναλύσεων.
- Διαχειριστικό υποσύστημα, που μπορεί να είναι ένα ERP για την διαχείριση αποθηκών, υλικών, παραγγελιών, προσωπικού κλπ.
- Υποσύστημα απεικονιστικών μηχανημάτων (RIS) για την διαχείριση των ιατρικών εικόνων από τα αντίστοιχα μηχανήματα.
- Υποσύστημα παρακολούθησης εξοπλισμού, EquipmentMaintenanceSystem (EMS)
- Υποσυστήματα για το κάθε ιατρικό τμήμα.
- Συμβατότητα με την ηλεκτρονική συνταγογράφηση [6].
- Πολυγλωσσικότητα.

Από αυτά φυσικά θα έχουμε τα εξής πλεονεκτήματα:

- Καλύτερη οργάνωση του μεγάλου όγκου πληροφοριών.
- Καλύτερη διαχείριση και παρακολούθηση των τμημάτων του νοσοκομείου.
- Μείωση Γραφειοκρατίας (περιορισμός χειρόγραφων διαδικασιών κτλ.).
- Μείωση των διοικητικών εργασιών.
- Μείωση ανθρώπινων λαθών.
- Βελτίωση ποιότητας υπηρεσιών.
- Καλύτερη φροντίδα των ασθενών.
- Μείωση του χρόνου αναμονής.
- Μείωση του χρόνου νοσηλείας.
- Μείωση κόστους περίθαλψης (ορθολογικότερη διαχείριση, αποφυγή άσκοπων ιατρικών πράξεων, κτλ.) Καλύτερη επικοινωνία μεταξύ των εμπλεκομένων στον χώρο της υγείας.
- Βελτίωση της χρήσης των πόρων.
- Καλύτερη διαχείριση του κόστους.
- Καλύτερη συνολική λειτουργία.
- Καλύτερη εξυπηρέτηση και φροντίδα των πελατών και βελτίωση εικόνας προς τους πελάτες.
- Αύξηση παραγωγικότητας.

Οι δύο βασικοί άξονες ενός ΠΣΝ είναι το Κλινικό και το Διοικητικό Πληροφοριακό Σύστημα. Το Κλινικό Πληροφοριακό Σύστημα περιλαμβάνει νοσηλευτικά συστήματα, συστήματα παρακολούθησης, συστήματα καταχώρησης εντολών, συστήματα διαχείρισης φαρμακείου, καθώς και εργαστηριακά και ακτινολογικά συστήματα (Laboratory Information System - LIS). Το Διοικητικό Σύστημα σχετίζεται με συστήματα εγγραφής ασθενών, συστήματα διαχείρισης οικονομικών στοιχείων, ανθρώπινων πόρων, κινδύνου, συναλλαγών με τρίτους και συστήματα διασφάλισης ποιότητας.

Για να θεωρήσουμε τώρα ένα ΠΣΝ αξιόπιστο και να μπορεί να χρησιμοποιηθεί χωρίς προβλήματα και δυσκολίες πρέπει να λάβουμε υπόψη μας τις παρακάτω παραμέτρους:

- Πιστοποίηση (authentication): έλεγχος της αυθεντικότητας της ταυτότητας των μερών μιας ανταλλαγής δεδομένων.
- Εξουσιοδότηση (authorisation): η πρόσβαση του χρήστη πρέπει να είναι εξουσιοδοτημένη.
- Εμπιστευτικότητα (confidentiality): η τήρηση του απορρήτου των δεδομένων.
- Ακεραιότητα (integrity): τα δεδομένα θα πρέπει να παραμείνουν ακέραια, δηλαδή να μην υποστούν αλλοίωση.
- Μη δυνατότητα άρνησης συμμετοχής (non-repudiation): ο χρήστης δεν πρέπει να μπορεί να αρνηθεί τη συμμετοχή του στην ανταλλαγή των δεδομένων.
- Δυνατότητα ελέγχου (revision / audit): κάθε τροποποίηση ή επεξεργασία των δεδομένων πρέπει να μπορεί να ελεγχθεί, δηλαδή από ποιόν έγινε και πότε.
- Ευθύνη (accountability): πρέπει να προκύπτει ποιος είναι υπεύθυνος για την εισαγωγή, πρόσβαση ή τροποποίηση κάθε δεδομένου.
- Διαφάνεια (transparency): πρέπει να γίνεται τεκμηρίωση των διαδικασιών της επεξεργασίας ώστε να μπορούν να ελεγχθούν.
- Διαθεσιμότητα (availability): τα δεδομένα πρέπει να είναι διαθέσιμα όταν απαιτείται.

1.2 Πληροφοριακά Συστήματα Νοσοκομείου (ΠΣΝ) στην Ελλάδα

Όπως είναι γνωστό η ιατρική είναι από τις σπουδαιότερες επιστήμες για τον άνθρωπο και η ανάπτυξη που έχει τα τελευταία χρόνια με την βοήθεια της τεχνολογίας είναι ραγδαία. Όλες οι ανεπτυγμένες χώρες επενδύουν μεγάλα κεφάλαια στην υλοποίηση και ανάπτυξη πληροφοριακών συστημάτων για την υγεία αν και στην χώρα μας οι επενδύσεις τα τελευταία χρόνια είναι περιορισμένες λόγω της οικονομικής κρίσης. Στην Ελλάδα δεν χρησιμοποιούνται ευρέως τα Πληροφοριακά Συστήματα Υγείας εξαιτίας και των σημαντικών ελλείψεων εκπαιδευμένου και εξειδικευμένου προσωπικού, απουσίας από τον τακτικό προϋπολογισμό των φορέων υγείας ικανού ποσοστού επενδύσεων για την ανάπτυξη της πληροφορικής και απουσίας θεσμικού φορέα για θέματα Ιατρικής Πληροφορικής.

Η ανάγκη για ένα ολοκληρωμένο πληροφοριακό σύστημα υγείας στην εποχή μας είναι μεγάλη. Τα προβλήματα που αντιμετωπίζουν τα κέντρα υγείας σήμερα είναι πολλά λόγω της πολυπλοκότητας της διαχείρισης και των ξεπερασμένων μεθόδων που χρησιμοποιούνται οι οποίες δεν διευκολύνουν καθόλου το έργο των εργαζόμενων. Αυτό είναι γνωστό και στην κυβέρνηση και στους υπεύθυνους του κάθε κέντρου υγείας γι'αυτό και γίνονται πολλές προσπάθειες υλοποίησης και εφαρμογής τέτοιων συστημάτων χωρίς όμως τα αναμενόμενα αποτελέσματα. [7][8]

Συμπεραίνουμε λοιπόν ότι η επιτυχής ολοκλήρωση ενός τέτοιου συστήματος θα φέρει πολλά οφέλη και δεν θα μείνει απαρατήρητη. Για να είναι όμως επιτυχημένο το σύστημά μας πρέπει να παρατηρήσουμε και να μάθουμε από τα λάθη του παρελθόντος και να μελετήσουμε παρόμοια συστήματα που λειτουργούν στην Ελλάδα και στο Εξωτερικό.

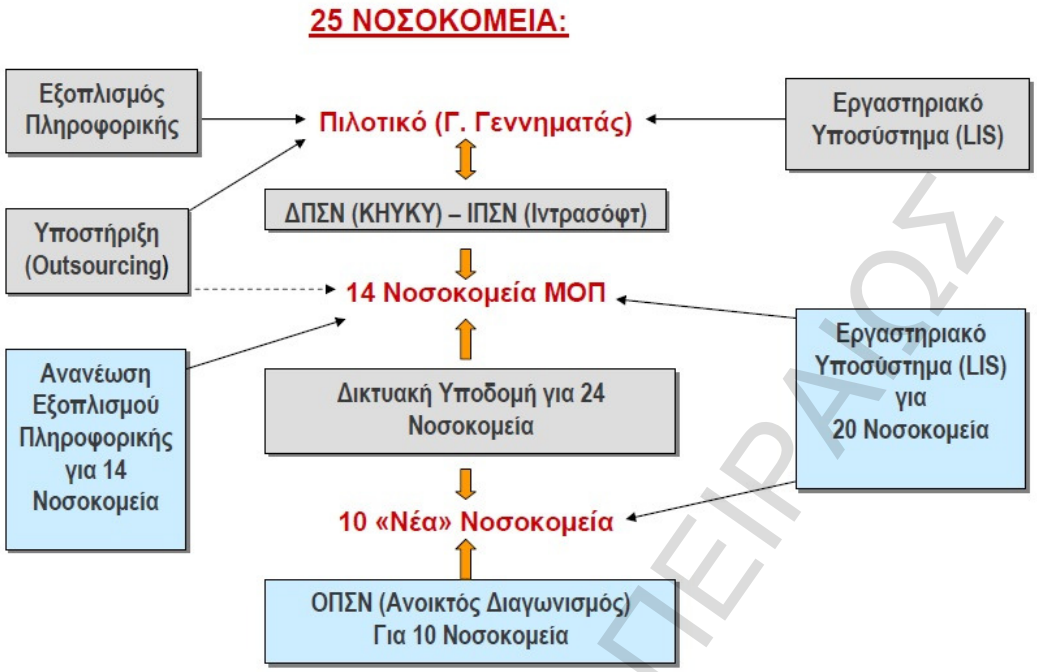
1.2.1 Εξέλιξη ΠΣΝ στην Ελλάδα

Η εξέλιξη των Πληροφοριακών συστημάτων στην Ελλάδα είναι αργή και χρονικά καθυστερημένη σε σχέση με την πρόοδοάλλων χωρών καθώς ακολουθείτην εξέλιξη της πληροφορικής στην χώρα μας η οποία είναι αρκετά χρόνια πίσω από την διεθνή ανάπτυξη.Έτσι οι πρώτες απόπειρες χρήσης

υπολογιστικών συστημάτων στα νοσοκομεία ξεκίνησαν προς το τέλος της δεκαετίας του 1980 κυρίως στα οικονομικά τμήματα για την διαχείριση απλών οικονομικών στοιχείων που αφορούν την λειτουργία του νοσοκομείου. Αυτό δεν απέφερε θεαματικά αποτελέσματα διότι οι λειτουργίες ήταν περιορισμένες λόγωτου περιορισμού της τεχνολογίας και της έλλειψης διασύνδεσης με άλλα τμήματα. Επίσης ο κύριος όγκος της δουλειάς και της γραφειοκρατίας που απαιτούνται σε ένα νοσοκομείο βρίσκεται στα υπόλοιπα τμήματα που αφορούν κυρίως τον ασθενή.

Η προσπάθεια ανάπτυξης της χρήσης εφαρμογών Τεχνολογίας Πληροφορικής και Επικοινωνίας (Τ.Π.Ε.) στη χώρα μας στον χώρο των Νοσοκομείων άρχισε με την έγκριση, το 1985, των Μεσογειακών Ολοκληρωμένων Προγραμμάτων (Μ.Ο.Π.) και συνεχίστηκε με τα έργα του Β' και Γ' Κοινοτικού Πλαισίου Στήριξης. Κεντρικό σημείο αναφοράς των μέχρι σήμερα δράσεων, αποτέλεσε η προσπάθεια για εισαγωγή πληροφοριακών συστημάτων στα Νοσοκομεία της χώρας. Στα πλαίσια του Β' Κ.Π.Σ οι σχετικές παρεμβάσεις ήταν μικρής κλίμακας και περιορίστηκαν στο επίπεδο του σχεδιασμού. Πιο συγκεκριμένα εκπονήθηκαν:[9]

- Ένα σύνολο μελετών για την κωδικοποίηση - ταξινόμηση ιατρικών δεδομένων. Οι κωδικοποιήσεις αυτές, πέρασαν από μια συστηματική διαδικασία ελέγχων από ιατρικές εταιρείες και επιστημονικούς συλλόγους και δοκιμάζεται η χρήση τους στο πληροφοριακό σύστημα που έχει εγκατασταθεί στο πιλοτικό Νοσοκομείο «Γ. ΓΕΝΝΗΜΑΤΑΣ». Επίσης, στα πλαίσια της προσπάθειας διάδοσής τους, διανέμονται στα Νοσοκομεία μέσω της ιστοσελίδας του Υπουργείου Υγείας Πρόνοιας (Υπ.Υ.Π.).
- Ένα σύνολο προδιαγραφών για τον σχεδιασμό, την υλοποίηση και τη λειτουργία πληροφοριακών συστημάτων στο χώρο της υγείας. Οι προδιαγραφές αυτές χρησιμοποιήθηκαν κατά βάση στα έργα του Β' Κ.Π.Σ., αλλά μπορούσαν να αξιοποιηθούν,μετά από κατάλληλη επικαιροποίηση και στο Γ' Κ.Π.Σ.
- Επιχειρησιακό σχέδιο για την εφαρμογή τηλεϊατρικών υπηρεσιών σε απομακρυσμένες περιοχές της χώρας ή περιοχές με ανεπαρκή νοσοκομειακή υποστήριξη. Το σχέδιο ανέδειξε τα αναγκαία θεσμικά μέτρα που πρέπει να ληφθούν, ώστε να δημιουργηθεί μια πρώτη κρίσιμη μάζα χρηστών και να διευκολυνθεί η παραγωγική χρήση συστημάτων τηλεϊατρικής.
- Για την εκπαίδευση του προσωπικού στην πληροφορική, ενέργεια περιορισμένης κλίμακας, που αφορούσε μόνο βασική εκπαίδευση στη χρήση των υπολογιστών και γενική εισαγωγή στη θεωρία των πληροφοριακών συστημάτων Νοσοκομείων.
- Ακολούθησε η εγκατάσταση εφαρμογών λογισμικού στα πλαίσια του Α΄ Κοινοτικού Πλαισίου Στήριξης (ΚΠΣ) ενώ με το Β΄ ΚΠΣ, δίνεται ιδιαίτερη έμφαση στην εκπαίδευση των λειτουργών της υγείας.



Εικόνα 2Ανάπτυξη ΠΣΝ στο Β' Κοινοτικό Πλαίσιο Στήριξης.

Τα βήματα προόδου που έγιναν στα 25 πρώτα νοσοκομεία ήταν μία αρχή αλλά υπήρχαν πολλές δυσκολίες. Με το Γ' ΚΠΣ έγινε προσπάθεια να βελτιώθεί το υπάρχον λογισμικό και να επεκταθεί η χρήση και οι λειτουργίες του και να προσαρμοστεί στο νέο μοντέλο των Ευρωπαϊκών συστημάτων υγείας. Όμως και πάλι διαπιστώθηκαν προβλήματα όπως καθυστερήσεις (διαγωνισμοί, υλοποίηση, παράδοση έργων) και στις περισσότερες περιπτώσεις τα έργα δεν παραδόθηκαν ή δεν τέθηκαν σε παραγωγική λειτουργία. Τα προβλήματα είχαν κυρίως να κάνουν με την πολυπλοκότητα των εφαρμογών και την εκπαίδευση και την προσαρμογή των χρηστών [8].

Σήμερα σχεδόν σε όλα τα κέντρα υγείας έχουν εγκατασταθεί πληροφοριακά συστήματα υγείας είτε ολοκληρωμένα είτε ορισμένα τμήματα διαχείρισης και στόχος είναι να εξοπλιστούν όλες οι μονάδες με ένα σύγχρονο και αξιόπιστο σύστημα. Αυτό όμως δεν αρκεί καθώς πρέπει και το προσωπικό του νοσοκομείου να είναι εξειδικευμένο με τις νέες τεχνολογίες και να μπορεί να τις αξιοποιεί στο έπακρον. Αυτό όμως είναι δύσκολο να συμβεί καθώς πολλοί ιατροί δεν είναι εξοικειωμένοι στην χρήση ηλεκτρονικών υπολογιστών.

1.2.2 Κριτήρια Επιλογής ΠΣΝ

Η επιλογή ενός ΠΣΝ δεν είναι καθόλου εύκολη υπόθεση. Πρέπει να ληφθούν υπόψη πολλά κριτήρια άλλα καίρια και άλλα λιγότερο σημαντικά. Στην συνέχεια θα δούμε σε ποια από αυτά πρέπει να επικεντρωθούμε.

1. Κύρια προϋπόθεση για να θεωρήσουμε οποιοδήποτε πληροφοριακό σύστημα αξιόπιστο και ασφαλές είναι να περιλαμβάνει όλα τα παρακάτω στοιχεία τα οποία έχουμε αναλύσει και προηγουμένως: Πιστοποίηση, Εξουσιοδότηση, Εμπιστευτικότητα, Ακεραιότητα, Μη δυνατότητα άρνησης συμμετοχής, Δυνατότητα ελέγχου, Ευθύνη, Διαφάνεια, Διαθεσιμότητα.
2. Διαλειτουργικότητα. Πολύ σημαντικός παράγοντας είναι η δυνατότητα του συστήματος να επικοινωνήσει με άλλα συστήματα υγείας για ανταλλαγή δεδομένων. Για παράδειγμα, εάν υπάρχει άλλο ηλεκτρονικό σύστημα που να εξάγει αποτελέσματα εξετάσεων σε κάποια άλλη ηλεκτρονική μορφή όπως "xml" αρχεία θα πρέπει το σύστημά να έχει την δυνατότητα να δέχεται αυτήν την πληροφορία και να την επεξεργάζεται αναλόγως.

3. Κόστος εφαρμογής όχι μόνο αγοράς αλλά και συντήρησης και του λογισμικού και του υλικού που απαιτείται για την λειτουργία του. Στις μέρες μας κάθε ευρώ έχει αξία και πρέπει να επενδύεται σωστά και όχι άσκοπα.
4. Αποδοτικότητα και αποτελεσματικότητα.Ο λόγος ύπαρξης των συστημάτων αυτών είναι η διευκόλυνση σε διάφορους τομείς χωρίς να μειώνεται η ποιότητα και η εγκυρότητα της υπηρεσίαςπου παρέχεται. Πρέπει λοιπόν να αξιολογηθεί το σύστημα από άτομα του χώρου της υγείας για να επιβεβαιωθεί ότι λειτουργεί σωστά και εξυπηρετεί τον σκοπό του σε όλους τους τομείς που απαιτείται.
5. Ευχρηστία. Άλλος ένας σημαντικός παράγοντας είναι η ευκολία χρήσης του περιβάλλοντος από τους υπαλλήλους. Ένα πρόβλημα που έχει παρουσιαστεί στα ελληνικά νοσοκομεία που χρησιμοποιούν πληροφοριακά συστήματα είναι η δυσκολία προσαρμογής των υπαλλήλων σε αυτά. Έτσι είναι απαραίτητο το σύστημα να είναι εύκολο στην χρήση ώστε οι υπάλληλοι να μπορούν να προσαρμοστούν εύκολα σε αυτό μετά την εκπαίδευσή τους η οποία μπορεί να γίνει με σεμινάρια και μαθήματα παρουσίασης καικάποια videosχρήσης του συστήματος για την κάθε ομάδα χρηστών.

1.3 Παρόμοια συστήματα

Υπάρχουν πολλές εταιρίες πληροφορικής στην Ελλάδα και το εξωτερικό που ασχολούνται αποκλειστικά με την δημιουργία και ανάπτυξη λογισμικού για συστήματα υγείας. Τα συστήματα αυτά μπορεί να είναι από απλά διαχειριστικά προγράμματα για μικρές κλινικές μέχρι ολοκληρωμένα συστήματα πληροφορικής για μεγάλα νοσοκομεία. Στο παρόν κεφάλαιο θα μελετήσουμε τέτοια συστήματα και θα δούμε τι μας παρέχουν και ποιες καινοτομίες διαθέτουν. Κάποιες από αυτές τις εταιρίες είναι η Intracom , η Datamed, η Apollokαι άλλες.

1.3.1 IntraHealth

HIntracom**Error! Reference source not found.**είναι μία από τις μεγαλύτερες εταιρίες λογισμικού στην Ελλάδα και δεν θα μπορούσε να μην διαθέτει λύσεις και για Πληροφοριακά Συστήματα Νοσοκομείου. Συγκεκριμένα έχει εγκαταστήσει Πληροφοριακά Συστήματα Νοσοκομείου στο Νοσοκομείο «Η Αγία Σοφία». Το Νοσοκομείο Παίδων «Η Αγία Σοφία», διαθέτει 36 κλινικές και 16 εργαστήρια ενώ η δυναμικότητά του φτάνει τις 600 κλίνες. Πρόκειται για το μεγαλύτερο νοσοκομείο Παίδων στην Ελλάδα και ένα από τα μεγαλύτερα στα Βαλκάνια. Στο εγκατεστημένο σύστημα θα είναι συνδεδεμένοι συνολικά περίπου 250 σταθμοί εργασίας του Νοσοκομείου. Το Σύστημα αυτό καλύπτει τις λειτουργίες του Οργανισμού οι οποίες αφορούν στον ιατρικό και νοσηλευτικό φάκελο ασθενούς, στην διοικητική και οικονομική δραστηριότητα του νοσοκομείου και στην διεκπεραίωση των εργαστηριακών εξετάσεων. Παράλληλα με την εφαρμογή του Ιατρικού Συστήματος και του Συστήματος Διαχείρισης Ασθενών μηχανογραφήθηκε ο Ιατρικός και ο Νοσηλευτικός Φάκελος, το Φαρμακείο, το Γραφείο Κινήσεως, το Λογιστήριο ασθενών, τα Εξωτερικά Ιατρεία και η Διαλογή (Triage) και, τέλος, η Διαδικτυακή πύλη (Portal) συνθέτοντας έτσι την εφαρμογή IntraHealth.

Όσον αφορά στο Πληροφοριακό Σύστημα Εργαστηρίων χρησιμοποιήθηκε το λογισμικό MediLABτης εταιρείας CCS,με την οποία συνεργάστηκεη INTRACOM, το οποίο ενσωματώθηκε με την εφαρμογή IntraHealth. Το συγκεκριμένο προϊόν αποτελεί το βασικό τροφοδότη του κλινικού ιατρικού φακέλουτου ασθενή σε σχέση με τις εργαστηριακές εξετάσεις, των οποίων οι παραγγελίες και τα αποτελέσματα ανταλλάσσονται με το υπόλοιπο Πληροφοριακό Σύστημα με αυτοματοποιημένο και διαφανή τρόπο.

Το Ολοκληρωμένο Σύστημα (IntraHealth) αποτελείται από πολλά υποσυστήματα. Κάποιααπόαυτάφαίνονται στον πίνακα 1.

Medical care / impatience	Nursing care
Delivery	X-ray treatment
Medical examinations	Anesthesiology

Renal desorder department	Oncology ratiotherapy
Medical stuff scheduling	Nursing staff scheduling
Outpatient units	Pharmacy materials management
Emergency outpatient units	Dietary
General leger	Contracts – suppliers - bids
Budget	Fixed assets
Hospital organisational chart	Cost accounting
Laboratory examinations	Stock management / maintenance
Personnel payroll	Cash flow
Secretary of outpatient units	Admission office
Patient accounting parameters	Cost and patient accounting
Hosptialisation charges of inpatient units	

Πίνακας 1 Υποσυστήματα του ΠΣΝ - IntraHealth της Intracom

Τεχνολογικά Χαρακτηριστικά του συστήματος:

- Web αρχιτεκτονική και τεχνικές τεχνολογίας n-tier**Error! Reference source not found.** στις βασικές εφαρμογές
- HL 7 Συμβατότητα
- Συνεργασία με τις περισσότερες βάσεις δεδομένων (Oracle, DB 2, SQL .)
- Βασισμένο σε τεχνολογία Java και Oracle

1.3.2 Medico

Το 'Medico'**Error! Reference source not found.**της Datamedέχει εφαρμοστεί σε πάνω από πενήντα δημόσια, πανεπιστημιακά, ιδιωτικά και στρατιωτικά νοσοκομεία, νοσοκομεία ασφαλιστικών ταμείων καθώς και σε ιδιωτικά νοσοκομεία (Ωνάσειο Καρδιοχειρουργικό Κέντρο) σε Ελλάδα και Κύπροπροσπαθώντας να καλύψει τις διαχειριστικές και ιατρικές τους απαιτήσεις.

Υποστηρίζει ότι διαθέτει επαρκεί ασφάλεια και έχει ευέλικτο σύστημα καθώς είναι κατανεμημένο σε υποσυστήματα τα οποία μπορούν να λειτουργήσουν πάνω από τα βασικά υποσυστήματα "Διαχείριση Ασθενών" και "Ιατρικό Υποσύστημα".Σημαντικό χαρακτηριστικό του είναι η δυνατότηταυλοποίησης ορισμένων μόνο υποσυστημάτων, ανάλογα με τις επιχειρησιακές ανάγκες κάθε εγκατάστασης και επιπλέον υποστηρίζει έξυπνες κάρτες (smartcards). Εκτός από τα βασικά υποσυστήματα διαθέτει τα εξής πρόσθετα υποσυστήματα:

- Υποσύστημα διαχείρισης χειρουργείων και αναισθησιολογικού τμήματος
- Υποσύστημα παρακολούθησης φυσιολογικών παραμέτρων ασθενών (ΜΕΘ, ΜΑΦ κλπ)
- Υποσύστημα διαχείρισης ακτινοδιαγνωστικού τμήματος (RIS)
- Υποσύστημα αρχειοθέτησης εγγράφων και εικόνων
- Υποσύστημα οργάνωσης νοσηλευτικού και ιατρικού προσωπικού
- Υποσύστημα Διαχείρισης Ιατρικού Αρχείου
- Υποσύστημα τεκμηρίωσης νοσηλευτικών υπηρεσιών
- Υποσύστημα υποστήριξης αποφάσεων διοίκησης (αποθήκευση δεδομένων - datawarehousing)

1.3.3 Concerto

HApollo**Error! Reference source not found.**είναι άλλη μία δυναμική εταιρία στον χώρο της ηλεκτρονικής ιατρικής και συνεργάζεται με σχεδόν 40 δημόσια νοσοκομεία της Αττικής και της περιφέρειας και σχεδόν 30 ιδιωτικές κλινικές.Ιδρύθηκε το 1995 και έκτοτε αναπτύσσει λογισμικό για

υπολογιστικά συστήματα όχι μόνο νοσοκομείων αλλά και γενικότερα εμπορικών εταιριών.Διαθέτει ιατρικές διαδικτυακές εφαρμογές και έχει δημιουργήσει ένα αξιόλογο νοσοκομειακό σύστημα για νοσοκομεία το 'ConcertoMedicalApplicationsPortal'.

Η σειρά πληροφοριακών συστημάτων υγείας που προωθεί υποστηρίζει:

- Κάρτα Υγείας με πολυσυλλεκτικό αντικείμενο υπηρεσιών.
- Τράπεζα Ιατρικού Φακέλου.
- Εγκαθίδρυση ενιαίου δικτύου ανταλλαγής ιατρικών δεδομένων από τους ιατρούς και τους φορείς που εμπλέκονται στο σύστημα
- Εγκαθίδρυση δικτύου ιατρικής διασύνδεσης σε επίπεδο επιστημονικής υποστήριξης
- Δημιουργία κέντρου αναφοράς διαγνωστικού και θεραπευτικού χαρακτήρα.
- Δημιουργία τμήματος επιστημονικής υποστήριξης και συνεχιζόμενης κατάρτισης του ιατρικού και νοσηλευτικού προσωπικού.
- Δημιουργία ειδικού τμήματος παρηγορητικής υποστήριξης ασθενών με χρόνια νοσήματα.

Επίσης διαθέτει μικρότερα πακέτα υπηρεσιών για μεμονωμένα ιατρεία και εργαστήρια με λειτουργίες που είναι αποκλειστικά για αυτά και δεν σχετίζονται με πολύπλοκα συστήματα νοσοκομείων.

1.3.4 Διεθνής χώρος

Στον διεθνή χώρο οι λύσεις που υπάρχουν είναι εκατοντάδες και απευθύνονται σε όλα τα μεγέθη και είδη νοσοκομειακών κέντρων και κέντρων υγείας. Το καθένα είναι προσαρμοσμένο στις ανάγκες της κάθε χώρας και στον τρόπο λειτουργίας των κέντρων. Κάποιες από αυτές τις εταιρίες είναι η BTBusinessSolutions[14], CTG[15], YASASII[16], Quanta[17], LHP[18]και πολλές άλλες. Λεπτομέρειες για τις τεχνολογίες που χρησιμοποιούν και τις καινοτομίες που διαθέτουν δεν είναι διαθέσιμες καθώς υπάρχει μεγάλος ανταγωνισμός και η κάθε εταιρία θέλει να εκμεταλλευτεί τα πλεονεκτήματά της. Σε γενικές γραμμές η κάθε εταιρία χρησιμοποιεί και διαφορετικές τεχνολογίες όπως για παράδειγμα κάποιοι χρησιμοποιούν Oracleβάση δεδομένων ενώ άλλοι MySQL, PostgreSQL, ομοίως και με τις γλώσσες προγραμματισμού όπου συναντάμε Java, .Net, C++ , Pythonκαι πολλές ακόμα. Το σημαντικό είναι να επιλέξουμε τεχνολογίες που να καλύπτουν τις απαιτήσεις μας και να είναι αρκετά γρήγορες στο να επεξεργάζονται μεγάλο όγκο δεδομένων χωρίς προβλήματα.

2. Μεθοδολογία Ανάπτυξης Λογισμικού

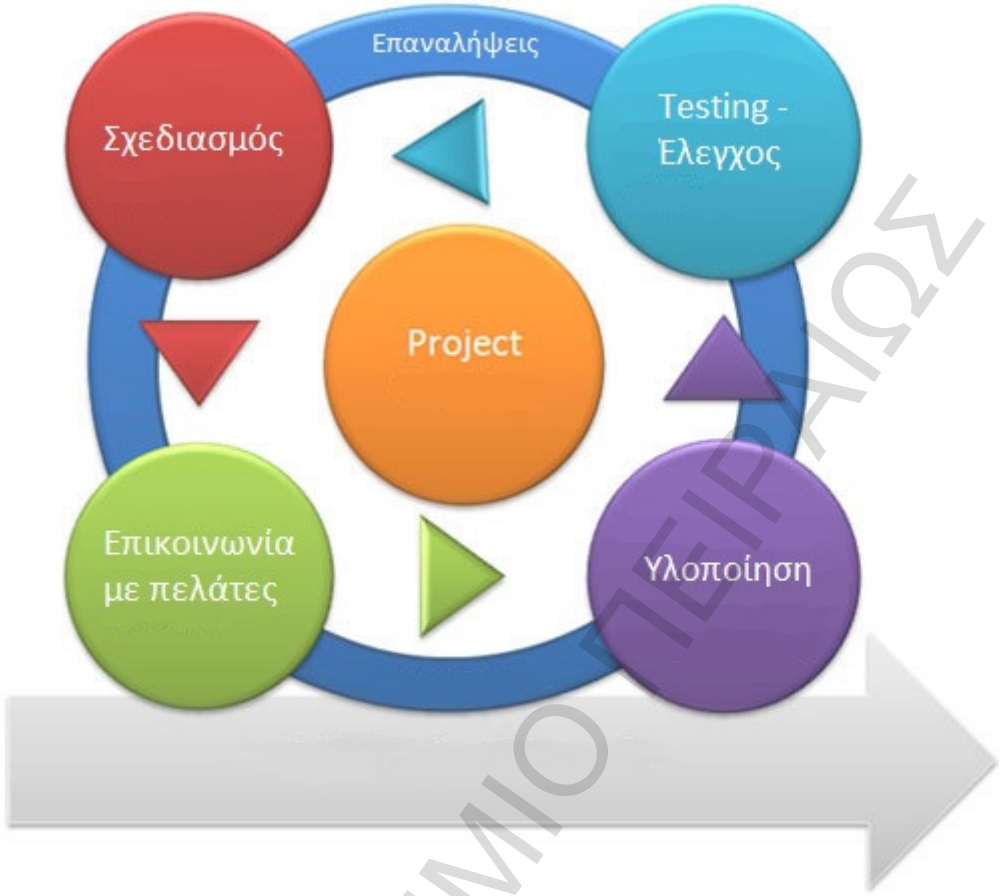
Μια σύγχρονη μεθοδολογία ανάπτυξης που χρησιμοποιείται στο WebApplicationDevelopment είναι αυτή του AgileSoftwareDevelopment. Η συγκεκριμένη μεθοδολογία δίνει ιδιαίτερη αξία στις ακόλουθες πτυχές κατά την ανάπτυξη του λογισμικού.

- **Τα άτομα και οι Αλληλεπιδράσεις** - στην Agile ανάπτυξη, η αυτο-οργάνωση και η ενθάρρυνση είναι σημαντικά στοιχεία, όπως και οι αλληλεπιδράσεις και η συστέγαση και ο προγραμματισμός σε ζευγάρια για καλύτερη συνεργασία.
- **Ανάπτυξη λογισμικού**–το λογισμικό το δουλεύουν κατά τμήματα και γρήγορα, έτσιθα είναι πιο χρήσιμοκαθώς θα μπορεί να το παρουσιαστεί στους πελάτες τμηματικά ώστε να βλέπουν αποτελέσματα κάτι που είναι πιο ευπρόσδεκτο από την απλή παρουσίαση εγγράφων και εικόνων.
- **Συνεργασία Πελατών**–οι απαιτήσεις δεν μπορούν να συλλέγονται πλήρως κατά την έναρξη του κύκλου ανάπτυξης του λογισμικού, ως εκ τούτου η συνεχήςσυμμετοχή των ενδιαφερομένων είναι πολύ σημαντική.
- **Αντίδραση σε αλλαγές** - η ευκίνητη ανάπτυξη επικεντρώνεται στην ταχεία αντιμετώπιση των αλλαγών και τη συνεχή ανάπτυξη.

Η Agile μέθοδοςσπάει τα καθήκοντα σε μικρά βήματα, με ελάχιστες απαιτήσεις σε σχεδιασμό, γιατί δεν αφορούν άμεσα τον μακροπρόθεσμο σχεδιασμό. Οι επαναλήψεις είναι σύντομες προθεσμίες (time-boxes) που συνήθως διαρκούν από μία έως τέσσερις εβδομάδες. Κάθε επανάληψη περιλαμβάνει μια ομάδα εργασίας που αναλαμβάνει έναν πλήρη κύκλο ανάπτυξης λογισμικού, και περιλαμβάνει σχεδιασμό, ανάλυση απαιτήσεων, κωδικοποίηση, έλεγχο μονάδων λογισμικού, και αποδοχή δοκιμών όταν το προϊόν εργασίας πλέον θα παρουσιαστεί στα ενδιαφερόμενα μέρη. Αυτό ελαχιστοποιεί το συνολικό κίνδυνο και επιτρέπει στο έργο να προσαρμοστεί στις αλλαγές γρήγορα. Οι ενδιαφερόμενοι φορείς προσκομίζουν τις προδιαγραφές, όπως απαιτείται. Η επανάληψη δεν μπορεί να προσθέσει αρκετή λειτουργικότητα στο λογισμικό ώστε να δικαιολογούν την απελευθέρωση του στην αγορά, αλλά ο στόχος είναι να υπάρχει μια διαθέσιμη έκδοση (με ελάχισταbugs) στο τέλος κάθε επανάληψης. Πολλαπλές επαναλήψεις μπορεί να χρειαστούν για την κυκλοφορία ενόςπροϊόντος ή νέων χαρακτηριστικών.

Δώδεκα αρχές που διέπουν την Agileμεθοδολογία ανάπτυξης είναι οι ακόλουθες:[19]

- Η ικανοποίηση των πελατών από την ταχεία παράδοση χρήσιμου λογισμικού.
- Αλλαγές σε απαιτήσεις είναι καλοδεχούμενες, ακόμη και αργά στην ανάπτυξη.
- Το λογισμικό που αναπτύσσεται παραδίδεται συχνά (εβδομάδες και όχι μήνες).
- Το λογισμικόπου αναπτύσσεται είναι το κύριο μέτρο της προόδου.
- Η βιώσιμη ανάπτυξη, είναι σε θέση να διατηρήσει ένα σταθερό ρυθμό.
- Στενή, καθημερινή συνεργασία μεταξύ των ανθρώπων των επιχειρήσεων και των προγραμματιστών.
- Η πρόσωπο με πρόσωπο συνομιλία είναι η καλύτερη μορφή επικοινωνίας (colocation).
- Τα έργα λογισμικού είναι χτισμένα γύρω από δραστήρια πρόσωπα, τα οποία θα πρέπει να είναι αξιόπιστα.
- Συνεχής έμφαση στην τεχνική αριστεία και καλό σχεδιασμό.
- Απλότητα.
- Αυτο-οργάνωση ομάδων.
- Η τακτική προσαρμογή στις μεταβαλλόμενες συνθήκες.



Εικόνα 3Agileανάπτυξη λογισμικού[20]

Από τα παραπάνω που αφορούν την μεθοδολογία AgileDevelopment παρατηρούμε ότι ξεφεύγει από το πιο αυστηρό και δύσκαμπτό παραδοσιακό μοντέλο τύπου «καταρράκτη» και κάνει πιο ευέλικτη την προσαρμογή στις αλλαγές των αναγκών του πελάτη ενώ παράλληλα επιταχύνει τα timelinesχωρίς να θίγει την ποιότητα που παραγόμενου λογισμικού αλλά να την προσαρμόζει καλυτέρα στις απαιτήσεις των ενδιαφερόμενων. Για αυτούς τους λογούς θεωρήσαμε αυτήν την μεθοδολογία την καλύτερη για να την υιοθετήσουμε και να βασίσουμε τα στάδια ανάπτυξης του συστήματός μας.

Για να εφαρμοστεί η παραπάνω μεθοδολογία πρέπει να την προσαρμόσουμε στις δικές μας ανάγκες και συνθήκες. Στην περίπτωσή μας δεν έχουμε ομάδες εργασίας για να υλοποιήσουμε τα διάφορα τμήματα του λογισμικού οπότε ναι μεν χωρίζουμε το λογισμικό μας σε πολλά μικρά κομμάτια, αλλά αυτά θα υλοποιηθούν από ένα άτομο. Κατά την υλοποίηση υπάρχει συνεχής επαφή με άτομα που χρησιμοποιούν παρόμοια συστήματα και δουλεύουν στον χώρο της υγείας σε διάφορα νοσοκομεία.Στην συνέχεια κάθε τμήμα που υλοποιείται ελέγχεται και γίνονται οι απαραίτητες διορθώσεις εάν χρειαστούν. Έτσι βήμα-βήμα θα φτάσουμε στο τελικό αποτέλεσμα που θα είναι και η ολοκληρωμένη εφαρμογή μας.

2.1 Ανάλυση απαιτήσεων

Το πρώτο στάδιο πριν την υλοποίηση οποιουδήποτε έργου είτε πληροφοριακού είτε όχι είναι η ανάλυση των απαιτήσεων. Πρέπει δηλαδή να προσδιορίσουμε τις ανάγκες ενός νοσοκομείου τις οποίες θα επιχειρήσουμε να καλύψουμε με το σύστημα που θα δημιουργήσουμε. Για να συγκεντρώσουμε όλες αυτές τις πληροφορίες απευθυνθήκαμε σε άτομα του χώρου της υγείας που εργάζονται σε νοσοκομεία και χρησιμοποιούν παρόμοια συστήματα ώστε να μπορέσουμε να υλοποιήσουμε τα τμήματα που αυτοί θεωρούν σημαντικά και χρήσιμα και να βελτιώσουμε κάποια ελαττωματικά και δυσλειτουργικά κομμάτια. Παρακάτω θα παραθέσουμε ταξινομημένα ανάλογα με το τμήμα τις απαιτήσεις που

συγκεντρώθηκαν μετά από πολλές συναντήσεις και αναλύσεις με τα αρμόδια άτομα του χώρου της υγείας.

Το σύστημά μας πρέπει να διαθέτει ομάδες χρηστών οι οποίες θα έχουν συγκεκριμένες αρμοδιότητες και δικαιώματα. Οι ομάδες που θα χρειαστούμε εμείς είναι διαχειριστές, ιατροί, φαρμακοποιοί, νοσοκόμοι-νοσηλευτές, εργαστηριακοί ιατροί και ασθενείς. Παρακάτω θα αναφέρουμε τις δυνατότητες που θα έχει η κάθε ομάδα χρηστών.

Γενικές λειτουργίες για όλους τους χρήστες:

- Εγγραφή. Κάθε χρήστης πρέπει να έχει δικαίωμα να κάνει αίτηση για δημιουργία νέου λογαριασμού συμπληρώνοντας μία φόρμα με τα στοιχεία του.
- Σύνδεση - Αυθεντικοποίηση. Κάθε χρήστης θα μπορεί να συνδεθεί με το σύστημα χρησιμοποιώντας όνομα χρήστη και κωδικό που έχει ο ίδιος ορίσει. Ο κωδικός πρέπει να είναι κωδικοποιημένος για μεγαλύτερη ασφάλεια.
- Ασφάλεια. Το σύστημά μας πρέπει να είναι ασφαλές και η κάθε ομάδα χρηστών να έχει πρόσβαση μόνο στα δεδομένα και στοιχεία που τους αφορούν.
- Προφίλ. Κάθε εγγεγραμμένος χρήστης πρέπει να διαθέτει προφίλ με τα προσωπικά του στοιχεία στα οποία έχει πρόσβαση και δυνατότητα τροποποίησης.

Διαχειριστής:

- Ενεργοποίηση νέων χρηστών. Οι νέοι χρήστες (ιατροί, φαρμακοποιοί, νοσηλευτές, εργαστηριακοί ιατροί) αφού κάνουν αίτηση για νέο λογαριασμό πρέπει να ενεργοποιηθούν από έναν διαχειριστή του συστήματος μέσω της μεθόδου τις αυθεντικοποίησης ή να απορριφτούν μέσω τις διαγραφής.
- Διαγραφή ή Κλείδωμα λογαριασμών. Θα πρέπει να έχει την δυνατότητα να διαγράφει ή να κλειδώνεικάποιον λογαριασμό που έχει αυθεντικοποιηθεί και είναι σε λειτουργία.

Ιατρός:

- Δημιουργία φακέλου ασθενή. Κατά την εισαγωγή του ασθενή ο ιατρός πρέπει να έχει την δυνατότητα να δημιουργεί ηλεκτρονικό φάκελο για τον ασθενή ή σε περίπτωση που υπάρχει ήδη φάκελος θα προστίθεται νέα εισαγωγή ασθενή στον υπάρχοντα φάκελο με την ημερομηνία εισαγωγής.
- Επεξεργασία φακέλου ασθενή. Ο Ιατρός πρέπει να είναι ο μόνος αρμόδιος που θα έχει πρόσβαση στον πλήρη φάκελο του ασθενή και θα μπορεί να επεξεργάζεται και να διορθώνει το περιεχόμενό του.
- Πληροφορίες φακέλου ασθενή. Ένας φάκελος ασθενή πρέπει να διαθέτει προσωπικά στοιχεία, στοιχεία σύνδεσης με το σύστημα, ιστορικό, διάγνωση ιατρού, προτεινόμενη θεραπεία από τον ιατρό, οδηγίες προς τους νοσοκόμους-νοσηλευτές, φαρμακευτική αγωγή, εργαστηριακές εξετάσεις και πληροφορίες για την αιτία εισαγωγής, αρμόδιο ιατρό, κλίνη που νοσηλεύεται και ημερομηνία εισαγωγής και εξαγωγής. Όλα αυτά φυσικά πρέπει να διατίθενται για κάθε νέα εισαγωγή του ασθενή στο νοσοκομείο και πρέπει να υπάρχει η δυνατότητα εκτύπωσης όλων των παραπάνω.
- Αναζήτηση ασθενή. Πρέπει να έχει δυνατότητα αναζήτησης ασθενή στην βάση με βάση το ονοματεπώνυμο για να δει και να επεξεργαστεί το φάκελό του.
- Διάγνωση. Μία προαιρετική λειτουργία που θα μπορούσε να έχει είναι ένα βοηθητικό σύστημα διάγνωσης ασθένειας. Με αυτό δίνεται η δυνατότητα στους ιατρούς να προσθέσουν ασθένειες και συμπτώματα ασθενειών στην βάση και να τα συνδυάσουν μεταξύ τους. Έπειτα κάνοντας μία απόπειρα διάγνωσης με βάση κάποια συμπτώματα το σύστημα θα παρουσιάζει πιθανές ασθένειες που θα βρει στην βάση του που να ταιριάζουν με τα συμπτώματα που έχουν εισαχθεί.

Φαρμακοποιός:

- Διαχείριση φαρμάκων. Ο Φαρμακοποιός πρέπει να διαχειρίζεται τα φάρμακα του νοσοκομείου, να έχει την δυνατότητα να προσθέτει και να αφαιρεί καινούρια φάρμακα στην λίστα των φαρμάκων και να τροποποιεί τις ποσότητές τους.

- Αποστολή φαρμάκων. Επίσης είναι υπεύθυνος για την αποστολή φαρμάκων σε ασθενείς.Πρέπει δηλαδή μόλις έρθει κάποια συνταγή από τον ιατρό (ηλεκτρονικά μέσω του συστήματος) να βρει τα φάρμακα και να τα δώσει στον αρμόδιο νοσοκόμο. Η ποσότητα των φαρμάκων που στάλθηκαν πρέπει αυτόματα να ανανεώνεται στην βάση ώστε να είναι γνωστή πάντα ηποσότητα που απομένει.

Νοσοκόμος Νοσηλευτής:

- Αναζήτηση ασθενή. Η ευθύνη που έχει ο νοσοκόμος είναι να φροντίζει τους ασθενείς.Έτσι πρέπει να έχει πρόσβαση στον ηλεκτρονικό φάκελο του ασθενή και να μπορεί να δει τα απαραίτητα στοιχεία που του χρειάζονται και τις οδηγίες του ιατρού για την φροντίδα η την φαρμακευτική αγωγή του ασθενή.

Εργαστηριακός Ιατρός:

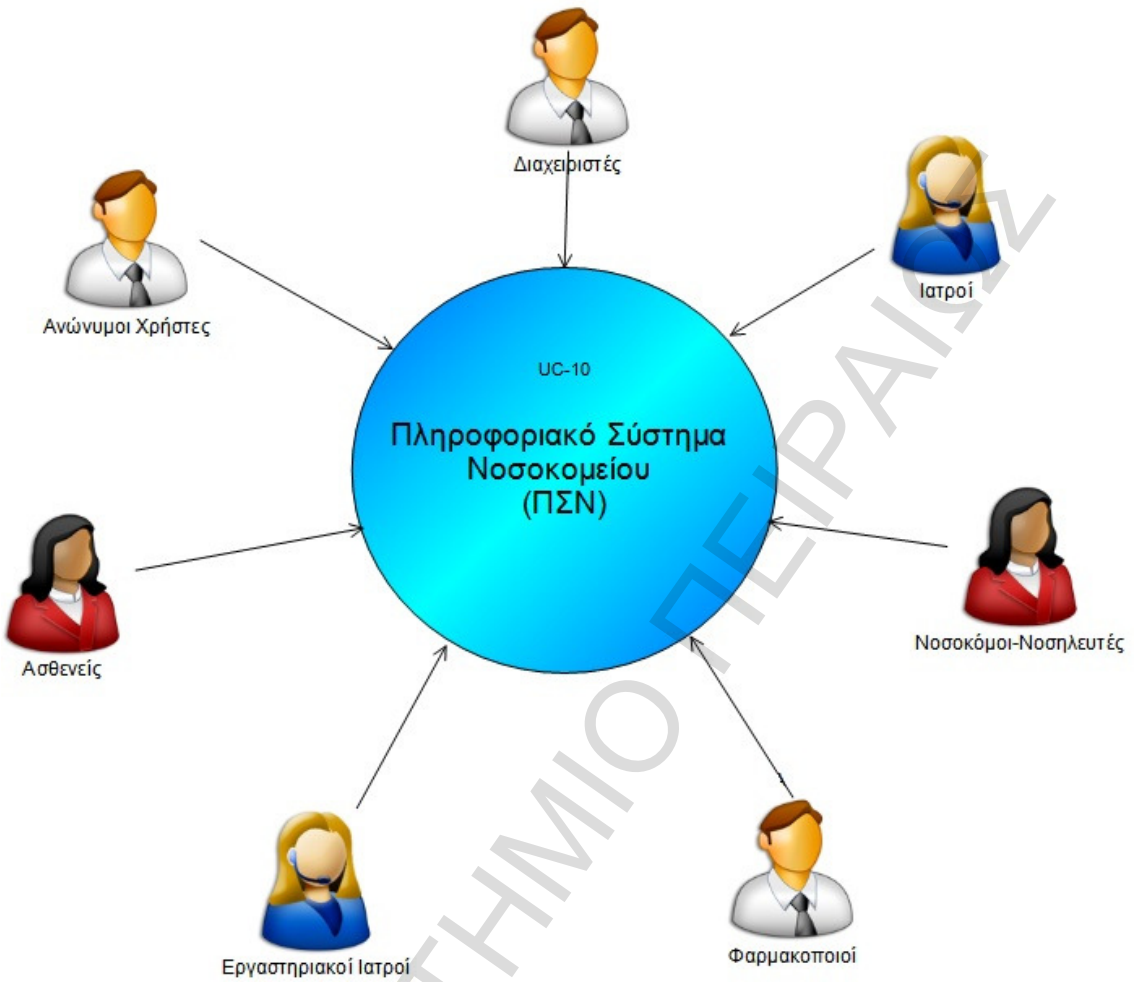
- Καταχώριση εργαστηριακών αποτελεσμάτων. Ο εργαστηριακός ιατρός πρέπει να έχει πρόσβαση σε μία λίστα με όλες τις εργαστηριακές εξετάσεις που πρέπει να γίνουν για τους ασθενείς του νοσοκομείου με την ημερομηνία πραγματοποίησής τους. Αφού γίνουν οι εξετάσεις τα αποτελέσματα πρέπει να περαστούν στο σύστημα για να τα λάβει ο ιατρός.

Ασθενής:

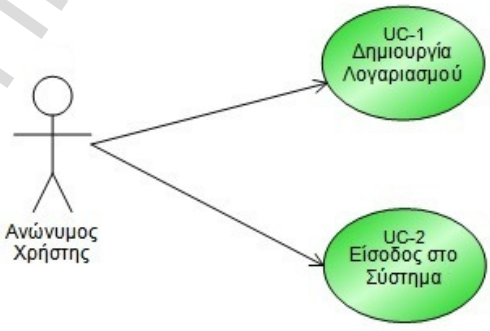
- Πρόσβαση στο ιστορικό του. Ο Ασθενής πρέπει να έχει πρόσβαση στο φάκελό του και να βλέπει όποια στοιχεία επιτρέπονται.

2.2 Διαγράμματα χρήσηςUML

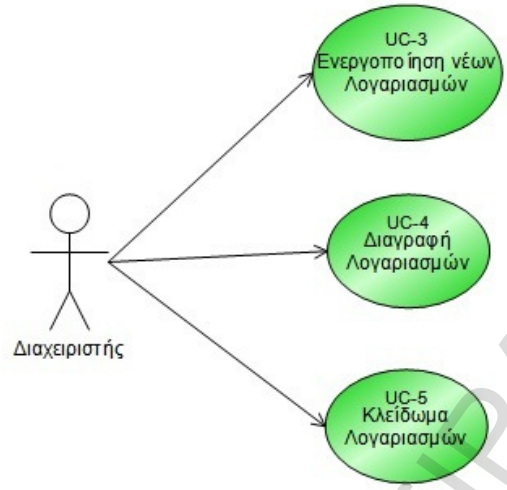
Παρακάτω θα παρουσιάσουμε τα διαγράμματα χρήσηςUML που προκύπτουν από την ανάλυση των απαιτήσεων που κάναμε προηγουμένως. Στην εικόνα 4 βλέπουμε τις ομάδες των χρηστών που αλληλοεπιδρούν με το σύστημα ενώ στις υπόλοιπες εικόνες 5α-5ζ βλέπουμε τις δυνατότητες που έχει η κάθε ομάδα χρηστών. Στην εικόνα 6 έχουμε τα περιεχόμενα ενός φακέλου ασθενή όπου μπορούμε να δούμε τις ιατρικές πληροφορίες που μπορούν να καταγραφούν στο σύστημά. Τέλος έχουμε και την εικόνα 7 όπου παρουσιάζουμε την χρήση της λειτουργίας διάγνωσης ασθενειών που διαθέτουμε στην εφαρμογή.



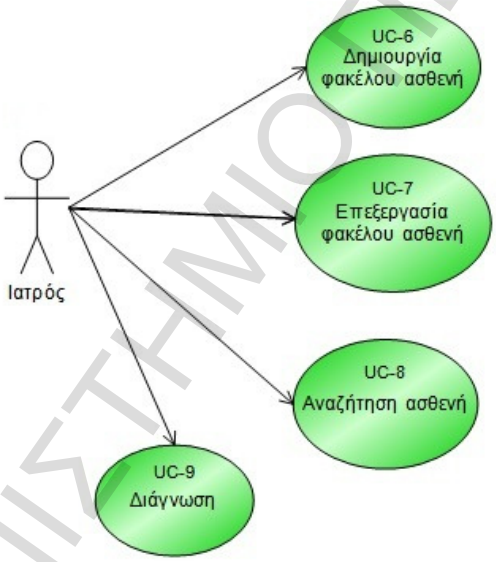
Εικόνα 4Διάγραμμα χρηστών του Συστήματος



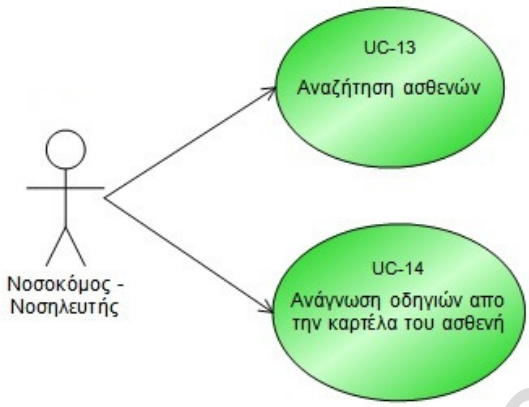
Εικόνα 5α Διάγραμμα χρήσης ανώνυμου χρήστη



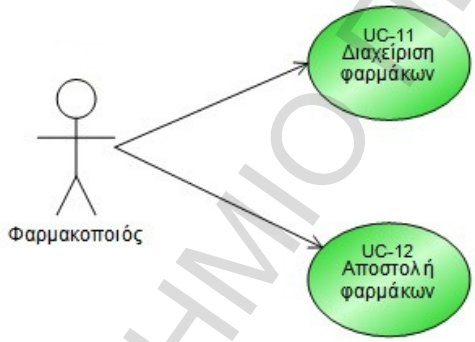
Εικόνα 5β Διάγραμμα χρήσης Διαχειριστή



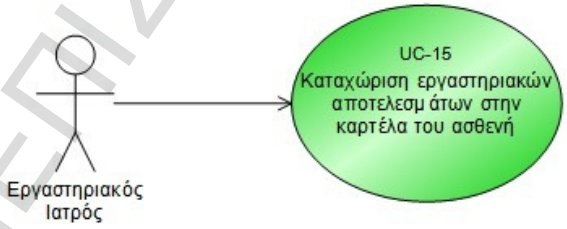
Εικόνα 5γ Διάγραμμα χρήσης Ιατρού



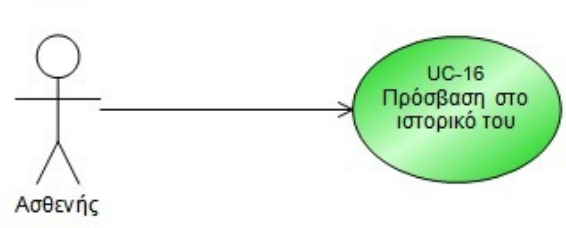
Εικόνα 5δ Διάγραμμα χρήσης Νοσοκόμου - Νοσηλευτή



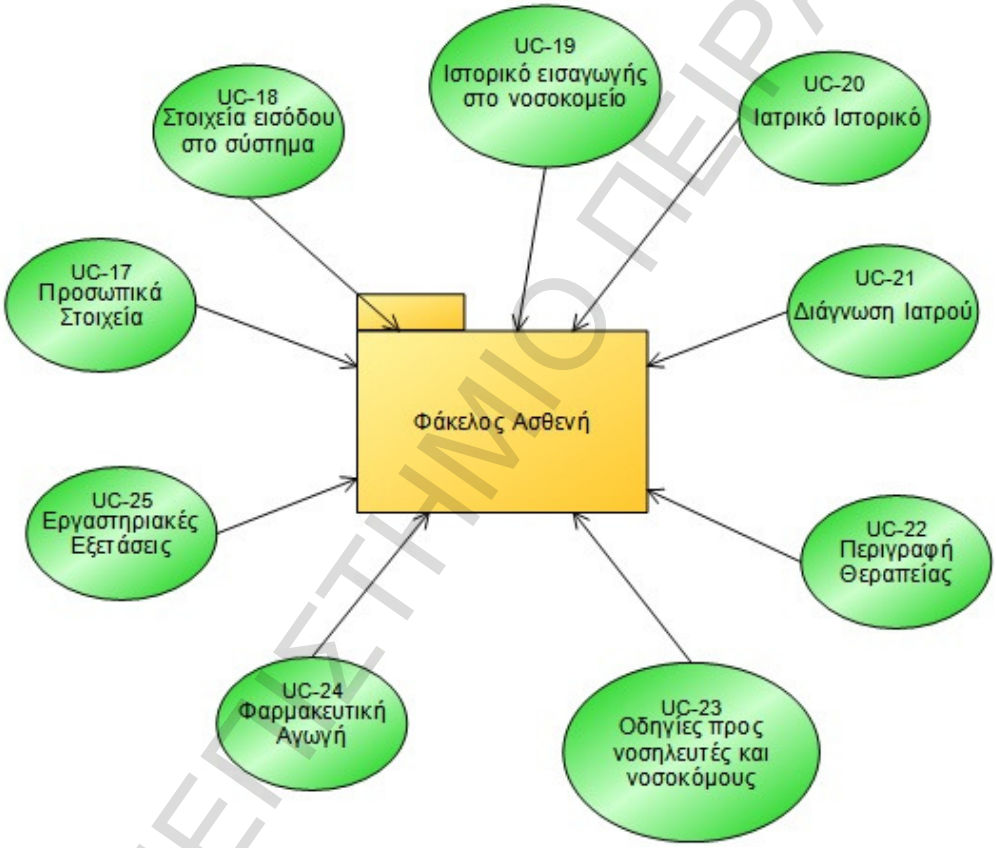
Εικόνα 5ε Διάγραμμα χρήσης Φαρμακοποιού



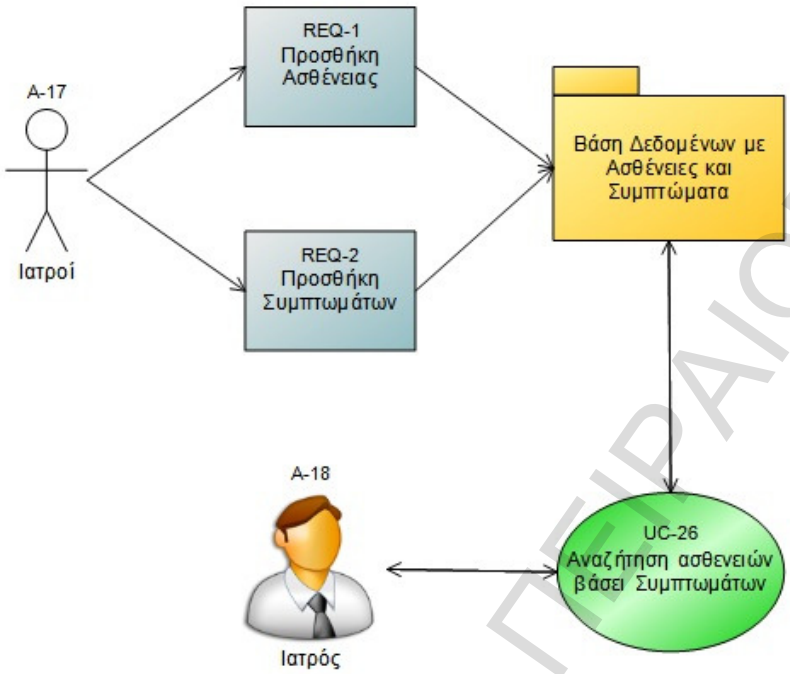
Εικόνα 5στ Διάγραμμα χρήσης ΕργαστηριακούΙατρού



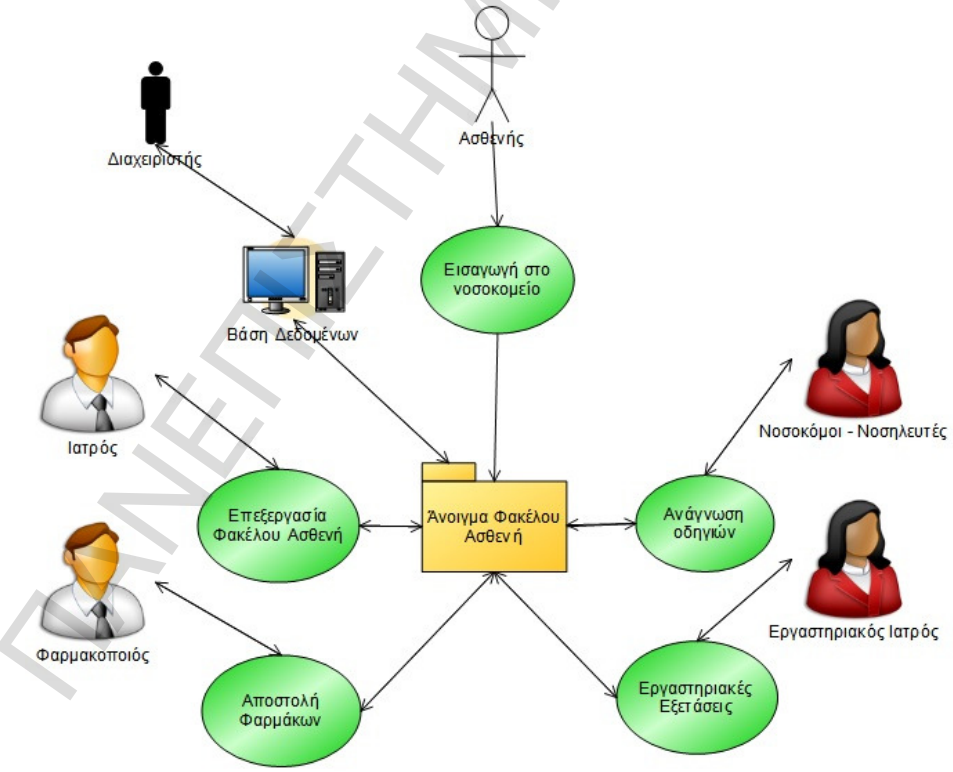
Εικόνα 5ζ Διάγραμμα χρήσης Ασθενή



Εικόνα 6Διάγραμμα χρήσης Φακέλου Ασθενή



Εικόνα 7Διάγραμμα Διάγνωσης



Εικόνα 8Διάγραμμα Χρήσης όλων των χρηστών

Τέλος στην εικόνα 8 έχουμε ένα συνοπτικό διάγραμμα χρήσης του συστήματος το οποίο δείχνει την αλληλεπίδραση του συστήματος με τους χρήστες και τον ασθενή που εισέρχεται στο νοσοκομείο. Μελετώντας την ανάλυση απαιτήσεων που κάναμε και βαδίζοντας με βάση τα διαγράμματαUML που σχεδιάσαμε θα υλοποιήσουμε το σύστημά και θα προσπαθήσουμε να καλύψουμε όλες τις ανάγκες που έχουμε περιγράψει.

3. Το προτεινόμενο σύστημα

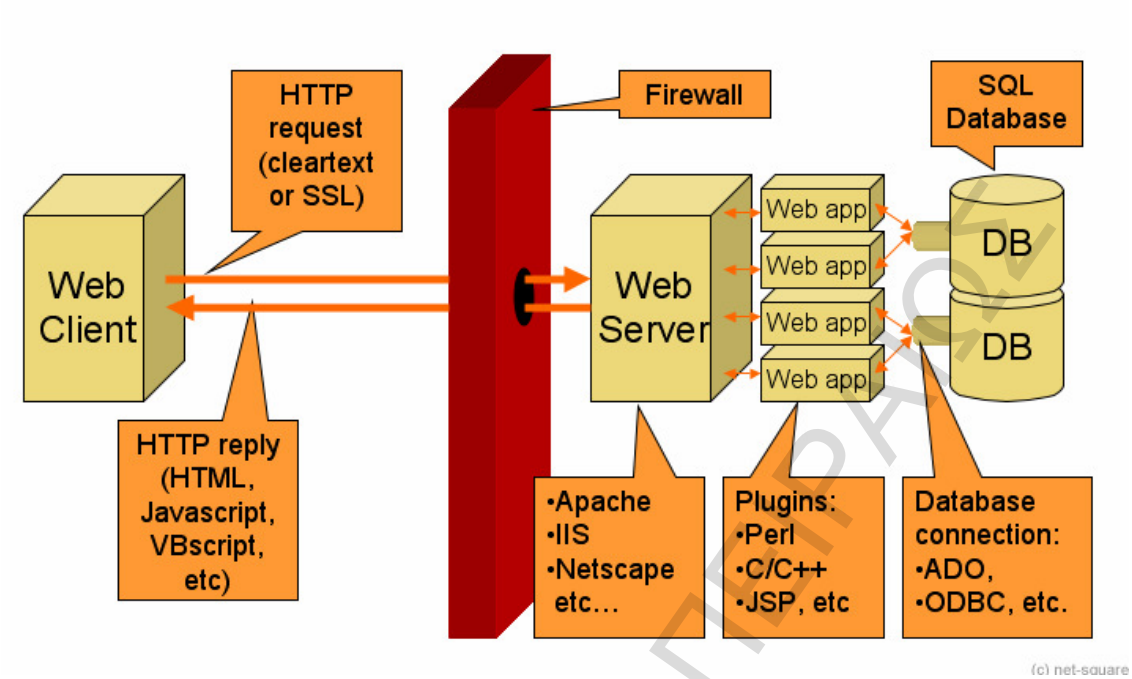
3.1 Η Προσέγγισή της Διατριβής

Ο τρόπος με τον οποίο θα προσεγγίσουμε την υλοποίηση της εφαρμογής μας βασίζεται στο μοντέλο που παρουσιάσαμε στο κεφάλαιο 2 «Μεθοδολογία ανάπτυξης λογισμικού». Έτσι θα εκτελούμε 4 βήματα τα οποία θα επαναλαμβάνουμε μέχρι να φτάσουμε στο τελικό μας αποτέλεσμα. Ξεκινώντας θα κάνουμε έναν αρχικό σχεδιασμό και ένα πλάνο για το τι θα υλοποιήσουμε και με ποια σειρά. Μετά θα γίνεται επικοινωνία με άτομα του χώρου τις ιατρικής που χρησιμοποιούν παρόμοια συστήματα για διευκρινίσεις και διορθώσεις πάνω στον σχεδιασμό μας και τις απαιτήσεις και λειτουργίες που θα υλοποιήσουμε. Στην συνέχεια θα σχεδιάσουμε το κομμάτι τις εφαρμογής που αναλύσαμε προηγουμένως χρησιμοποιώντας τις τεχνολογίες που έχουμε επιλέξει για να υλοποιήσουμε την εφαρμογή μας. Τέλος θα ελέγξουμε το τμήμα που μόλις ολοκληρώσαμε για να δούμε εάν χρειάζεται κάποιες διορθώσεις και να επιβεβαιώσουμε ότι λειτουργεί όπως το σχεδιάσαμε και όπως το θέλουμε εμείς. Το επόμενο βήμα είναι να ξεκινήσουμε την διαδικασία από την αρχή προγραμματίζοντας και σχεδιάζοντας το επόμενο τμήμα που θα υλοποιήσουμε και λειτουργώντας με παρόμοιο τρόπο ολοκληρώνουμε όλα τα μέρη του συστήματός μας και έχουμε το τελικό αποτέλεσμα. Φυσικά οποιαδήποτε στιγμή μπορούμε να προσθέσουμε ή να τροποποιήσουμε κομμάτια τις εφαρμογής μας λειτουργώντας με όμοιο τρόπο.

Μεγάλη έμφαση θα δώσουμε στην επικοινωνία με άτομα του χώρου της ιατρικής που χρησιμοποιούν παρόμοια συστήματα. Έτσι θα επωφεληθούμε από την εμπειρία τους και θα μάθουμε σε ποια στοιχεία πρέπει να δώσουμε έμφαση ώστε οι μελλοντικοί χρήστες της εφαρμογής να μην δυσκολευτούν στην χρήση της, αντιθέτως να την βρουν εύχρηστη και λειτουργική. Έτσι παράλληλα με την δημιουργία της εφαρμογής τα άτομα αυτά ελέγχουν τα ολοκληρωμένα τμήματα του συστήματος και αξιολογούν τις λειτουργίες τους παρατηρώντας παράλληλα δυσλειτουργίες και λάθη τα οποία διορθώνονται στην συνέχεια.

3.2 Τεχνική Περιγραφή της Εφαρμογής

Η εφαρμογή που υλοποιήθηκε στο πλαίσιο της παρούσας μεταπτυχιακής διατριβής αφορά ένα Πληροφοριακό Σύστημα Νοσοκομείου (ΠΣΝ). Είναι μία διαδικτυακή εφαρμογή (webapplication) και όχι εκτελέσιμηεφαρμογή για “Windows” οπότε δεν έχουμε περιορισμούς στις συσκευές που μπορούν να έχουν πρόσβαση σε αυτήν. Μπορούμε να χρησιμοποιήσουμε τον επιτραπέζιο υπολογιστή μας, τον φορητό υπολογιστή μας (laptop), κάποιο κινητό τηλέφωνο ή έναν υπολογιστή ταμπλέτα ανεξαρτήτως λειτουργικού που διαθέτουν. Θα χρειαστούμε μόνο πρόσβαση σε τοπικό δίκτυο ή στο διαδίκτυο και ένα οποιοδήποτε πρόγραμμα περιήγησης (webbrowser) που να υποστηρίζει ‘javascript’.



Εικόνα 9 Διαδικασία αποστολής δεδομένων σε μια διαδικτυακή εφαρμογή[21]

Στην εικόνα 9 μπορούμε να δούμε την ροή των δεδομένων που ανταλλάσσονται ανάμεσα στην συσκευή μας μέσω του προγράμματος περιήγησης (WebClient) και τουδιακομιστή(WebServer). Όπως βλέπουμε η διαδικασία ξεκινάει με την αποστολή ενός HTTPrequest, είτε σαν απλό κείμενο είτε σε κάποια ασφαλή SSLγραμμή. Το πακέτο αυτό αφού περάσει από το τείχος προστασίας (firewall) καταλήγει στον WebServermας ο οποίος διαθέτει κάποιον webserver (Apache, iss …) εκεί θα υποδεχτούμε το requestκαι θα αναλυθεί το είδος του πακέτου και θα προωθηθεί στην κατάλληλη διαδικτυακή εφαρμογή (webapp) που θα το εξυπηρετήσει. Σε έναν webservermπορεί να υπάρχουν πολλές εφαρμογές για διάφορες λειτουργίες και διαδικασίες. Το νοσοκομειακό σύστημα που μελετάμε είναι και αυτό ένα ‘webapp’. Κάποιες από τις πιο γνωστές γλώσσες προγραμματισμού που μπορούν να χρησιμοποιηθούν για την δημιουργία ενός webappliciationείναι Java,JSP,VB.Net, C#, Php, Python,Ruby και άλλες που χρησιμοποιούνται λιγότερο.

Στην δική μας περίπτωση χρησιμοποιούμε JAVAγια να επεξεργαστούμε τα δεδομένα και JSPκυρίως για να διαμορφώσουμε την τελική ιστοσελίδα που θα προβάλουμε. Έτσι λοιπόν αφού το πακέτο φτάσει στο τμήμα του webappαυτό αναλύεται και εκτελείται το αντίστοιχο κομμάτι κώδικα που ζητήθηκε. Η εκτέλεση αυτού του κώδικα μπορεί να επιφέρει πολλά και ποικίλα αποτελέσματα, από το πιο απλό το να ανοίξουμε δηλαδή μία στατική σελίδα έως το πιο πολύπλοκο που περιλαμβάνει επικοινωνία με μία βάση δεδομένων(MySQL, Oracle, MicrosoftSQLServer …)επεξεργασία των δεδομένων και διαμόρφωση των δεδομένων που θα σταλθούν πίσω στον ‘webclient’. Μόλις η απάντηση είναι έτοιμη αυτή προωθείται στονwebservero οποίος με την σειρά του την μεταβιβάζει πίσω στον χρήστη που ξεκίνησε την επικοινωνία. Στην μεριά του χρήστη πλέον με την βοήθεια της HTML που ορίζει την δομή της σελίδας, της CSSπου διαχειρίζεται την εμφάνιση και της JavaScriptπου βοηθάει στην δομή, στην εμφάνιση τον έλεγχο και ακόμα και στην επεξεργασία κάποιων τελικών δεδομένων, έχουμε το τελικό αποτέλεσμα που θα εμφανιστεί στον περιηγητή του χρήστη. Η διαδικασία αυτή ακολουθείται για κάθε είδους requestπου κάνουμε στον servermας.

3.3 Τεχνολογίες που Χρησιμοποιήθηκαν

Οι επιλογές τεχνολογιών για την υλοποίηση μίας διαδικτυακής εφαρμογής είναι πάρα πολλές και η κάθε μία έχει τα πλεονεκτήματα και τα μειονεκτήματά της. Παρακάτω θα παρουσιάσουμε τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της δικής μας εφαρμογής και τα πλεονεκτήματα που μας δίνουν. Η

επιλογή πολλών εξ αυτόν έγινε με σκοπό την εκμάθηση της εκάστοτε τεχνολογίας καθώς θεωρούνται εμπορικές, χρησιμοποιούνται ευρέως και αναπτύσσονται και βελτιώνονται συνεχώς.

Για το clientsideκομμάτι της εφαρμογής, που ορίζει το περιβάλλον χρήσης του χρήστη (UserInterface) και εκτελείται αποκλειστικά στο πρόγραμμα περιήγησης (webbrowser) μας, χρησιμοποιήσαμε:

- HTML για να ορίσουμε τα δομικά στοιχεία που χρειαζόμαστε στην σελίδα μας.
- CSSγια να διαμορφώσουμε την εμφάνιση τις σελίδας ώστε να είναι απλή και ευχάριστη.
- JavaScriptγια να δημιουργήσουμε πιο σύνθετα δομικά στοιχεία όπως πίνακες, μενού. αναδυόμενα παράθυραακόμα και κάποιους μικρούς ελέγχους και απλές πράξεις που χρειάστηκαν μέσα στην εφαρμογή μας.

Ο ServerSideτις εφαρμογής μας είναι το κομμάτι που εκτελείται στον servermας και το οποίο κάνει όλες τις πολύπλοκες διεργασίες όπως την ανάκτηση δεδομένων από την βάση, την επεξεργασία των δεδομένων για έλεγχο ασφάλειας και γενικότερά όλες τις εσωτερικές λειτουργίες του συστήματός μας. Εδώ χρησιμοποιήθηκαν οι παρακάτω τεχνολογίες:

- JSP (JavaServerPages), βασίζεται σε τεχνολογία Javakαι προτιμήθηκε μιας και η εφαρμογή μας θα λειτουργήσει κυρίως με Java και θα έχουμε καλύτερα συνεργασία μεταξύ τους και μας βοηθάει να εμφανίσουμε τα δεδομένα που λαμβάνουμε από τον server και να κάνουμε κάποιες απλές τροποποιήσεις εάν χρειαστεί. Όλες οι σελίδες που περιέχουν τον κώδικα για το Front–End κομμάτι δηλαδή αυτό που βλέπει ο χρήστης έχουν κατάληξη.jsp. Ένα από τα χαρακτηριστικά μίας σελίδας JSPείναι ότι γίνεται ‘compiled’σε ‘Javaservlet’ και στην ουσία δημιουργούμε μία Javaclass η οποία βασίζεται στο JavaServlet API**Error! Reference source not found.**δηλαδή είναι ένα αντικείμενο(‘object’)το οποίο μπορούμε να καλέσουμε με ένα ‘request’. Το ‘Javaservlet’ θα επεξεργαστεί τα δεδομένα θα διεξαγάγει τις κατάλληλες διεργασίες και θα εξαγάγει το αποτέλεσμα. Έτσι μπορούμε να εκμεταλλευτούμε τηντεχνολογία και την ταχύτητα που μας προσφέρει η Java.Τα ‘servlets’ αυτά είναι ο τρόπος με τον οποίο ηjavaδημιουργεί δυναμικές σελίδες όπως αντίστοιχα κάνουν η Phpκαι η ASP.NET.
- JavaEE 6 (EnterpriseEdition)**Error! Reference source not found.**είναι προγραμματιστική γλώσσα για δικτυακές και διαδικτυακές εφαρμογές. Είναι κατάλληλη για πολύπλοκες και μεγάλες εφαρμογές και θεωρείται πολύ γρήγορη στην εκτέλεση. Μπορεί να εκτελεστεί σε οποιαδήποτε πλατφόρμα βρίσκεται ο server μας (Windows, Linux κτλ.) με αποτέλεσμα να κάνει την εφαρμογή μας πιο ευέλικτη και φυσικά ό,τι λογισμικού χρειαστούμε για την υλοποίηση της εφαρμογής μας είναι και δωρεάν και opensource(π.χ. Eclipse, Linux, Apache HTTP, Mysql). Επίσης διαθέτει πολλά πλαίσιατα οποία μας βοηθάνε στην υλοποίηση του συστήματος μας κάνοντας πιο εύκολες κάποιες διαδικασίες που χωρίς αυτά θα απαιτούνταν περισσότερος χρόνος και θα υπήρχαν περισσότερες πιθανότητες για λάθη.
- MySQLπρόκειται για μία από τις πιο δημοφιλείς σχεσιακές βάσεις δεδομένων (relationaldatabasemanagementsystem RDBMS)**Error! Reference source not found.** που υπάρχουν αυτήν την στιγμή και φυσικά είναι και αυτή δωρεάν. Στην βάση μας θα αποθηκεύσουμε όλα τα δεδομένα που απαιτούνται για την λειτουργία της εφαρμογής μας. Παρακάτω θα εξεταστεί η χρήση της βάσης μας με περισσότερες λεπτομέρειες.
- ApacheServereίναι ένας εξυπηρετητής του παγκόσμιου ιστού (webserver) και είναι απαραίτητος ώστε να λειτουργήσουν όλα τα παραπάνω που αναφέραμε. Εξυπηρετεί τα πακέτα που έρχονται από τους χρήστες μέσω του προγράμματος πλοήγησης προωθώντας τα στο κατάλληλο webapplicationπου έχει από το οποίο δέχεται μία απάντηση την οποία προωθεί πίσω στο πρόγραμμα πλοήγησης.

3.4 Πλαίσια

Τα Πλαίσια είναι βοηθητικό λογισμικό που έχει σαν σκοπό να μας διευκολύνει στην ανάπτυξη της εφαρμογής μας. Δίνει μία κατεύθυνση και κάποιους κανόνες ως προς την δομή που θα έχει ο κώδικάς μας και την ροή της εκτέλεσης που θα ακολουθείται σε όλη την εφαρμογή μας. Αυτό που κερδίζουμε είναι ότι εκτός του ότι ο κώδικάς μας θα είναι πιο οργανωμένος με λιγότερες πιθανότητες για λάθη

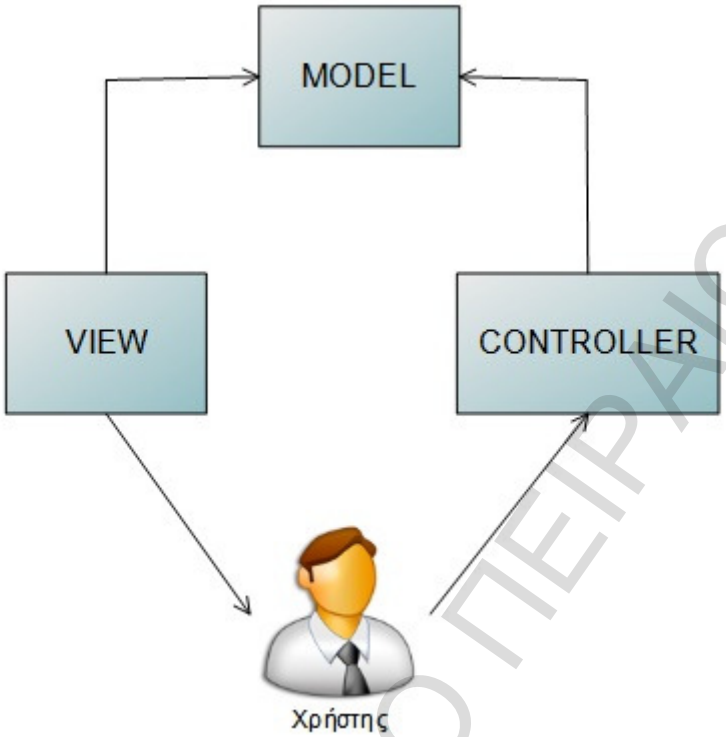
επιτυγχάνουμε κάποιες διεργασίες πιο γρήγορα και πιο εύκολα από το να γράφαμε τον κώδικα εξ αρχής. Φυσικά οι κλάσεις της Javaπου αποτελούν το πλαίσιομας έχουν την δυνατότητα να επεκταθούν ώστε να τροποποιούνται σε περίπτωση που ο χρήστης χρειαστεί κάποια αλλαγή στον τρόπο λειτουργίας του πλαισίουή απλά να προσθέσει κάποιες επιπλέον δυνατότητες. Παρακάτω, θα αναλύσουμε όλα τα πλαίσιαπου χρησιμοποιήσαμε στην εφαρμογή μας με παραδείγματα και τις απαραίτητες ρυθμίσεις που κάναμε για να λειτουργήσουν.

3.4.1 Spring

ToSpring**Error! Reference source not found.**είναι ένα πλαίσιοανοιχτού κώδικα(opensource) για Java εφαρμογές είτε είναι διαδικτυακές είτε όχι. Η πρώτη αναφορά για το springframework έγινε πριν από περίπου δέκα χρόνια τον Οκτώβριο του 2002 από τον 'Rod Johnson' ο οποίος δημιούργησε την πρώτη έκδοση και την δημοσίευσε με το βιβλίο του 'ExpertOne-on-One J2EE Design and Development'. Αργότερα τον Ιούνιο του 2003 δημοσίευσε στο διαδίκτυο το λογισμικό του με τα δικαιώματα που ορίζονται από το 'ApacheLicense'[26]. Η άδεια χρήσης αυτή είναι για ανοιχτό κώδικα και επιτρέπει σε όποιον θέλει να τροποποιεί το περιεχόμενο του κώδικα σύμφωνα με τις δικές του ανάγκες. Αυτό βοήθησε πάρα πολύ στην ανάπτυξη του λογισμικού καθώς δημιουργήθηκε μια ολόκληρη κοινωνία υποστήριξης και ανάπτυξης του λογισμικού. Μέχρι σήμερα έχουν βγει πολλές βελτιωμένες με την τελευταία να είναι η 3.2.2 η οποία δημοσιεύτηκε τον Μάρτιο του 2013.

Στην αρχή σχεδιάστηκε καθαρά για Javaεφαρμογές αργότερα όμως η ομάδα προγραμματιστών που υποστήριζε το πλαίσιοαποφάσισε να το επεκτείνουν και σε Webεφαρμογές και έτσι δημιούργησαν το MVCwebapplicationframework. Ο λόγος για αυτήν την απόφαση, όπως ισχυρίζονται, ήταν ότι θεωρούσαν τα άλλαπλαίσιαπου υπήρχαν τότε για διαδικτυακές εφαρμογές,όπως το Struts[27],κακοσχεδιασμένα και ελλιπή.

Τοπλαίσιοαυτό βασίζεται στο MVC μοντέλο το οποίο είναι και από τα πιο συνηθισμένα στον χώρο του διαδικτύου. Το ModelViewController (MVC) είναι ένα ευρέως χρησιμοποιούμενο πρότυπο αρχιτεκτονικής το οποίο βοηθάει πολύ τον προγραμματιστή να ξεχωρίσει τα τμήματα εκτέλεσης του κώδικα με ιδανικό τρόπο ώστε να δημιουργηθεί 'καθαρός' και ευανάγνωστος κώδικας ο οποίος έχει και την δυνατότητα επαναχρησιμοποίησης των ίδιων τμημάτων με διαφορετικό αποτέλεσμα ανάλογα και με τις εκάστοτε παραμέτρους και τα διαθέσιμα δεδομένα.



Εικόνα10ΜοντέλοMVC (Model View Controller)

Το MVCμοντέλο αποτελείται από τα Model, View,Controllerτα οποία εξυπηρετούν τον χρήστη και του προετοιμάζουν την σελίδα που θα δει. Αναλυτικά ο χρήστης μόλις κάνει κάποιο requestστον serverμας αυτό θα το παραλάβει ένας από τους controllersμας ο οποίος προηγουμένως έχει μετατραπεί σε SpringBeanκαθώς το πλαίσιομετατρέπει όλες τις κλάσεις που γράφουμε σε SpringBeans. Ο αριθμός των controllersπου μπορούμε να έχουμε στο σύστημά μας είναι απεριόριστος και ο κάθε ένας εξυπηρετεί κάποιο συγκεκριμένο request,έτσι ανάλογα με το linkπου επέλεξε ο χρήστης και την μέθοδο που επέλεξε (post,get) εκτελείται ο αντίστοιχος controller. Ο controller (ελεγκτής)είναι στην ουσία μία Javaκλάση και έχει όλες τις δυνατότητες που προσφέρει ηJava.Έτσι μπορούμε να επεξεργαστούμε τα δεδομένα όπως ακριβώς θέλουμε και να δημιουργήσουμε πολύπλοκες διεργασίες για να καλύψουμε όλες τις ανάγκες που μπορεί να έχει ένα νοσοκομείο όσο εξειδικευμένες και αν είναι αυτές.Έτσι λοιπόν αφού κληθεί το αντίστοιχος controllerκαι αφού λάβει υπόψη του τις παραμέτρους που δέχτηκε θα επικοινωνήσει (εάν είναι απαραίτητο) με το Model(Μοντέλο)για να λάβει τις πληροφορίες και τα δεδομένα που χρειάζεται. Στην δική μας περίπτωση τα δεδομένα αυτά βρίσκονται στην βάση δεδομένων και περιέχουν πληροφορίες για τους ασθενείς, τους υπάλληλους και γενικότερα οτιδήποτε έχει να κάνει με το νοσοκομείο. Εκτελώντας τα κατάλληλα sql ερωτήματα συγκεντρώνουμε τα απαραίτητα δεδομένα που θέλουμε και αφού γίνει και σε αυτά η οποιαδήποτε επεξεργασία χρειαστεί τα τελικά αποτελέσματα στέλνονται στο View. Το View(όψη) είναι αυτό που θα δεχτεί τα δεδομένα που θέλουμε να παρουσιάσουμε και θα δημιουργήσει την τελική σελίδα που θα αποσταλεί στον χρήστη για να του παρουσιάσουμε τα αποτελέσματα της αίτησης που έκανε στον server.

Παραπάνω περιγράψαμε μία κλασική και απλή επικοινωνία του χρήστη με το σύστημά μας με το μοντέλοMVC. Η σειρά όμως και ο τρόπος λειτουργίας του συστήματός μας μπορεί να διαφέρει ανάλογα με την περίσταση.Το μοντέλοMVC προσαρμόζεται αναλόγως για να διευθετήσει και να εξυπηρετήσει σωστά την αίτηση μας. Για παράδειγμα μπορεί μία αίτηση να μην χρειαστεί δεδομένα από το Modelοπότε ο controllerδεν θα επικοινωνήσει μαζί του αλλά θα καλέσει το κατάλληλο viewγια να παρουσιάσει την σελίδα που πρέπει. Αυτό μπορεί να συμβεί σε περίπτωση που θέλουμε να ανοίξουμε μία στατική σελίδα που δεν χρειάζεται δεδομένα από την βάση ή εάν ζητήσουμε δεδομένα στα οποία δεν έχουμε πρόσβαση οπότε δεν θα γίνει ποτέ η επικοινωνία με την βάση δεδομένων αλλά θα εμφανιστεί

απλά το μήνυμα άρνησης πρόσβασης. Τα σενάρια λειτουργίας του μοντέλου είναι πολλά και αναλόγως το μοντέλοMVCπροσαρμόζεται σε αυτά.

Η χρήση του Springμας βοηθάει να έχουμε πιο οργανωμένο και καθαρογραμμένο κώδικα και μας διευκολύνει στην γραφή του προγράμματος καθώς διευκολύνει και αυτοματοποιεί πολλές διαδικασίες. Εφαρμόζει το MVCμοντέλο που περιγράψαμε προηγουμένως και προσφέρει επίσης χρήσιμες βιβλιοθήκες για διάφορες λειτουργίες όπως:

- Επικοινωνία με οποιαδήποτε βάση δεδομένων θέλουμε.
- Αυθεντικοποίηση-πιστοποίηση και εξουσιοδότηση του χρήστη (σε συνδυασμό με το springsecurityπου θα δούμε παρακάτω).
- Εύκολη διαχείριση σφαλμάτων (errorhandling).
- Καταγραφή αρχείων αναφοράς (logfiles).
- Διαχείριση πολυγλωσσικών μηνυμάτων για εφαρμογές με πολλές γλώσσες.
- Επιπλέον Tagsγια τα jspαρχεία μας για δημιουργία φορμών HTMLκαι διαχείριση των δεδομένων που θα παρουσιάσουμε.

3.4.2 SpringSecurity

ToSpringSecurity**Error! Reference source not found.**είναιμία προέκταση του SpringFrameworkκαι αποτελεί ένα πλήρες πακέτο ασφάλειας που μπορεί να διαχειριστεί όλες τις ομάδες χρηστών που έχουμε στην εφαρμογή μας διαμοιράζοντας τα καθήκοντα και τις λειτουργίες της εφαρμογής μας στην αρμόδια ομάδα.Η πρώτη έκδοσή του βγήκε το 2003 και από τότε θεωρείται από τα πιο σταθερά και ισχυρά εργαλεία για την ασφάλιση διαδικτυακών εφαρμογών είτε είναι βασισμένα σε Springείτε όχι καθώς δεν είναι προϋπόθεση η εφαρμογή μας να είναι σε SpringFramework. Σύμφωνα με τα στοιχεία της springsource[28]αυτό το πλαίσιοασφάλειας χρησιμοποιείται εκτός άλλων σε κυβερνητικές, στρατιωτικές και τραπεζικές εφαρμογές πράγμα που το καθιστά αξιόπιστο. Έχει εκδοθεί και αυτό υπό την άδεια της 'Apache 2.0 License' και διαμοιράζεται δωρεάν προς χρήση και τροποποίηση από τους χρήστες του. Σκοπός της χρήσης του είναι η διευκόλυνση στην συγγραφή κώδικα και η διαφύλαξη της ασφάλειας στο σύστημά μας βασιζόμενοι σε μία εφαρμογή που συντηρείται από μία μεγάλη ομάδα έμπειρων προγραμματιστών με πολύχρονη εμπειρία στον χώρο της ασφάλειας.

Αναλαμβάνει την πιστοποίηση – αυθεντικοποίηση και την εξουσιοδότηση του κάθε χρήστη που συνδέεται στην εφαρμογή. Η σύνδεση στην εφαρμογή γίνεται με την εισαγωγή του ονόματος χρήστη και του κωδικού στην φόρμα σύνδεσης. Από εκεί τοSpringFrameworkαναλαμβάνει να πιστοποιήσει δηλαδή να αυθεντικοποιήσει τον χρήστη ελέγχοντας καταρχήν εάν υπάρχει αυτός ο χρήστης και στην συνέχεια εάν ο κωδικός που δόθηκε είναι σωστός. Για τον κωδικό του χρήστη δίνεται η δυνατότητα να χρησιμοποιηθεί κωδικοποίηση πολλών επαναλήψεων για να αυξήσουμε την ασφάλεια του συστήματος. Αφού γίνει η πιστοποίηση ο χρήστης ανάλογα με την ομάδα χρηστών που ανήκει θα αποκτήσει πρόσβαση στις ανάλογες σελίδες που του επιτρέπεται και θα απαγορεύει την είσοδο σε σελίδες και λειτουργίες που αφορούν άλλες ομάδες χρηστών.

Η διαδικασία αυτή που περιγράψαμε παραπάνω θα μπορούσε να γίνει και χωρίς κανένα πλαίσιοαπλά γράφοντας τις δικές μας κλάσεις και κάνοντας του αρμόδιους ελέγχους σε κάθε κλήση που κάνει ο χρήστης στο σύστημά μας. Αυτό όμως απαιτεί περισσότερο χρόνο για να γραφτούν πολλές γραμμές κώδικα και έχει μεγαλύτερες πιθανότητες για λάθη και κενά στην ασφάλεια καθώς πρέπει να καλυφθούν όλες οι ομάδες χρηστών για όλες τις κλάσεις και μεθόδους που θα δημιουργήσουμε. Χρησιμοποιώντας το springframeworkτο μόνο που χρειάζεται να κάνουμε είναι να προσθέσουμε μία γραμμή κώδικα, πριν από κάθε μέθοδο ή πριν τον ορισμό της κλάσης, και να ορίσουμε ποιες ομάδες χρηστών μπορούν να χρησιμοποιήσουν την μέθοδο ή την κλάση. Μία τέτοια γραμμή κώδικα μπορούμε να δούμε παρακάτω:

```
@PreAuthorize("hasAnyRole('ROLE_2','ROLE_3')")
publicStringshowPatientSearchForm() { … }
```

Παραπάνω βλέπουμε ένα απλό παράδειγμα όπου ορίζουμε ότι στην μέθοδο 'showPatientSearchForm' θα έχουν πρόσβαση μόνο οι χρήστες που έχουν τους ρόλους 2 ή 3 που στην

περίπτωση της εφαρμογής μας αφορά τους ιατρούς και τους νοσοκόμους. Οποιοσδήποτε άλλος χρήστης ανήκει σε διαφορετική ομάδα χρηστών και προσπαθήσει να χρησιμοποιήσει την μέθοδο αυτή δεν θα μπορέσει διότι το σύστημα θα του αρνηθεί την πρόσβαση και θα λάβει ένα μήνυμα άρνησης πρόσβασης.

3.4.3 Hibernate

Το Hibernateείναι ένα ακόμη βοηθητικό πλαίσιοτο οποίο συμβάλλει στην επικοινωνία της εφαρμογής Javaμε μία βάση δεδομένων. Είναι και αυτό δωρεάν λογισμικό το οποίο διανέμεται με την άδεια της GNU Lesser General PublicLicense[29]. Η πρώτη της έκδοση κυκλοφόρησε το 2001 ενώ αργότερα δημιουργήθηκε και το Hibernate 2 το οποίο είχε πολλές βελτιώσεις και προσθήκες νέων λειτουργιών κάτι που βοήθησε πολύ στην διάδοση και την ευρύτερη χρήσητου από προγραμματιστές τις Java.Αυτό δεν έμεινε απαρατήρητο και η RedHat[30]προσέλαβε την ομάδα υποστήριξης του HibernateFrameworkκαι δούλεψε μαζί τους στην βελτίωση και ανάπτυξή του. Έτσι το 2010 κυκλοφόρησε το Hibernate 3 ενώ τον Δεκέμβριο του 2011 το Hibernate 4. Αυτήν την στιγμή προετοιμάζουν την 5ή έκδοση του πλαισίου.

Όπως αναφέραμε προηγουμένως η χρήση του Hibernatemας βοηθάει στην επικοινωνία και την ανταλλαγή πληροφοριών της εφαρμογής που έχουμε με την βάση δεδομένων που χρησιμοποιούμε. Η εφαρμογή μας μπορεί να είναι απλή εφαρμογήJavaή διαδικτυακή εφαρμογή Javakαι υποστηρίζει επικοινωνία με όλες τις παρακάτω βάσεις δεδομένων:

1. DB2	2. Firebird
3. FrontBase	4. HP NonStop SQL/MX
5. HypersonicSQL	6. Informix
7. Ingres	8. Interbase
9. Microsoft SQL Server	10. MySQL
11. Oracle	12. Pointbase
13. PostgreSQL	14. SAP DB
15. Sybase	16. InterSystems
17. Cache'MS Access, Corel Paradox, CSV, plain text, Xbase database, MS Excel, Cobol data, XML document	

Πίνακας 2Υποστηριζόμενες βάσεις δεδομένων από το Hibernate

Η βασική λειτουργία του HibernateFrameworkείναι να αντιστοιχεί (mapping) πίνακες της βάσης μας με Javaκλάσεις που δημιουργούμε στην εφαρμογή μας όπου και ορίζουμε με μεταβλητές την δομή που έχει ο πίνακας που θα αντιστοιχίσουμε για να μπορούν να εισαχθούν οι τιμές των σειρών του πίνακα στις μεταβλητές που ορίσαμε. Έτσι δημιουργούμεJava αντικείμενα τα οποία τα γεμίζουμε με δεδομένα από την βάση και τα χρησιμοποιούμε όπως θέλουμε στην εφαρμογή μας ή μπορούμε και να τα αλλάξουμε αλλάζοντας παράλληλα και το περιεχόμενο του πίνακα στην βάσ.Επίσης έχει μηχανισμούς μετατροπής των δεδομένων από μορφή Javaστην μορφή που απαιτεί η βάση και το αντίστροφο. Αυτό είναι εξαιρετικά χρήσιμο καθώς πολύ συχνά παρατηρούνται ασυμβατότητες στην μορφή των δεδομένων που αποθηκεύονται στην βάση με αυτά που χρησιμοποιούμε στην εφαρμογή μας.

Ο προγραμματιστής αφού δημιουργήσει την κλάση του θα την αντιστοιχήσει με κάποιο πίνακα της βάσης μας με δύο εύκολους τρόπους.Είτε με την καταχώριση των κλάσεων σε κάποιο συγκεντρωτικό xmlαρχείο ή με την χρήση των “annotations”. Η δεύτερη μέθοδος είναι εξαιρετικά απλή και είναι αυτή που χρησιμοποιήθηκε και στην εφαρμογή μας για λόγους ευκολίας όπως θα δούμε και στο παράδειγμα παρακάτω όπου ορίζουμε μία κλάση patientκαι την αντιστοιχούμε χρησιμοποιώντας “annotation” στον πίνακα “Patients” της βάσης μας. Στην περίπτωση που το Hibernateδεν βρει τον πίνακα με τα στοιχεία που του έχουμε ορίσει στην βάση μας αυτό θα τον δημιουργήσει με βάση τα στοιχεία που έχουμε ορίσει στο αντικείμενο που έχουμε φτιάξει.

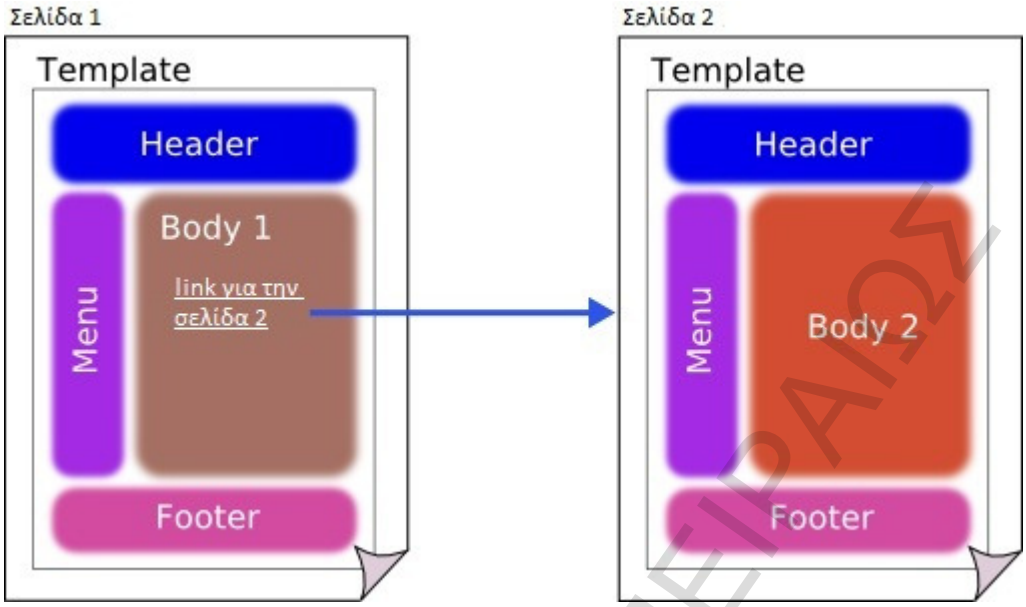
```
@Entity
@Table(name = "Patients")
publicclassPatient { … }
```

Στην συνέχεια όταν χρειαστούμε κάποια δεδομένα από την βάση μας είτε για προβολή είτε για τροποποίηση δεν θα έχουμε να κάνουμε πολλές διαδικασίες διότι θα μας διευκολύνει το πλαίσιο. Έτσι χρησιμοποιώντας της μεθόδους και τις κλάσεις που διαθέτει συνδεόμαστε άμεσα με την βάση και 'γεμίζουμε' τα αντικείμενα Javaπου έχουμε με δεδομένα από την βάση γράφοντας ελάχιστες γραμμές κώδικα σε σχέση με τη διαδικασία που θα έπρεπε να ακολουθήσουμε χωρίς την χρήση του πλαισίου.Οι δυνατότητες που έχουμε για την εξαγωγή δεδομένων από την βάση χρησιμοποιώντας την γλώσσα της Hibernate (hql) μέσω των κλάσεων που διαθέτειείναι ίδιες με τις δυνατότητες που θα είχαμε εάν χρησιμοποιούσαμε την οποιαδήποτε γλώσσα που διαθέτει η κάθε βάση δεδομένων. Το κυριότερο πλεονέκτημα που έχουμε είναι ότι μαθαίνοντας να χρησιμοποιούμε την hqlμπορούμε να γράψουμε μία εφαρμογή η οποία θα μπορεί να λειτουργήσει με οποιαδήποτε βάση δεδομένων και αν χρησιμοποιήσουμε. Δηλαδή εάν στην πορεία για οποιοδήποτελόγω αλλάξουμε την βάση δεδομένων μας δεν θα χρειαστεί να αλλάξουμε τον κώδικα διότι το HibernateFrameworkμεταφράζει την hqlγλώσσα που χρησιμοποιεί στην κατάλληλη μορφή που χρειάζεται η οποιαδήποτε βάση δεδομένων χρησιμοποιήσουμε.

3.4.4 Tiles

ΤαTiles**Error! Reference source not found.**είναι ένα ακόμα βοηθητικό πλαίσιοπου χρησιμοποιήσαμε και μας βοηθάει να διαχειριστούμε την εμφάνιση της ιστοσελίδας μας. Για την δημιουργία ιστοσελίδων χρησιμοποιούμε συνήθως κάποιο 'Template' το οποίο καθορίζει την εμφάνιση που θα έχει η σελίδαπου θα επισκεφτούμε. Αυτό το ακολουθούμε σε κάθε σελίδα που δημιουργούμε ώστε να έχουμε ένα ομοιόμορφο αποτέλεσμα χωρίς πολλές κουραστικές αλλαγές, στην εμφάνιση του ιστοτόπου, για το μάτι του επισκέπτη. Το πρόβλημα σε αυτό είναι ότι θα πρέπει σε κάθε σελίδα που δημιουργούμε να έχουμε κάποιο κώδικα ο οποίος θα επαναλαμβάνεταιμε αποτέλεσμα οποιαδήποτε αλλαγή θα πρέπει να γίνει σε όλες τις σελίδες που δημιουργήσαμε προηγουμένως.Εδώ είναι που μπορεί να μας βοηθήσει το TilesFrameworkτο οποίο ενώ στην αρχή δημιουργήθηκε αποκλειστικά για διαδικτυακές εφαρμογές Javatώρα πλέον μπορεί να χρησιμοποιηθεί και σε οποιαδήποτε εφαρμογήJava.

Αυτό που δημιουργούμε με τα Tileseίναι κάτι σαν 'κορνίζα' της σελίδας μας η οποία θα μένει σταθερή και εμείς θα αλλάζουμε το περιεχόμενό της.Για να γίνει αυτό χωρίζουμε σε τμήματα το περιεχόμενό μας και ορίζουμε ποια από αυτά θα είναι σταθερά και ποιο θα είναι μεταβλητό. Στην δικιά μας εφαρμογή έχουμε ορίσει τρία σταθερά τμήματα και ένα μεταβλητό. Τα σταθερά τμήματα είναι το πάνω μέρος(header) που περιλαμβάνει τον τίτλο και το λογότυπο, το αριστερό τμήμα(menu) που περιέχει το μενού μας και το κάτω μέρος (footer) το οποίο περιέχει ελάχιστες πληροφορίες για της σελίδας μας. Το μεταβλητό μέρος είναι το κύριο μέρος τις σελίδας μας (body) στο οποίο παρουσιάζουμε και το περιεχόμενο της εφαρμογής μας. Έτσι όταν ο χρήστης επιλέξει κάποια λειτουργία ή ανοίξει κάποια σελίδα από το αριστερό μενού τα τρία σταθερά τμήματα μένουν όπως ήταν και απλά αλλάζει το μεταβλητό κομμάτι το οποίο θα παρουσιάσει αυτό που ζήτησε ο χρήστης. Αυτό βλέπουμε και στην εικόνα 11 όπου αλλάζει μόνο το περιεχόμενο του Body 1 σε Body 2.



Εικόνα 11 Σελίδες με TilesFramework

3.5 Άλλα πρόσθετα

3.5.1 JavaScript

Η JavaScriptείναι μία από τις πιο γρήγορες γλώσσεςσεναρίων που κυκλοφορεί. Ο αρχικός της προορισμός ήταν να διαχειρίζεται το περιεχόμενο μίας ιστοσελίδας του διαδικτύου και να προσφέρει κάποιες χρήσιμες λειτουργίες. Πλέον,λόγω της επιτυχίας της γλώσσας χρησιμοποιείται και για δημιουργία μικροεφαρμογών και μικρών παιχνιδιών. Είναι δωρεάν, ευκολόχρηστη και τα αρχεία που δημιουργούμε δεν χρειάζονται compileαλλά μπορούν να εκτελεστούν όπως είναι σχεδόν από όλους τους φυλλομετρητές ιστοσελίδων (webbrowsers) ακόμα και από αυτούς που χρησιμοποιούν τα σύγχρονα κινητά τηλέφωνα.Παρά το όνομά της γλώσσας δεν θυμίζει καθόλου Java.Αντιθέτως, έχουν πολλές διαφορές και από ό,τι αναφέρεται έχει επηρεαστεί περισσότερο από την γλώσσα προγραμματισμού C.

Τα στοιχεία που κάνουν την γλώσσα χρήσιμη και ισχυρή είναι η ταχύτητά της και το γεγονός ότι εκτελείται στην πλευρά του χρήστη (client-side). Δηλαδή εάν επισκεφτούμε έναν ιστότοπο που περιέχει javascriptκώδικα αυτόσθα κατέβει στον υπολογιστή μας και θα εκτελεστεί εκεί χωρίς να επιβαρύνει τον server.Ενώ παράλληλα αυτό δημιουργεί και ένα αρνητικό στοιχείο επειδή το να εκτελείται το scriptστον υπολογιστή μας είναι επιβλαβές διότι μπορεί να αποκτήσει πρόσβαση σε προσωπικά στοιχεία όπως σε υποθηκευμένους κωδικούς και είτε να τα παραβιάσει είτε να τα καταστρέψει.

Οι δυνατότητες που δίνει αυτή η γλώσσα είναι πραγματικά πολλές και αφορούν ελέγχους δεδομένων, μετατροπές περιεχομένου, μαθηματικές πράξεις, εμφάνιση η απόκρυψη στοιχείων με βάση κάποια κριτήρια και πολλά άλλα. Γενικότερα μπορεί να δώσει μία διαδραστική εικόνα στην σελίδα μας χωρίς αυτή να κάνει κλήσεις στον διακομιστή αλλά λειτουργώντας πάντα στο περιβάλλον του χρήστη. Αυτό γιατί με την Javascript μπορούμε να δημιουργήσουμε κάποιες συναρτήσεις οι οποίες θα εκτελούνται μετά από κάποιο συγκεκριμένο γεγονός που θα ορίσουμε εμείς (event). Έτσι για παράδειγμα όταν ο χρήστης επιλέξει κάτι συγκεκριμένο η υποβάλει κάποια φόρμα εμείς μπορούμε πριν στείλουμε τη αίτηση στον διακομιστή να την ελέγξουμε ή να επεξεργαστούμε κάποια από τα στοιχεία που συμπλήρωσε. Επίσης, μπορούμε να εμφανίσουμε κάποια μηνύματα με αναδυόμενα παράθυρα (popup) για να ενημερώσουμε τον χρήστη μας για κάτι ή να εμφανίσουμε κάποιο μήνυμα λάθους.

Στην εφαρμογή μας χρησιμοποιούμε την Javascriptπολύ συχνά είτε σαν απλά κομμάτια κώδικα είτε με την βοήθεια βιβλιοθηκών όπως η jQuery για την οποία θα αναφερθούμε παρακάτω. Κάποιες λειτουργίες για τις οποίες χρησιμοποιήσαμε την Javascriptείναι:

- Κεντρικό μενού στην αριστερή πλευρά της σελίδας μας.
- Τα tabsγια τον φάκελο του ασθενή (με την βοήθεια της βιβλιοθήκης jQuery).
- Αναδυόμενα παράθυρα (popups με την βοήθεια της βιβλιοθήκης jQuery).
- Αναδυόμενο βοηθητικό παράθυρο για την εισαγωγή ημερομηνίας (popups με την βοήθεια της βιβλιοθήκης jQuery).
- Πίνακες ταξινόμησης δεδομένων (με την βοήθεια pluginτης jQuery).
- Άνοιγμα σελίδας σε layout σαν αναδυόμενο παράθυρο πάνω από το δικό μας (με την βοήθεια τουpluginτης jQuery toframewarp[32]).
- Έλεγχοι τιμών και μαθηματικές πράξεις (φαρμακείο).
- TextAreasμε εργαλεία του word(με την βοήθεια της βιβλιοθήκης tinymce[33])
- Προετοιμασία περιεχομένων φακέλου ασθενή για εκτύπωση.

3.5.2 jQuery

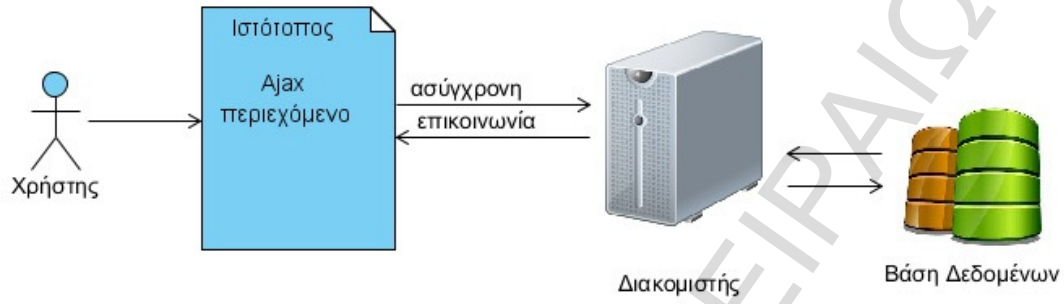
HjQuery**Error! Reference source not found.**είναι μία πολύ χρήσιμη με μεγάλο περιεχόμενο βιβλιοθήκη της JavaScriptη οποία μας βοηθάει να βάλουμε κάποιες βασικές λειτουργίες στο 'UserInterface' της σελίδας μας. Είναι και αυτό ένα πλαίσιοόμως δεν είναι για το ServerSideκομμάτι της εφαρμογής μας αλλά για το ClientSideδηλαδή αυτό που παρουσιάζεται στην χρήστη. Η διαφορά είναι ότι η εκτέλεση του κώδικα της JavaScriptγίνεται στον ηλεκτρονικό υπολογιστή του χρήστη και όχι στον διακομιστή που έχουμε την εφαρμογή μας. Αυτό διευκολύνει τον διακομιστή μας ώστε να έχει λιγότερο φόρτο εργασίας επιβαρύνοντας στο ελάχιστο τον υπολογιστή του χρήστη μας διότι η γλώσσα αυτή είναι ταχύτατη και γι'αυτό και εκτελείται και σε σύγχρονα κινητά χωρίς δυσκολίες. Μας βοηθάει στο να κάνουμε το περιεχόμενό μας πιο εύχρηστο και πιο διαδραστικό με τον χρήστη μας. Η jQuery πρωτοεμφανίστηκε τον Ιανουάριο του 2006 στο BarCamp[35] από τον JohnResig. Πρόκειται για μια βιβλιοθήκη JavaScript ανοιχτού κώδικα, υπό τις άδειες MIT License και την GNU GeneralPublicLicense. Αυτό μας δίνει την δυνατότητα να τροποποιήσουμε το περιεχόμενο της αλλάζοντας οποιαδήποτε λειτουργία της σύμφωνα με τις δικές μας ανάγκες.Τα πλεονεκτήματα που μας προσφέρει είναι:

1. Ευκολία χρήσης λόγω καλογραμμένου και απλού κώδικα.
2. Πλήρη περιγραφή κάθε λειτουργίας και ηλεκτρονική παρουσίαση με παραδείγματα στην ιστοσελίδα τηςjQuery.
3. Υποστήριξη από μεγάλη και ενεργή κοινότητα προγραμματιστών. Βελτιώνεται και εξελίσσεται συνέχεια με νεότερες εκδόσεις.
4. Μικρό μέγεθος καθώς ο κώδικας είναι καθαρογραμμένος και έχει δυνατότητα 'συμπίεσης' που βοηθάει στην γρηγορότερη εκτέλεσή του.
5. Ποικιλία χαρακτηριστικών και δυνατοτήτων που προσφέρει στον προγραμματιστή μέσω της μεγάλης βιβλιοθήκης που διαθέτει.
6. Δυνατότητα επέκτασης προσθέτοντας η αλλάζοντας κάποια κομμάτια κώδικα.
7. Υποστήριξη Ajax.
8. Plug-insπολλά πρόσθετα που διατίθενται δωρεάν στο διαδίκτυο τα οποία μπορούν να επεκτείνουν τις δυνατότητες που έχουμε με το jQuery.

3.5.3 Ajax

Ajax (AsynchronousJavaScriptandXML)Programming**Error! Reference source not found.**δεν είναι γλώσσα προγραμματισμού αλλά είναι μία τεχνολογία που μας επιτρέπει την ασύγχρονη επικοινωνία της πλευράς του χρήστη (clientside) με την πλευρά του διακομιστή (serverside) χρησιμοποιώντας τις υπάρχουσες γλώσσες. Λέγοντας ασύγχρονη επικοινωνία εννοούμε ότι η επικοινωνία αυτή και η οποιαδήποτε ανταλλαγή δεδομένων γίνεται συμβαίνει σιωπηλά στο 'παρασκήνιο' χωρίς ο χρήστης να αντιλαμβάνεται το παραμικρό. Τέτοιου είδους επικοινωνία είναι χρήσιμη σε περίπτωση που θέλουμε κάποια δεδομένα από την βάση μας αλλά δεν θέλουμε να χάσουμε την σελίδα στην οποία βρισκόμαστε ή

να διακόψουμε τον χρήστη από αυτό που κάνει. Αυτό συμβαίνει και στην εφαρμογή μας στην περίπτωση που ο ιατρός πληκτρολογεί κάποιο όνομα φαρμάκου και ταυτόχρονά έχουμε μία ασύγχρονη επικοινωνία με τον διακομιστή. Εκεί αναφέρουμε τα γράμματα που πληκτρολογεί ο χρήστης και δημιουργούμε μία λίστα με πιθανές ονομασίες φαρμάκων, από την βάση μας, που η ονομασία τους ταιριάζει με αυτό που πληκτρολογεί ο γιατρός. Στην συνέχεια αυτή η λίστα στέλνεται στον χρήστη και παρουσιάζεται σαν πιθανές επιλογές σε μία λίστα αυτόματης συμπλήρωσης. Αυτό όλο συμβαίνει απαρατήρητα και σε στιγμιαίο χρονικό διάστημα οπότε ο χρήστης δεν αντιλαμβάνεται κάτι διαφορετικό παρά μόνο βλέπει την λίστα της αυτοσυμπλήρωσης με τα πιθανά ονόματα για να επιλέξει όποιο του χρειάζεται.



Εικόνα 12Ajax Επικοινωνία

3.6 Βάση Δεδομένων

Σε ένα πολύπλοκο περιβάλλον όπως είναι η υγεία και η περίθαλψη τα δεδομένα που δημιουργούνται καθημερινά έχουν πολύ μεγάλο όγκο και πρέπει να αποθηκεύονται και να παραμένουν προσβάσιμα και στο μέλλον. Τα δεδομένα αυτά είναι μόνιμα και αυξάνονται καθημερινά οπότε η αποθήκευσή τους και η διαφύλαξη τους είναι από τα κυριότερα στοιχεία του συστήματός μας. Η αποθήκευση φυσικά θα γίνει σε κάποια βάση δεδομένων. Όμως ο τύπος της βάσης δεδομένων που θα χρησιμοποιήσουμε και η ρύθμισή της είναι μία πολύπλοκη διαδικασία.

Οι σχεσιακές βάσεις δεδομένων (relationaldatabase) είναι από τις πιο συνηθισμένες βάσεις που χρησιμοποιούνται για τέτοιου είδους πολύπλοκα συστήματα. Μερικές από τις πιο γνωστές εμπορικές βάσεις του είδους είναιοι OracleDatabase, IBMDB2, MicrosoftSQLServer, SAPSybaseκαι Teradata ενώ υπάρχουν και δωρεάν εκδόσεις όπως η MySQLκαι η PostgreSQL. Στην εφαρμογή μας θα χρησιμοποιήσουμε MySQLμιας και είναι δωρεάν και είναι αρκετά καλή και γρήγορη βάση δεδομένων και ικανοποιεί τις απαιτήσεις μας.

Μία άλλη μορφή βάσης που χρησιμοποιείται είναι η αντικειμενοστρεφής βάση δεδομένων (objectoriented)η οποία κρατάει τα δεδομένα που αποθηκεύονται σαν αντικείμενα. Έτσι ένας πελάτης μπορεί να θεωρηθεί ότι είναι ένα αντικείμενο και όλα τα δεδομένα που τον αφορούν αποθηκεύονται σε αυτό το αντικείμενο. Αυτό μας θυμίζει τον τρόπο λειτουργίας της Javaπου είναι και αυτή αντικειμενοστρεφής γλώσσα προγραμματισμού και από την τάση να χρησιμοποιούνται τα αντικείμενα όλο και περισσότερο δημιουργήθηκε και η αντίστοιχη βάση δεδομένων η οποία συνεργάζεται καλύτερα με τις αντικειμενοστρεφείς γλώσσες προγραμματισμού.

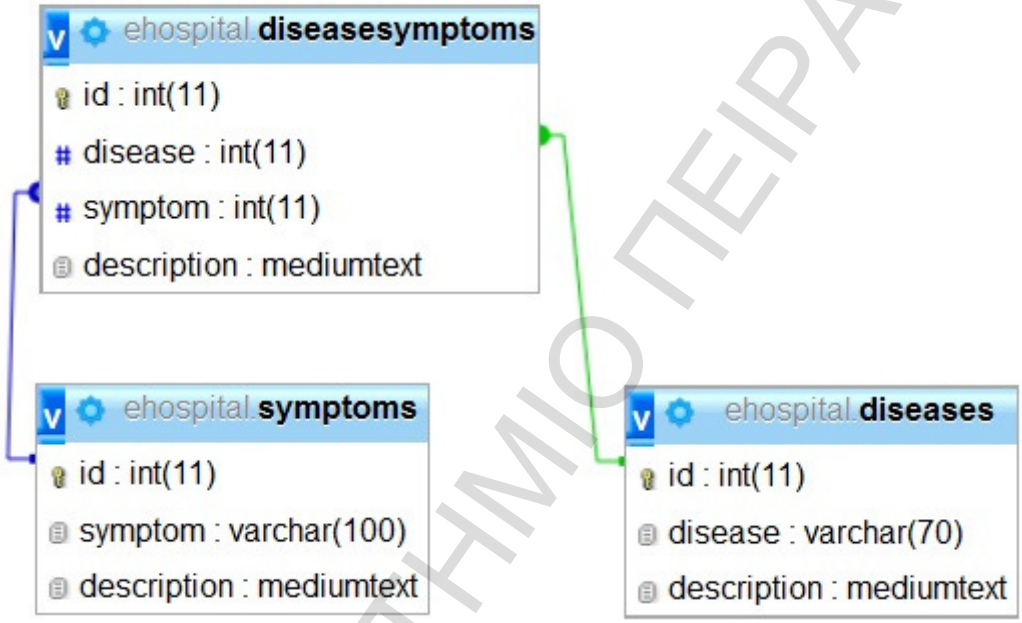
Τέλος μία ακόμη βάση δεδομένων που ξεκίνησε γύρω στο 2000 είναι η post-relational ή αλλιώς NoSQL (NotOnlySQL)βάση δεδομένων η οποία θεωρείται ότι είναι η εξέλιξη της σχεσιακής βάσης δεδομένων που αναφέραμε προηγούμενος. Δεν απαιτεί αυστηρά σχήματα και συσχετίσεις μεταξύ των πινάκων ενώ μπορεί να αποτελείταικαι από αντικείμενα. Αυτές οι βάσεις σχεδιάστηκαν για να διαχειρίζονται πολλά δεδομένα μεγάλου όγκου σε μικρό χρονικό διάστημα. Τέτοια βάση δεδομένων είναι η Cachekαι σύμφωνα και με κάποιες μελέτες[37] που έχουν γίνει παρατήρησαν ότι ανταποκρίνεται πιο γρήγορα και χρησιμοποιεί λιγότερη υπολογιστική ισχύ από ότι σχεσιακές βάσεις δεδομένων όπως η Oracle.

Στην δική μας εφαρμογή όπως αναφέραμε προηγουμένως χρησιμοποιήσαμε την MySQL, μία από τις πιο διαδεδομένες γλώσσες με εκατομμύρια χρήστες σε όλο τον κόσμο να την επιλέγουν για την

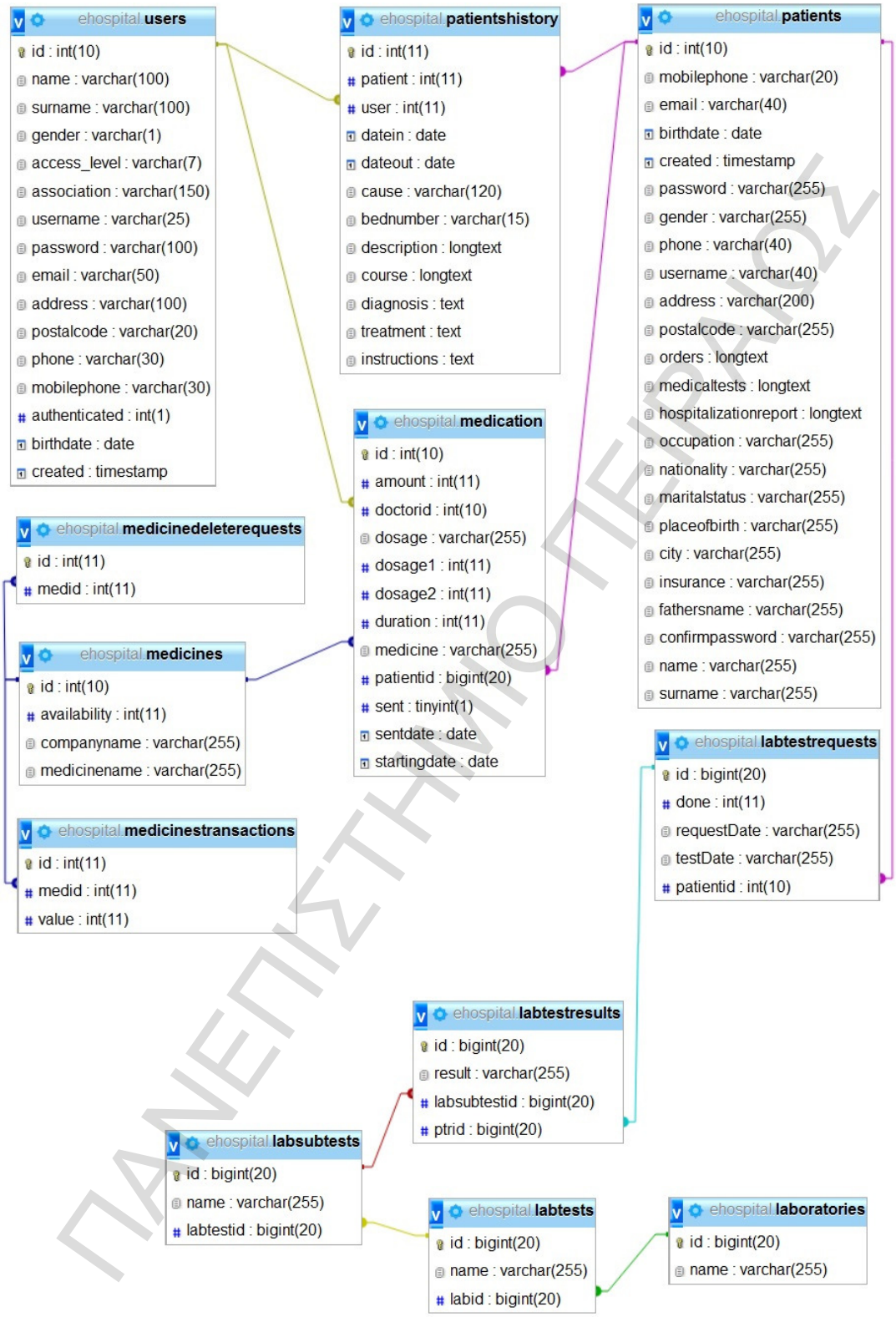
αποθήκευση των δεδομένων τους. Σίγουρα αυτό οφείλεται και στο ότι είναι ΄ανοιχτού κώδικα΄ και διανέμεται δωρεάν χωρίς καμία χρέωση. Εκτός όμως από αυτό είναι και αξιόπιστη και γρήγορη με αποτέλεσμα να καλύπτει τις ανάγκες των περισσότερων εφαρμογών.

Στα παρακάτω σχήματα θα παρουσιάσουμε την δομή και τις συσχετίσεις των πινάκων της βάσης που δημιουργήσαμε. Οι εικόνες έχουν δημιουργηθεί με το εργαλείο 'phpMyAdmin' το οποίο έχει την δυνατότητα να παρουσιάζει την βάση την οποία έχουμε σε γράφημα.

Στο πρώτο σχήμα έχουμε τους τρεις πίνακες που αφορούν την λειτουργία της διάγνωσης. Οι πίνακες αυτοί δεν σχετίζονται με κανέναν άλλον από την βάση μας και περιέχουν όλες τις πληροφορίες που απαιτεί η εφαρμογή της διάγνωσης δηλαδή όλες τις ασθένειες και τα συμπτώματα που έχουν καταχωρηθεί και τις απαραίτητες συσχετίσεις μεταξύ τους ώστε να μπορεί να γίνει η διάγνωση.



Εικόνα 13 Πίνακες Ασθενειών καιΣυμπτωμάτων.



Εικόνα 14Υπόλοιποι πίνακες της βάσης.

Στην προηγούμενη εικόνα 14βλέπουμε τους κυριότερους πίνακες που έχουμε δημιουργήσει για την βάση μας και περιέχουν τον κύριο όγκο των πληροφοριών που διαχειρίζεται η εφαρμογ. Έχουμε τον

πίνακα ‘users’ο οποίος αφορά τους χρήστες της εφαρμογής μας (διαχειριστές, ιατροί, νοσηλευτές, φαρμακοποιοί, εργαστηριακοί ιατροί). Τον πίνακα ‘patients’ο οποίος περιέχει τους ασθενείς και τα στοιχεία τους, τον πίνακα ‘patientshistory’ όπου αποθηκεύεται το ιστορικό του ασθενή. Τους πίνακες ‘medication’, ‘medicines’, ‘medicinestransactions’ και ‘medicinedeleterequest’ οι οποίοι περιέχουν τα φάρμακα και την διαθεσιμότητά τους, την φαρμακευτική αγωγή που πρέπει να ακολουθήσει ο κάθε ασθενής και διάφορες διαδικασίες καταγραφής των ενεργειών των φαρμακοποιών που αφορούν προσθαφαίρεση και γενικότερη διαχείρισηφαρμάκων. Τέλος είναι οι πίνακες ‘labtestrequests’, ‘labtestresults’, ‘labsubtests’, ‘labtests’, ‘laboratories’ που αφορούν τις εργαστηριακές εξετάσεις που μπορεί να ζητήσει ο ιατρός για κάποιον ασθενή και περιέχουν όλες τις δυνατές εξετάσεις, τις αιτήσεις προς τα εργαστήρια και τα αποτελέσματα από τους εργαστηριακούς ιατρούς.

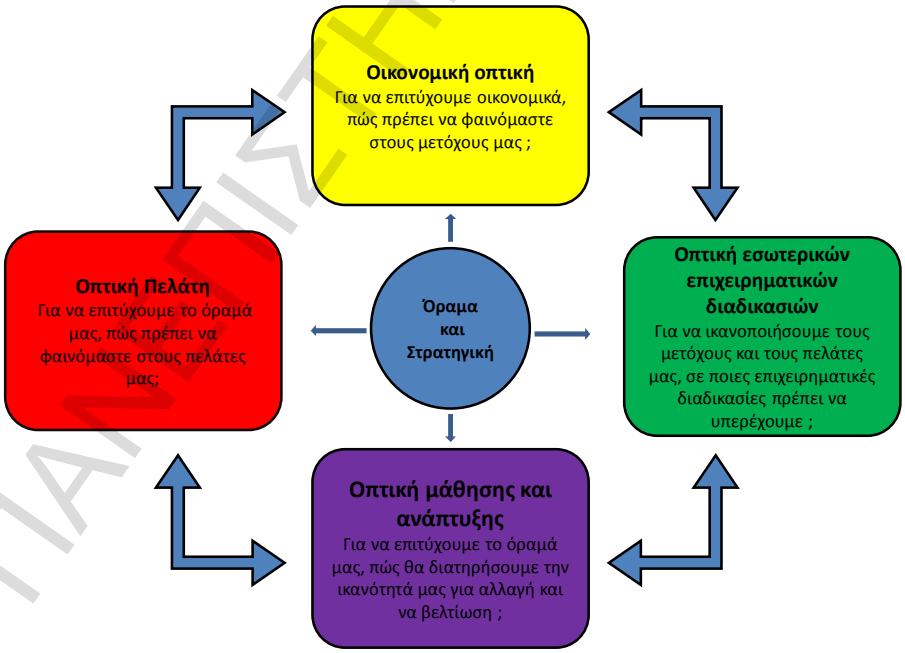
4. Αξιολόγηση Συστήματος

Η αξιολόγηση ενός πληροφοριακού συστήματος είναι μία πολύπλοκη, δύσκολη αλλά ταυτόχρονα πολύ σημαντική διαδικασία που πρέπει να εφαρμοστεί στην εφαρμογή μας για να διαπιστώσουμε το κατά πόσο έχουμε πετύχει τους στόχους μας και εάν αξίζει το κόστος του. Η καλύτερη αξιολόγηση ενός συστήματος φυσικά επιτυγχάνεται μόνο αφού εγκατασταθεί και χρησιμοποιηθεί η εφαρμογής από τα εμπλεκόμενά άτομα για κάποιο χρονικό διάστημα. Έτσι θα μπορούσαμε να έχουμε μία πλήρη εικόνα για

τα οφέλη και τις διευκολύνσεις που μας παρέχει συγκρίνοντας τους χρόνους λειτουργίας, τις αξιολογήσεις των ασθενών και των υπαλλήλων καθώς και τα οικονομικά κέρδη σε σχέση με την κατάσταση που επικρατούσε πριν την εγκατάσταση και χρήση της εφαρμογής. Αυτό όμως επειδή είναι πρακτικά δύσκολο θα χρησιμοποιηθεί μία λιγότερο πρακτική και περισσότερο θεωρητική προσέγγιση χρησιμοποιώντας την μέθοδο BalancedScorecard (Εξισορροπημένη αξιολόγηση)[38][39]**Error! Reference source not found.**. ToBSCείναι ένα εργαλείο που χρησιμοποιήται από μάνατζερ για την αξιολόγηση της λειτουργίας μίας εταιρίας. Είναι ένα από τα δημοφιλέστερα εργαλεία σε αυτό τον τομέα και παρουσιάστηκε στις αρχές της δεκαετίας του 1990 από τους Drs. Robert S. KaplankαιDavid P. Norton. Από την πρώτη παρουσίαση έχουν βγει βελτιώσεις στο μοντέλο και πλέον θεωρείται 'αναγκαίο καλό' για τις επιχειρήσεις που θέλουν να αξιολογήσουν την λειτουργία τους και την αποδοτικότητά τους.[41]

Αυτό που μπορούμε να κάνουμε είναι να υποθέσουμε ότι έχουμε εφαρμόσει το σύστημά μας σε μία νοσοκομειακή μονάδα και να εξετάσουμε την λειτουργία του νοσοκομείου από τέσσερεις διαφορετικές οπτικές που μας προτείνει η μέθοδος BSC.

- Η οικονομική οπτική αφορά οικονομικούς δείκτες υψηλού επιπέδου χωρίς αναφορές σε περαιτέρω οικονομικές λεπτομέρειες χαμηλότερου επιπέδου. Προσπαθούμε δηλαδή να παρουσιάσουμε μετρήσεις που θα μας δείξουν την εικόνα μας προς τους μετόχους της εταιρίας οι οποίοι ενδιαφέρονται για τα κέρδη που θα παρουσιάσει η εταιρία.
- Η οπτική του πελάτη αντίστοιχα αφορά δείκτες που δείχνουν την εικόνα που έχει η εταιρία στους πελάτες της.
- Η οπτική των εσωτερικών εταιρικών διαδικασιών αφορά δείκτες που μετράνε το κατά πόσο η εταιρία μας έχει επιτύχει στους τομείς στους οποίου πρέπει να είμαστε καλοί.
- Η οπτική της μάθησης και ανάπτυξης αφορά μετρήσεις που προκύπτουν από το ερώτημα 'Πώς μπορούμε να συνεχίσουμε να βελτιωνόμαστε, να δημιουργούμε αξία και να καινοτομούμε'. Κάθε εταιρία πρέπει να λαμβάνει σοβαρά υπόψη της αυτούς τους παράγοντες εάν θέλει να παραμείνει ανταγωνιστική και αξιόπιστη. Για να πετύχει αυτό πρέπει να ακολουθήσει κάποιο πετυχημένο πρόγραμμα εκμάθησης και εκπαίδευσης των υπαλλήλων.



Εικόνα 15Οπτικές της μεθόδου BalancedScorecard(BSC)

Υπολογίζοντας τους δείκτες που αντιπροσωπεύουν τις παραπάνω οπτικές αποκτάμε μια γενική εικόνα για την απόδοση μίας επιχείρησης ή ενός οργανισμού. Η επαγγελματική χρήση της παραπάνω μεθόδου προϋποθέτει περισσότερα βήματα, πιο πολλές μετρήσεις και πιο πολύπλοκες διαδικασίες οι οποίες παρουσιάζονται σε πολλά βιβλία και άρθρα [42]. Παρ' όλα αυτά εμείς θα χρησιμοποιήσουμε την σύντομη και πιο γενική έκδοση καθώς δεν είναι αρμοδιότητά μας το μάρκετινγκ. Η μέθοδος αυτή θεωρείται από της πιο διαδεδομένες και αποδοτικές μεθόδους αξιολόγησης για τους μάνατζερ **Error! Reference source not found.**οπότε αναμφίβολα θα έβρισκε χρήση και στον τομέα της υγείας μιας και τα νοσοκομεία και τα κέντρα υγείας αποτελούν μεγάλους και πολύπλοκους οργανισμούς και φυσικά χρειάζονται και αυτά κάποιου ίδιους αξιολόγηση για να μπορέσουμε να τα βελτιώσουμε. [44]

Παρακάτω θα δούμε την επιρροή του συστήματός μας, σε ένα κέντρο υγείας που θα το εφαρμόσει, από τις τέσσερεις οπτικές που αναφέραμε προηγουμένως.

Οικονομία

Από την οικονομική οπτική παρατηρούμε ότι η εγκατάσταση ενός τέτοιου συστήματος κοστίζει ακριβά καθώς εκτός από την αγορά του λογισμικού θα πρέπει να εγκατασταθεί ασύρματο και ενσύρματο δίκτυο να προμηθευτούν τα γραφεία των εργαζομένων με υπολογιστές ή ταμπλέτες για μεγαλύτερη ευκολία φορητότητα και οικονομία. Παράλληλα όμως ο εκσυγχρονισμός και η ταχύτερη και ποιοτικότερη εξυπηρέτηση θα αποφέρουν περισσότερα κέρδη καθώς η μονάδα θα δέχεται περισσότερους ασθενείς οπότε μακροχρόνια θα καλυφθούν τα έξοδα και θα έχουμε μόνο έσοδα από αυτήν την επένδυση. Επίσης λόγω της καλύτερης διαχείρισης και παρακολούθησης των φαρμάκων θα μειωθούν τα έξοδα και οι δαπάνες για φαρμακευτικές αγωγές και θα μειωθούν τα φαινόμενα υπερκατανάλωσης και άσκοπης χορήγησης περιττών φαρμάκων.

Πελάτης

Από την οπτική του πελάτη δηλαδή του ασθενή η εικόνα της κλινικής αλλάζει προς την θετική κατεύθυνση. Ο εκσυγχρονισμός πάντα δίνει μια καλή εντύπωση και μεγαλύτερη εμπιστοσύνη.Η ταχύτερη και ποιοτικότερη εξυπηρέτηση είναι πολύ θετικά στοιχεία και φυσικά όταν το σύστημα επιτρέπει και στον ίδιο τον ασθενή να έχει ηλεκτρονικά πρόσβαση σε δεδομένα που τον αφορούν σίγουρα ο πελάτης δεν μπορεί παρά να μείνει ικανοποιημένος.

Εσωτερικές εταιρικές διαδικασίες

Η οπτική αυτή αφορά τον τρόπο λειτουργίας της εταιρίας και το κατά πόσο είναι αποτελεσματικός. Η εσωτερική λειτουργία μίας κλινικής που διαθέτει ηλεκτρονικό υπολογιστικό σύστημα γίνεται ταχύτερη καθώς πολλές εντολές όπως η φαρμακευτική αγωγή, οι οδηγίες παρακολούθησης ασθενών για τους νοσοκόμους και οι αιτήσεις για εργαστηριακές εξετάσεις γίνονται ηλεκτρονικά και άμεσα. Επίσης η γρήγορη και άμεση ηλεκτρονική εύρεση του ιστορικού οποιουδήποτε ασθενή μας επιτρέπει να αποφύγουμε την χρονοβόρα διαδικασία που επικρατούσε προηγουμένως με τους φακέλους των ασθενών που βρίσκονταν σε μεγάλες αποθήκες. Όλα αυτά συμβάλλουν στο να είναι το κέντρο καλύτερο και πιο αξιόπιστο στον τομέα της εξυπηρέτησης του πελάτη – ασθενή.

Εκπαίδευση και ανάπτυξη

Η εκπαίδευση και ανάπτυξη σε οποιαδήποτε εταιρία-επιχείρηση-οργανισμό αποτελεί πολύ καίριο ζήτημα ώστε να συνεχιστούν να παρέχονται ποιοτικές υπηρεσίες. Έτσι και ένα κέντρο υγείας για να συνεχίσει να είναι ανταγωνιστικό πρέπει να έχει σωστή εκπαίδευση του προσωπικού και να βελτιώνεται συνέχεια σε όλους του τομείς χρησιμοποιώντας τις δυνατότητες της τεχνολογίας και της ιατρικής εξέλιξης. Μία από τις πολλές δυνατότητες που μας δίνει η τεχνολογία είναι η χρήση ενός ηλεκτρονικού υπολογιστικού συστήματος για νοσοκομεία. Αποτελεί δημιουργία της τεχνολογίας και συμβάλλει στην βελτίωση της λειτουργίας του νοσοκομείου αλλά φυσικά προϋποθέτει σωστή εκπαίδευση των υπαλλήλων για την σωστή χρήση και αξιοποίηση του συστήματος. Το ίδιο το σύστημα μπορεί και αυτό να παρέχει δυνατότητες εκπαίδευσης στους ίδιους τους υπαλλήλους και χρήστες του συστήματος είτε με κάποια εκπαιδευτικά βίντεο είτε με κάποιες εκπαιδευτικές σελίδες στο ίδιο το σύστημα. Εκτός από αυτό το σύστημα μπορεί να χρησιμοποιηθεί σαν εργαλείο συγκέντρωσης και ανάλυσης ιατρικών ιστορικών τα οποία μπορούν να χρησιμοποιηθούν σε μελλοντικές υποθέσεις όπως θα αναφερθεί και παρακάτω στο κεφάλαιο 'DataMining'.

Με βάση τις πληροφορίες που παρουσιάσαμε μπορούμε να κάνουμε μία θεωρητική αξιολόγηση του συστήματος για το πόσο θα βελτιώσει τους τομείς που προαναφέρθηκαν. Η πρακτική αξιολόγηση ενός τέτοιου συστήματος μπορεί να γίνει μόνο με την εφαρμογή και χρήση του σε κάποιο κέντρο υγείας

για τουλάχιστον μερικούς μήνες. Έτσι θα μπορέσουμε να δούμε τον βαθμό προσαρμογής των υπαλλήλων και των ασθενών και να συγκρίνουμε οικονομικούς δείκτες για να δούμε το κατά πόσο πέτυχε το συγκεκριμένο υπολογιστικό σύστημα. Υπάρχουν βεβαίως και περιπτώσεις που η εφαρμογή ενός υπολογιστικού συστήματος σε κάποια νοσοκομεία δεν ήταν και τόσο πετυχημένες μιας και απαιτείταιεξοικείωση των χρηστών με υπολογιστικά συστήματα. Τέτοια παραδείγματα έχουμε από χώρες όπως είναι η Νότια Αφρική όπου παρουσιάστηκαν δυσκολίες και πολλά προβλήματα που δεν είχαν προβλεφθεί **Error! Reference source not found.**. Κάποιοι από τους παράγοντες που οδήγησαν στην αποτυχία της εφαρμογής των υπολογιστικών συστημάτων ήταν η έλλειψη υποδομών, αποθηκευτικών χώρων για τους υπολογιστές ακόμα και η ελλιπής παροχή ρεύματος για τα συστήματα. Επίσης υποτιμήθηκε η πολυπλοκότητα που έχει η εφαρμογή ενός τέτοιου συστήματος και η εξοικείωση με την χρήση του παρόλο που πρόκειται για μεγάλη αλλαγή στον τρόπο λειτουργίας των κέντρων υγείας. Παρόμοια προβλήματα είχαμε και στην εφαρμογή των υπολογιστικών συστημάτων στα Ελληνικά κέντρα υγείας μιας και είχαμε και εδώ πολλές παρατυπίες καθώς δεν μάθαμε από τα λάθη που έγιναν σε άλλες χώρες. Παρόλαυτά και παρά τις δυσκολίες τα περισσότερα κέντρα υγείας στην Ελλάδα διαθέτουν πλέον υπολογιστικό σύστημα αν και σε πολλές περιπτώσεις δεν εκμεταλλευόμαστε όλες τις δυνατότητές που θα μπορούσαν να μας παρέχουν.

Παρακάτω θα παρουσιάσουμε τον πίνακα 3 με θεωρητικές μετρήσεις σε στρατηγικούς στόχους που θέλει να βελτιώσει το κάθε κέντρο υγείας. Θα δούμε ότι θα χωρίσουμε τους στρατηγικούς στόχους σε 4 κατηγορίες οι οποίες είναι οι οπτικές που αναφέραμε προηγουμένως. Στην συνέχεια για την κάθε στρατηγική θα παρουσιάσουμε τον επιθυμητό στόχο, την θεωρητική απόδοση που μπορούμε να πετύχουμε με το σύστημά μας, το ποσοστό επιτυχίας ανάμεσα στην απόδοση και τον στόχο μας, το βάρος που έχει ο κάθε στόχος και, τέλος, υπολογίζουμε το τελικό σκορ της κάθε στρατηγικής που προκύπτει από την επιτυχία και το βάρος της κάθε στρατηγικής που μελετήσαμε. Οι μετρήσεις είναι θεωρητικές και υποθετικές και για μεγαλύτερη ακρίβεια πρέπει να εφαρμοστεί το σύστημα σε κάποιο κέντρο υγείας και να γίνουν πρακτικές και πραγματικές μετρήσεις μετά την χρήση του συστήματος για μεγάλο χρονικό διάστημα. Το τελικό σκορ που προκύπτει 76,06 θα μπορούσαμε να πούμε είναι αρκετά ικανοποιητικό για να θεωρήσουμε το υπολογιστικό σύστημα πετυχημένο. Από την θεωρία όμως στην πράξη έχουμε πολλά βήματα ακόμα.

Οπτική	**Στρατηγική**	**Στόχος**	**Θεωρητική απόδοση**	**Επιτυχία**	**Βάρος**	**Balanced Score**
Οικονομική	Μείωση Κόστους Νοσηλείας	5%	4%	80%	15%	12%
	Μείωση Κόστους Φαρμάκων	5%	4%	80%	10%	8%
Πελάτη	Βελτιωμένη και ακριβέστερη διάγνωση	10%	7%	70%	15%	10,5%
	Ικανοποίηση πελατών	10%	9%	90%	10%	9%
	Διευκόλυνση πρόσβασης σε ιατρικά δεδομένα	70%	50%	71,42%	5%	3,57%
Εσωτερικές εταιρικές διαδικασίες	Ταχύτερη εσωτερική επικοινωνία	10%	7%	70%	5%	3,5%
	Επιτυχία θεραπείας	3%	2%	66,66%	15%	9,99%
Εκπαίδευση	Χρόνος	10%	8%	80%	15%	12%

και ανάπτυξη	παραμονής ασθενή στο νοσοκομείο					
	Εκπαίδευση εργαζομένων	8%	6%	75%	10%	7,5%
Σύνολο					100%	**76,06**

Πίνακας 3 Κάρτα Εξισορροπημένης αξιολόγησης (BalancedScorecard)

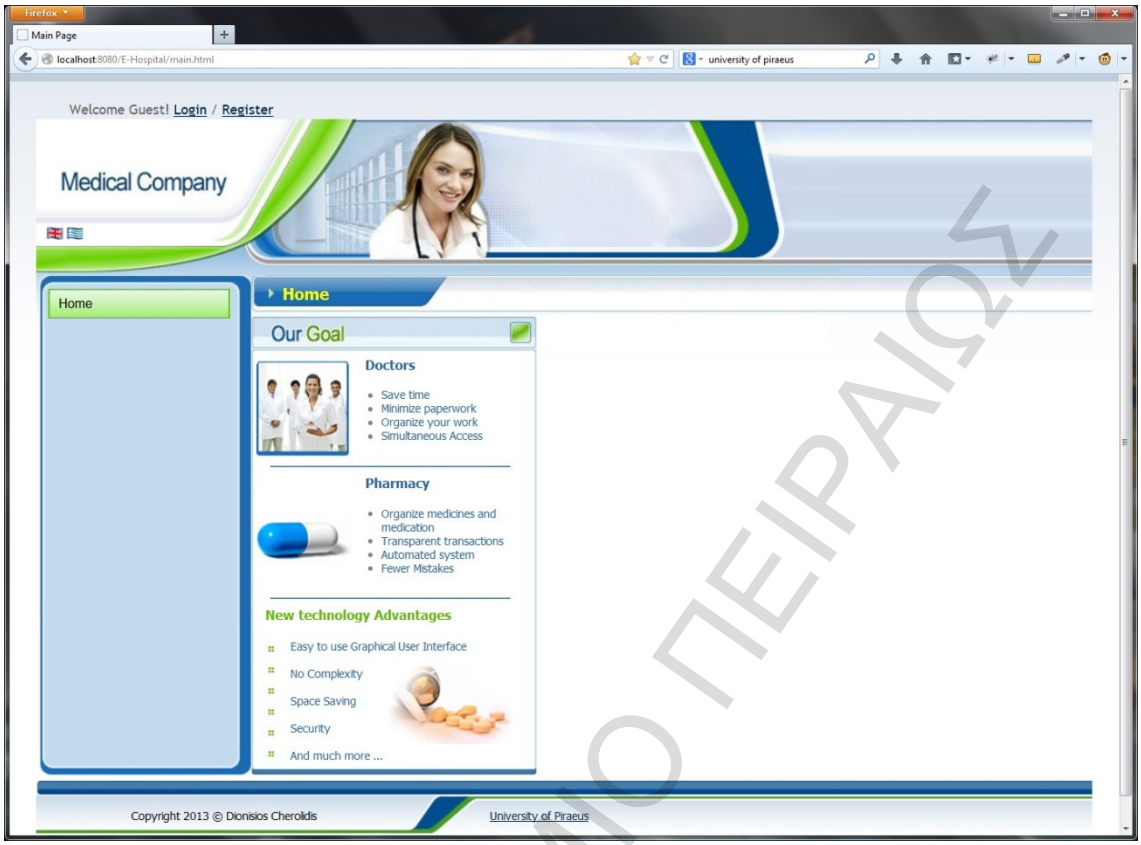
5. Παρουσίαση της εφαρμογής

Σε αυτό το κεφάλαιο θα παρουσιάσουμε την λειτουργικότητα της εφαρμογής και θα δούμε στην πράξη τις δυνατότητες που έχει ο κάθε χρήστης. Βοηθητικά θα υπάρξουν και screenshotsγια μεγαλύτερη ευκολία στην κατανόηση.

5.1 Γενικά

Η αρχική σελίδα που βλέπουμε όταν εισάγουμε την διεύθυνση στην οποία βρίσκεται η εφαρμογή φαίνεται στο σχήμα 13 και εκτός από τις γενικές πληροφορίες που βλέπουμε έχουμε τρείς ακόμα λειτουργίες:

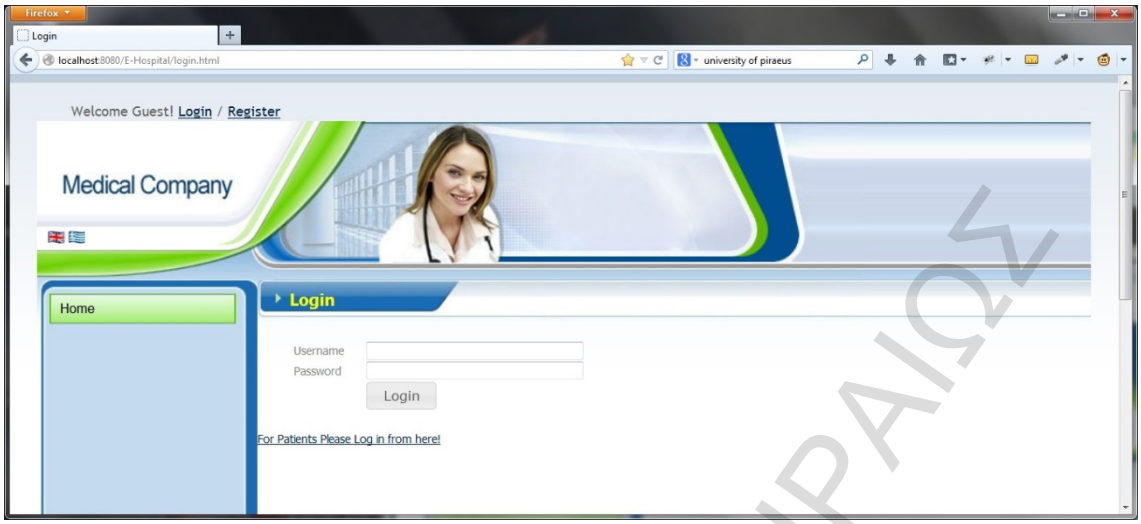
- Μετάβαση στην σελίδα Εισόδου στο σύστημα: Πάνω αριστερά κάνοντας κλικ στην επιλογή "Login" μεταβαίνουμε στην σελίδα εισόδου στο σύστημα.
- Μετάβαση στην σελίδα Εγγραφής στο σύστημα: Πάνω αριστερά κάνοντας κλικ στην επιλογή "Register" μεταβαίνουμε στην σελίδα εγγραφής στο σύστημα
- Αλλαγή γλώσσας: Πάνω αριστερά κάτω από τις επιλογές εγγραφής και εισόδου βλέπουμε δύο μικρές σημαίες, μία αγγλική και μία ελληνικά οι οποίες αντιστοιχούν στην γλώσσα που θέλουμε για να βλέπουμε στην εφαρμογή μας. Επιλέγοντας μία από τις δυο σημαιούλες αλλάζουμε την γλώσσα στα αγγλικά ή τα ελληνικά.



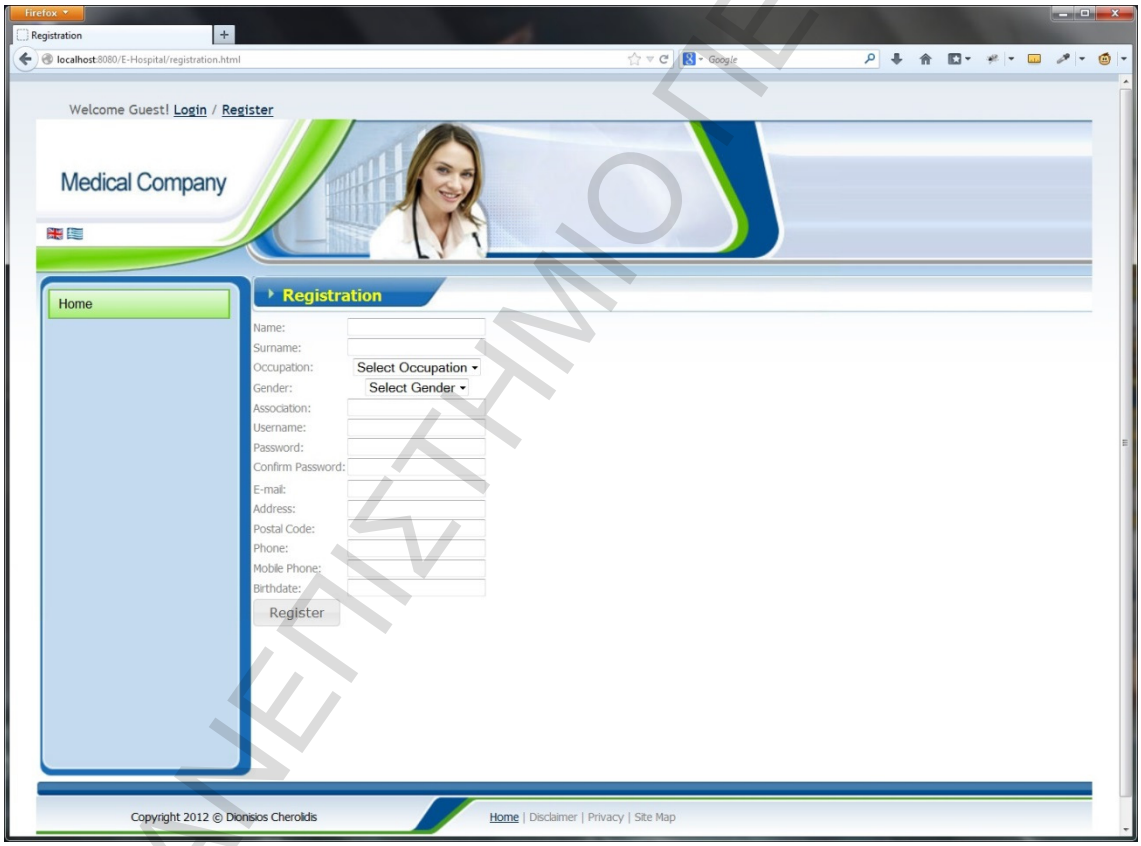
Εικόνα 16Αρχική σελίδα

Στην σελίδα εισόδου στο σύστημα εισάγουμε το όνομα χρήστη και τον κωδικό για να εισέλθουμε σαν διαχειριστής, ιατρός, νοσηλευτής, φαρμακοποιός ή εργαστηριακός ιατρός (Σχήμα 14). Ενώ εάν θέλουμε να εισέλθουμε σαν ασθενής επιλέγουμε την επιλογή " For PatientsPleaseLog in fromhere!" και εισάγουμε τα στοιχεία μας στην καινούρια φόρμα. Επιλέγοντας "Login" αφού γίνει επιβεβαίωση των στοιχείων μας εισερχόμαστε στο σύστημα.

Στην εικόνα 18 βλέπουμε την φόρμα εγγραφής στο σύστημα αυτή η φόρμα δεν είναι για ασθενείς αλλά για διαχειριστή, ιατρό, νοσηλευτή, φαρμακοποιό ή εργαστηριακό ιατρό. Αφού κάνει την εγγραφή πρέπει να περιμένει έγκριση από την διαχειριστή του συστήματος για να ενεργοποιηθεί ο λογαριασμός και να εισέλθει στο σύστημα.

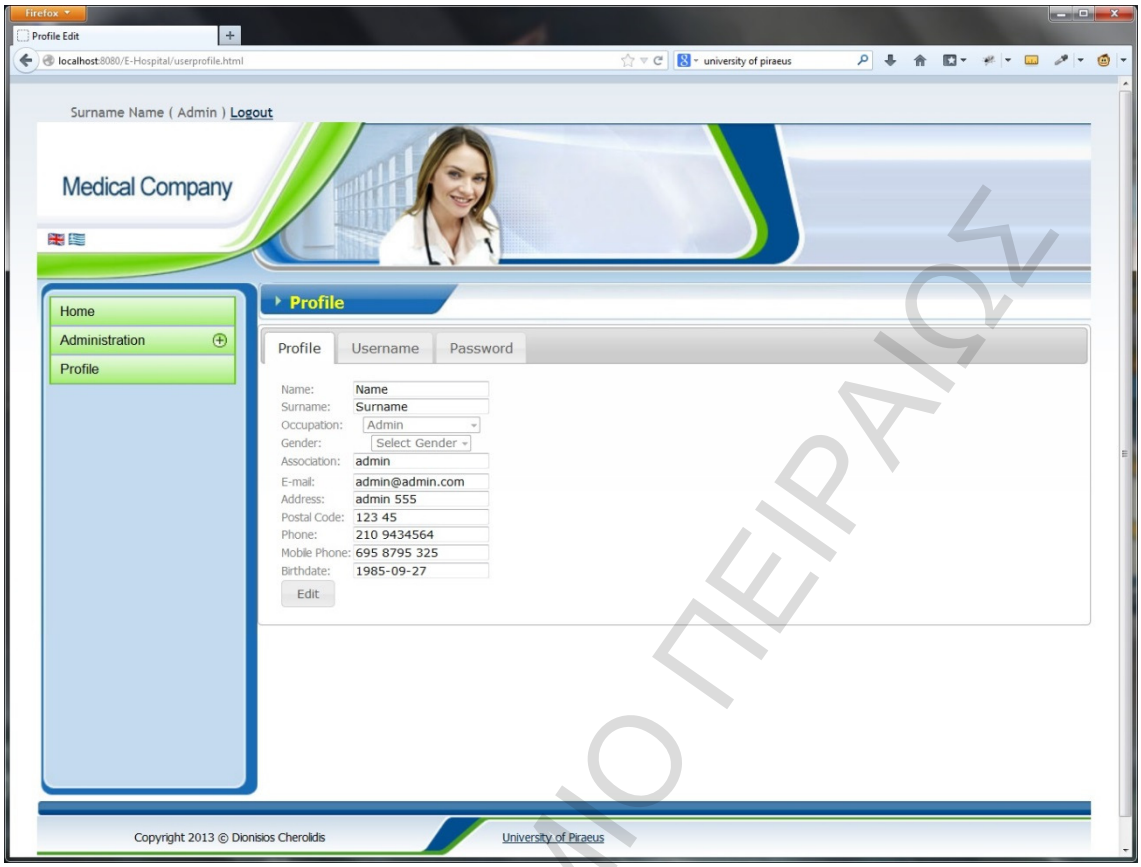


Εικόνα 17Είσοδος στο σύστημα



Εικόνα 18 Εγγραφή χρήστη

Αφού κάποιος χρήστης εισέλθει στην εφαρμογή, σε όποια ομάδα χρηστών και να ανήκει, εμφανίζεται το μενού "Profile" (Εικόνα 19)όπου μπορεί να επεξεργαστεί τα προσωπικά του στοιχεία αλλά και το όνομα χρήστη και ο κωδικός που χρησιμοποιεί για την είσοδο.

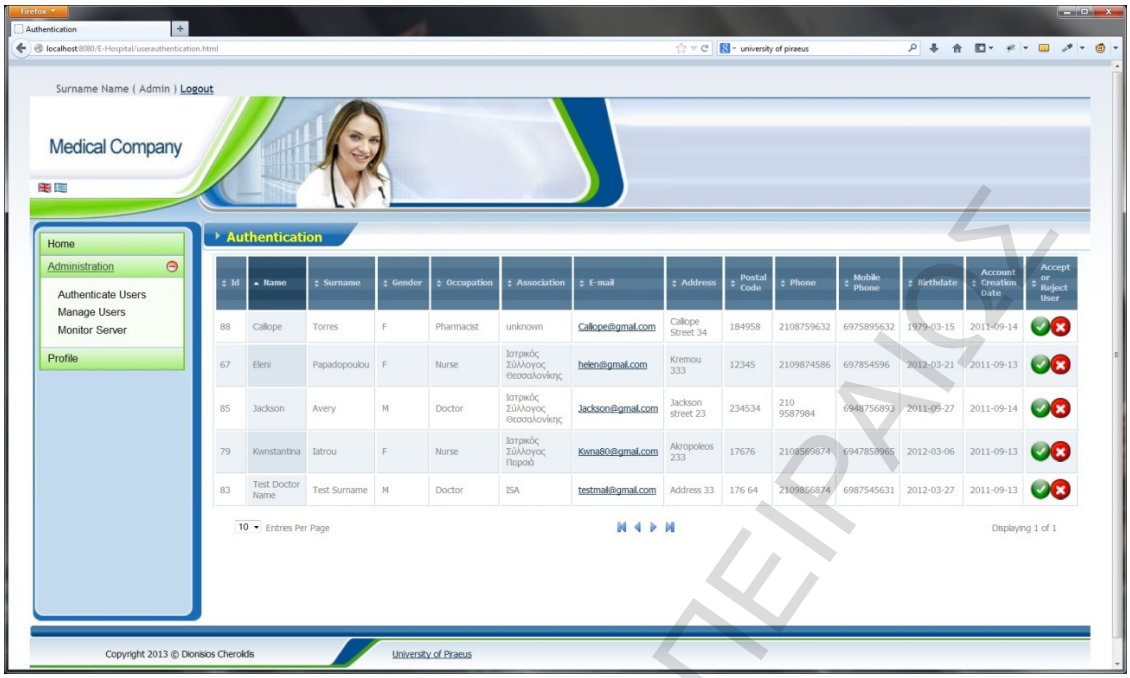


Εικόνα 19 Προφίλ Χρήστη

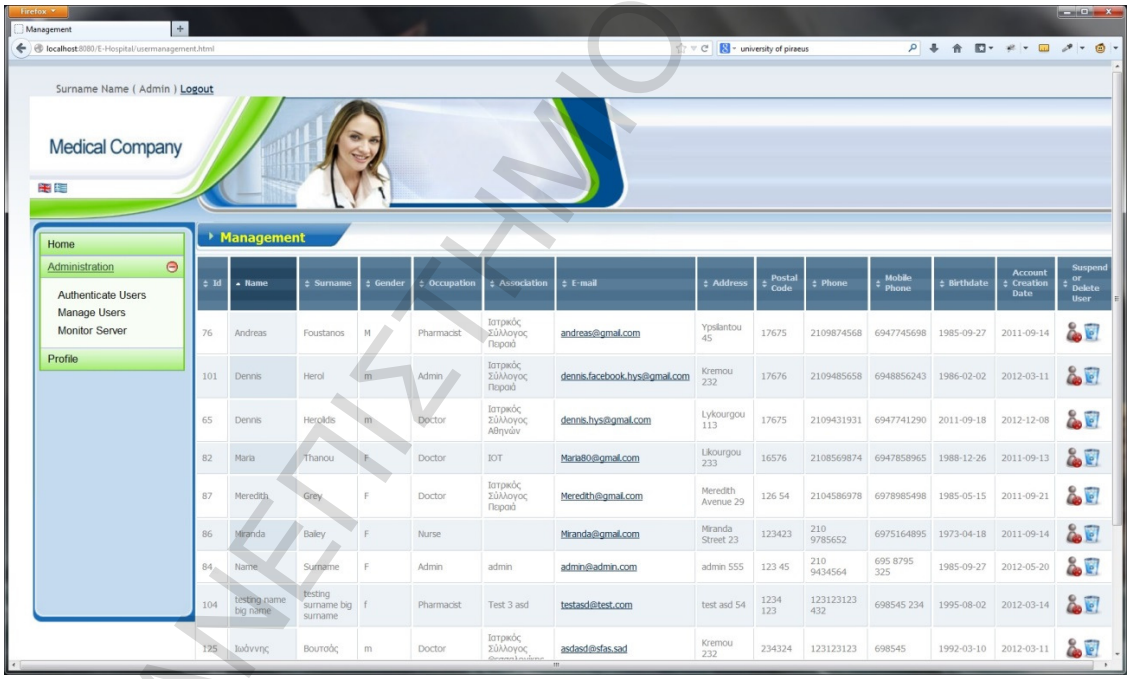
5.2 Διαχειριστής

Ο Διαχειριστής είναι αυτός που διαχειρίζεται τους χρήστες του συστήματος και τους παρέχει πρόσβαση. Μετά από κάθε εγγραφή χρήστη ο διαχειριστής πρέπει να επιβεβαιώσει τα στοιχεία και να ενεργοποιήσει τον λογαριασμό. Σε αντίθετη περίπτωση εάν δεν είναι σωστά τα στοιχεία μπορεί να απορρίψει τον χρήστη από το σύστημα διαγράφοντάς τον. Αυτό το πετυχαίνει επιλέγοντας από το αριστερό μενού το "Administration" και στην συνέχεια "AuthenticateUsers" και θα δει την σελίδα που βλέπουμε στην εικόνα 20.

Η άλλη επιλογή που έχει ο διαχειριστής είναι το "ManageUsers" όπου μπορεί να διαχειριστεί τους χρήστες που έχουν ήδη αυθεντικοποιηθεί και χρησιμοποιούν το σύστημα. Από εκεί μπορεί να απενεργοποιήσει κάποιον λογαριασμό προσωρινά ή να τον διαγράψει μόνιμα από το σύστημα. (Εικόνα 21)



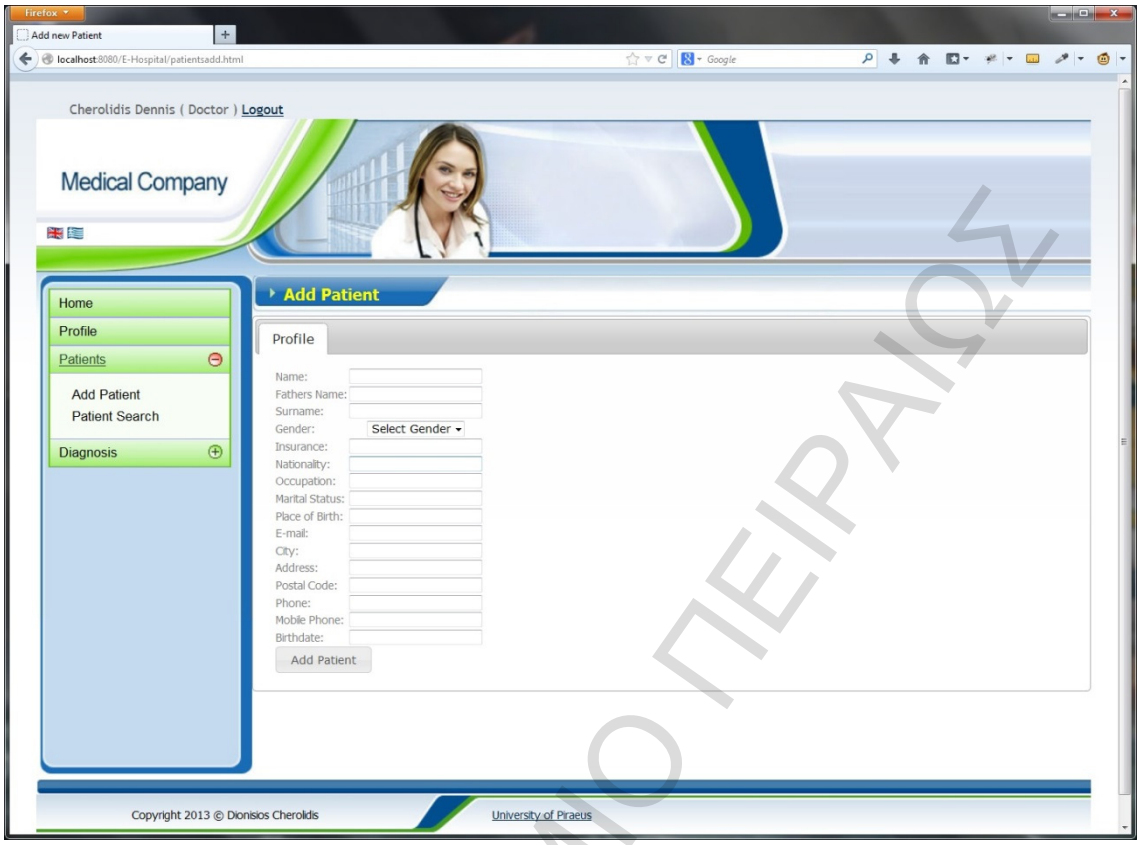
Εικόνα 20 Ενεργοποίηση - Διαγραφή χρήστη



Εικόνα 21Διαχείριση Χρήστη

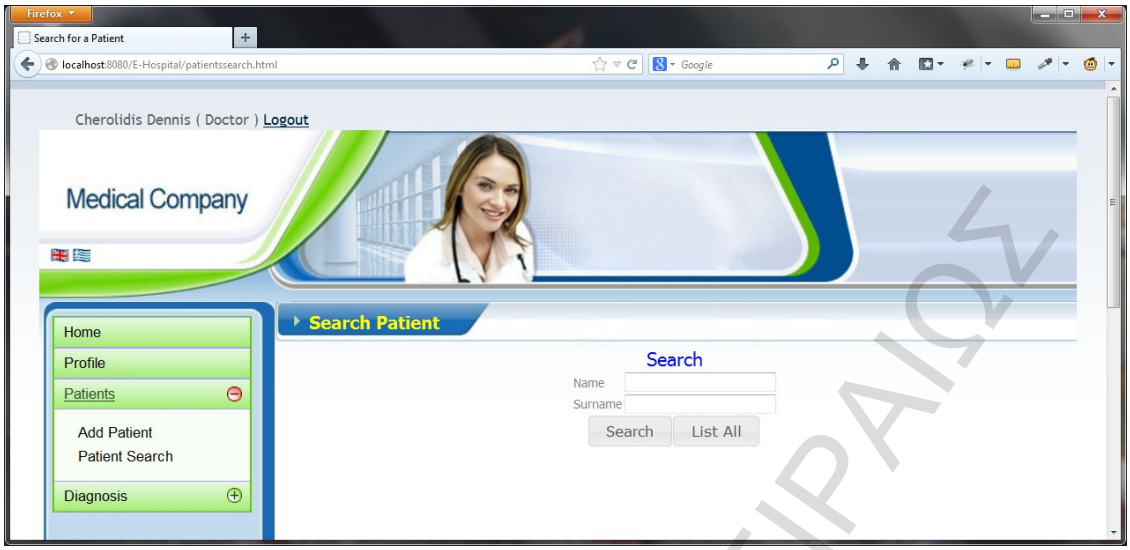
5.3 Ιατρός

Ο Ιατρός είναι αυτός που διαχειρίζεται τους ασθενείς και τις καρτέλες με το ιστορικό τους. Η πρώτη δυνατότητα που έχουν είναι η προσθήκη νέου ασθενή στο σύστημα. Αυτό γίνεται επιλέγοντας το μενού “Patients” στην συνέχεια “AddPatient” και συμπληρώνουμε την φόρμα που εμφανίζεται όπως βλέπουμε και στην εικόνα 22. Αφού προσθέσουμε τον ασθενή θα μεταβούμε στην καρτέλα ασθενή και θα δούμε τα στοιχεία εισόδου του χρήστη στο σύστημα τα οποία πρέπει να δοθούν στην ασθενή σε περίπτωση που θελήσει να δει το προφίλ του.

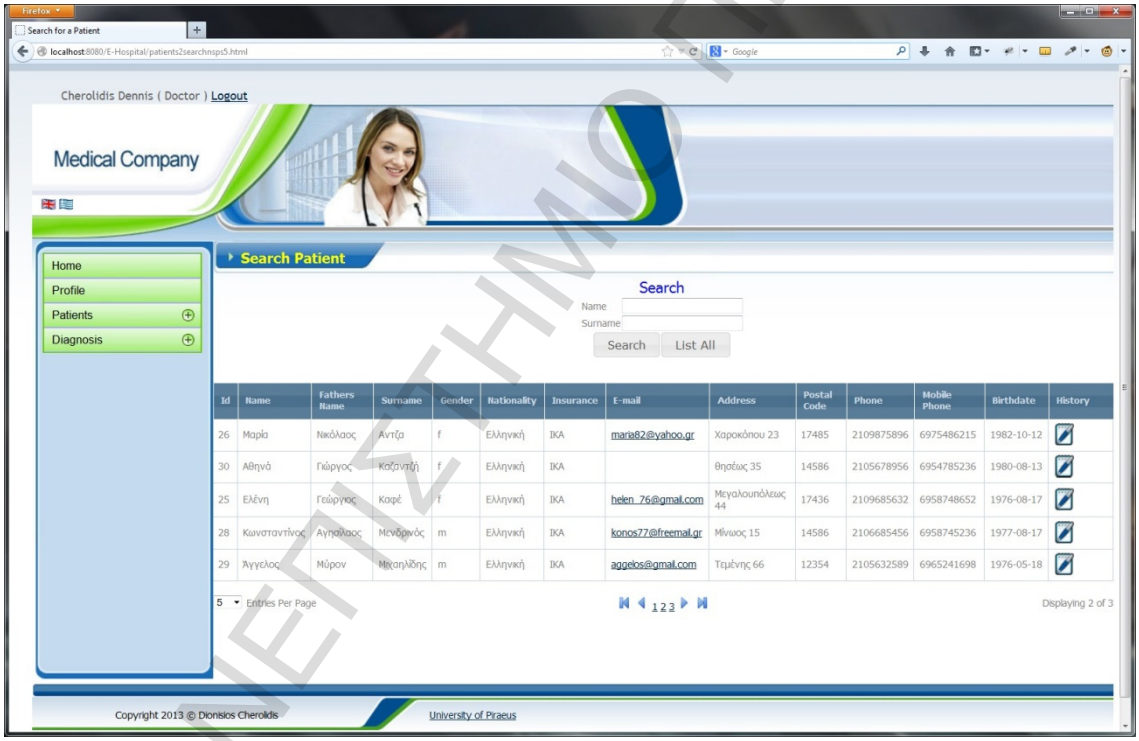


Εικόνα 22Προσθήκη Ασθενή

Στην συνέχεια έχουμε την επιλογή αναζήτησης ασθενή από το αριστερό μενού “PatientSearch” το οποίο μας οδηγεί στην εικόνα 23 όπου μπορούμε να αναζητήσουμε τον ασθενή με βάση το όνομά του και το επίθετό του. Αλλιώς μπορεί να επιλέξει το “ListAll” για να εμφανίσουμε όλους τους ασθενείς που έχουμε στο σύστημά μας όπως βλέπουμε στην εικόνα 24. Ακόμα και εάν είναι πολλές χιλιάδες τα αποτελέσματα δεν θα έχουμε προβλήματα ανταπόκρισης καθώς έχει σχεδιαστεί έτσι ώστε να εμφανίζει κάθε φορά από 5 μέχρι 100 αποτελέσματα ανάλογα με την επιλογή του χρήστη. Τα υπόλοιπα θα χωριστούν σε σελίδες και θα καλούνται τμηματικά ανάλογα με την σελίδα που θα επιλέξει ο χρήστης. Για να δούμε τον φάκελο κάποιου ασθενή τον εντοπίζουμε στην λίστα μας και επιλέγουμε την εικόνα με το σημειωματάριο που βρίσκετε στην στήλη “History”.



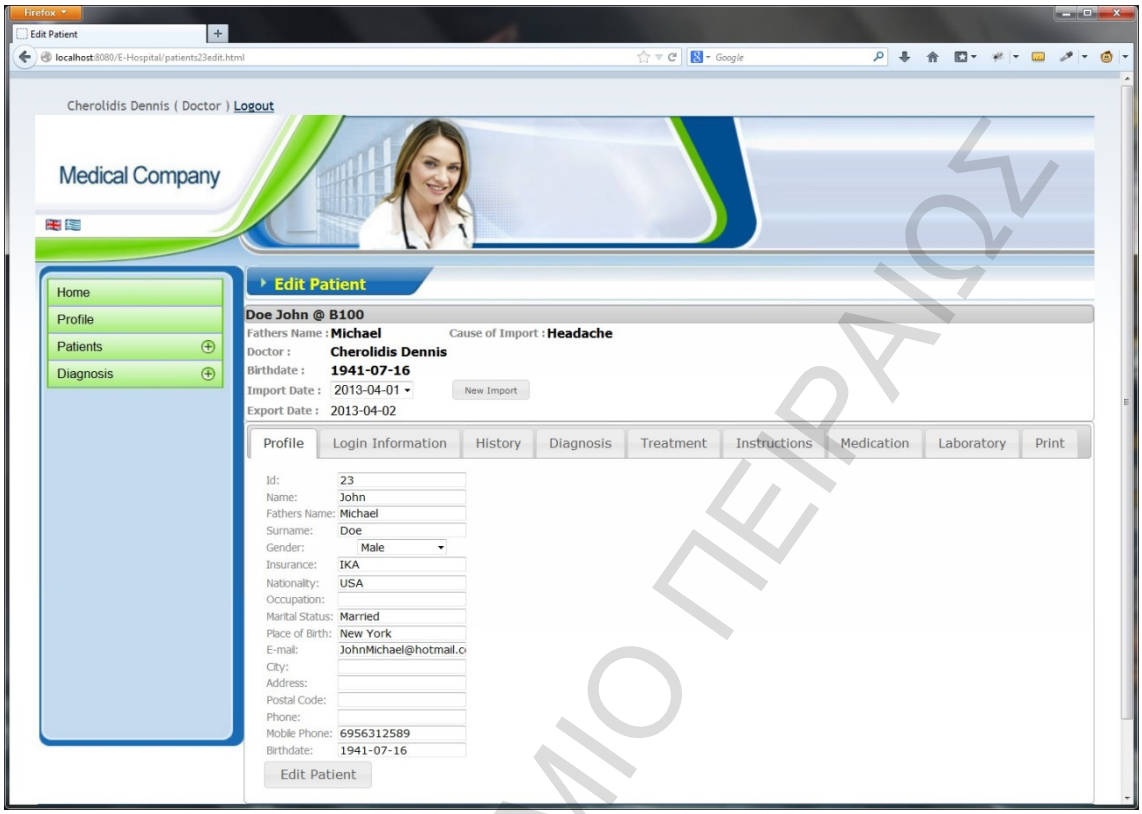
Εικόνα 23 Αναζήτηση ασθενή



Εικόνα 24 Αποτέλεσμα αναζήτησης ασθενή

Αφού επιλέξουμε να ανοίξουμε τον ηλεκτρονικό φάκελο κάποιου ασθενή θα μεταβούμε στην σελίδα που βλέπουμε στην εικόνα 25. Εκεί στο πρώτο τμήμα βλέπουμε στην πρώτη σειρά το όνομα του ασθενή και το δωμάτιο στο οποίο βρίσκεται εάν έχει γίνει εισαγωγή και στην συνέχεια έχουμε κάποια γενικά στοιχεία όπως το όνομα πατρός, την αιτία εισαγωγής, τον θεράπων ιατρό, την ημερομηνία γέννησης και τις ημερομηνίες εισαγωγής και εξόδου του ασθενή από το νοσοκομείο. Οι ημερομηνίες εισαγωγής είναι μία λίστα στην οποία επιλέγοντας κάποια από τις επιλογές που έχει θα ανοίξουμε το ιστορικό που αφορά την συγκεκριμένη εισαγωγή του ασθενή στο νοσοκομείο. Ενώ για να δημιουργήσουμε μία νέα εισαγωγή επιλέγουμε το "NewImport" και έτσι ορίζεται η τρέχουσα ημέρα σαν

ημέρα νέας εισαγωγής στο νοσοκομείο και δημιουργείται νέο ιστορικό για την συγκεκριμένη εισαγωγή το οποίο ο ιατρός μπορεί να συμπληρώσει.



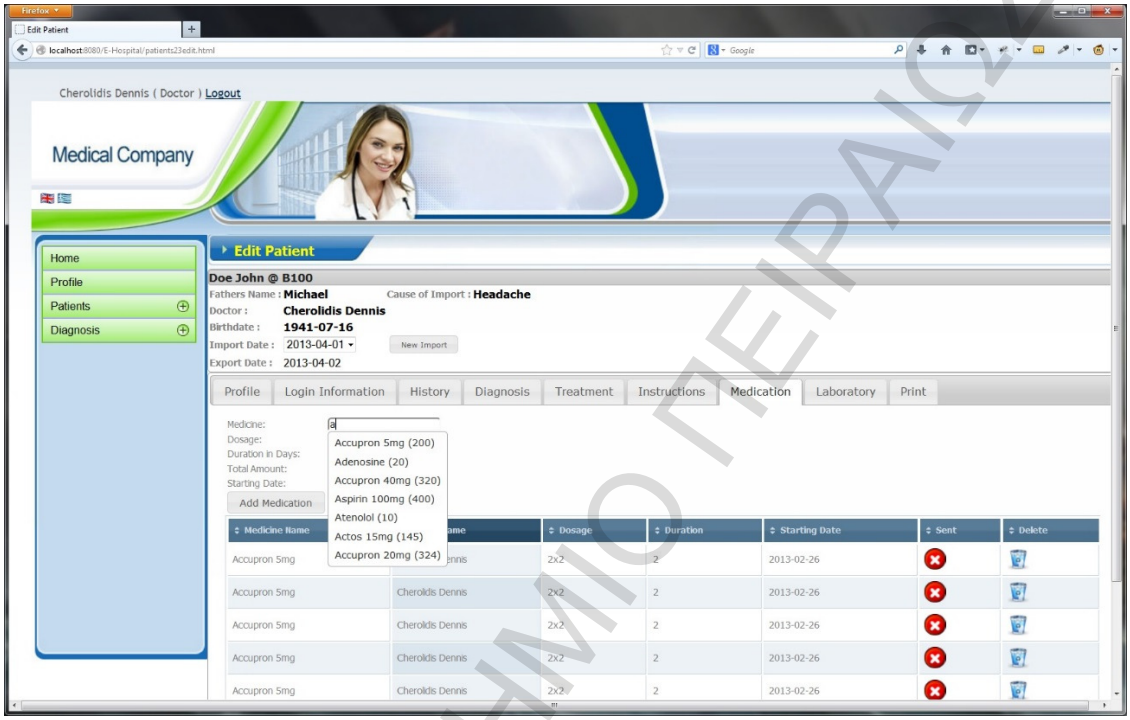
Εικόνα 25 Καρτέλα ασθενή

Για κάθε εισαγωγή έχουμε νέα σειρά καρτελών.Οι πληροφορίες που διαθέτουν είναι οι έξης:

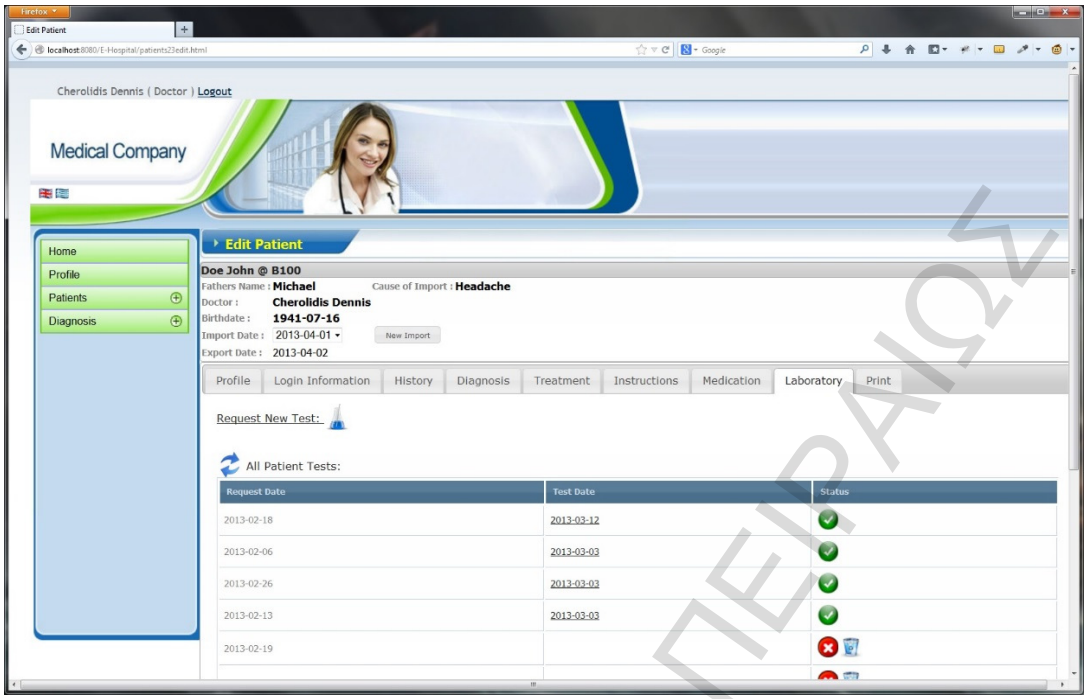
- Profile: Το προφίλ του ασθενή με τα προσωπικά του στοιχεία.
- LoginInformation: Στοιχεία εισόδου στο σύστημα για τον ασθενή όπου μπορεί να γίνει και η αλλαγή του ονόματος χρήστη και του κωδικού εάν το επιθυμεί ο ασθενής.
- History:Το ιστορικό της ασθένειας του ασθενή όπου έχουμε πληροφορίες όπως αιτία εισαγωγής αριθμός κλίνης, ημερομηνία εισαγωγής, ημερομηνία εξαγωγής, Περιγραφή της ασθένειας και πορεία της ασθένειας.
- Diagnosis: Η διάγνωση που έχει κάνει ο ιατρός
- Treatment: Η θεραπεία που έχει προτείνει ο ιατρός να ακολουθηθεί.
- Instructions: Διάφορες οδηγίες οι οποίες είναι κυρίως για νοσηλευτές και νοσοκόμους που περιθάλπουν τον ασθενή.
- Medication:Το ιστορικό των φαρμάκων που έχει λάβει ή λαμβάνει ο ασθενής. Στο πεδίο ‘medicine’ ο ιατρός πληκτρολογεί το όνομα του φαρμάκου και εμφανίζονται σε λίστα τα προτεινόμενα φάρμακα που μπορεί να επιλέξει ενώ σε παρένθεση εμφανίζεται η διαθεσιμότητα του κάθε φαρμάκου (εικόνα 26). Από κάτω συμπληρώνει την διάρκεια που θα χορηγείτε το φάρμακο και την δοσολογία για να υπολογιστεί αυτόματα η ποσότητα που θα χρειαστεί συνολικά για να γίνει η παραγγελία στο φαρμακείο του νοσοκομείου.
- Laboratory: Εδώ ο ιατρός μπορεί να ζητήσει εργαστηριακές εξετάσεις για τον ασθενή από τα εργαστηριακά τμήματα του νοσοκομείου (εικόνα 27). Εδώ βλέπουμε όλες τις εξετάσεις που έχουν γίνει ή έχουν προγραμματιστεί να γίνουν στις ημερομηνίες που έχουν οριστεί από τον ιατρό. Ο ιατρός μπορεί είτε να δει τα αποτελέσματα επιλέγοντας την ημερομηνία μίας παλιότερης εξέτασης είτε να ζητήσει μία νέα εργαστηριακή εξέταση επιλέγοντας

'RequestNewTest'. Στο νέο παράθυρο που θα ανοίξει (εικόνα 28) ο ιατρός επιλέγει την ημερομηνία που επιθυμεί να γίνουν οι εξετάσεις και στην συνέχεια επιλέγει το εργαστήριο που επιθυμεί και τις εξετάσεις που πρέπει να πραγματοποιηθούν.

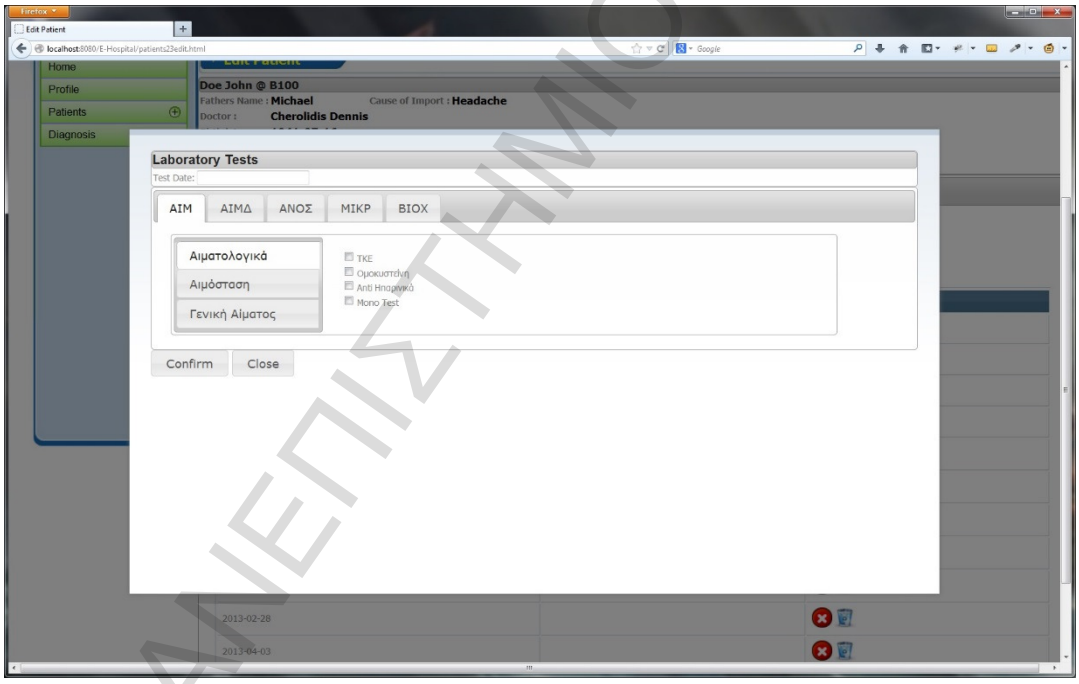
- Print:Στην τελευταία καρτέλα μπορεί να εκτυπώσει τμήματα των πληροφοριών που διαθέτει η καρτέλα του ασθενή είτε για να δοθούν στον ασθενή είτε για οποιαδήποτε άλλη χρήση. Έχει την δυνατότητα να αποκρύψει κάποιες πληροφορίες επιλέγοντας μόνο αυτά που χρειάζονται.



Εικόνα 26Φαρμακευτική αγωγή ασθενή



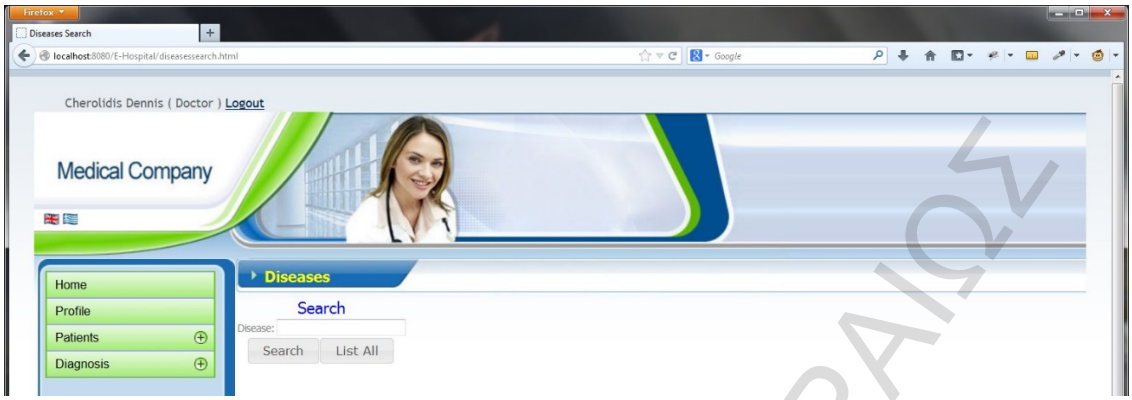
Εικόνα 27 Καρτέλα εξετάσεων ασθενή



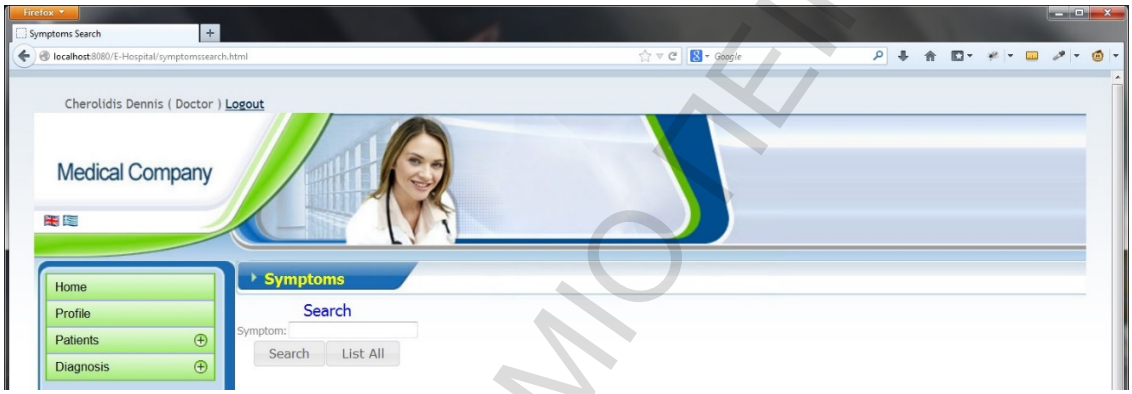
Εικόνα 28Καρτέλα για αίτηση εργαστηριακών εξετάσεων ασθενή

Στην συνέχεια έχουμε το μενού ‘Diagnosis’ το οποίο αφορά μόνο τους ιατρούς και είναι ένα βοηθητικό εργαλείο διάγνωσης ασθενειών με βάση τα συμπτώματα. Το εργαλείο αυτό λειτουργεί σαν μία μεγάλη εγκυκλοπαίδεια με πληροφορίες που θα εισαγάγουν οι ίδιοι οι ιατροί για να την εμπλουτίσουν. Για να προσθέσει κάποιος μία ασθένεια στην λίσταπρέπει πρώτα να ελέγξει μήπως υπάρχει ήδη κάνοντας αναζήτηση στο ‘searchdiseases’ (εικόνα 29). Στην συνέχεια θα πρέπει να ελέγξει εάν όλα τα συμπτώματά της υπάρχουν στην βάση δεδομένων, μέσω του ‘searchsymptoms’ (εικόνα 30), και όσα δεν υπάρχουν πρέπει να τα προσθέσει ο ίδιος, μέσω του ‘addsymptom’(εικόνα 31). Αφού γίνει αυτό προσθέτει και την

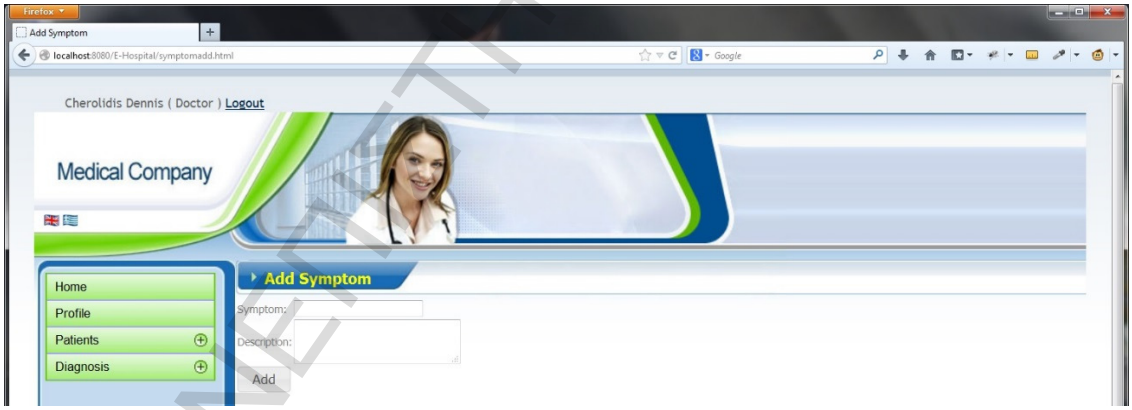
ασθένεια που επιθυμεί από το μενού ‘adddisease’ (εικόνα 32). Η προσθήκη ασθένειας γίνεται με την αντιστοίχησή της με όλα τα πιθανά συμπτώματα που παρουσιάζονται.



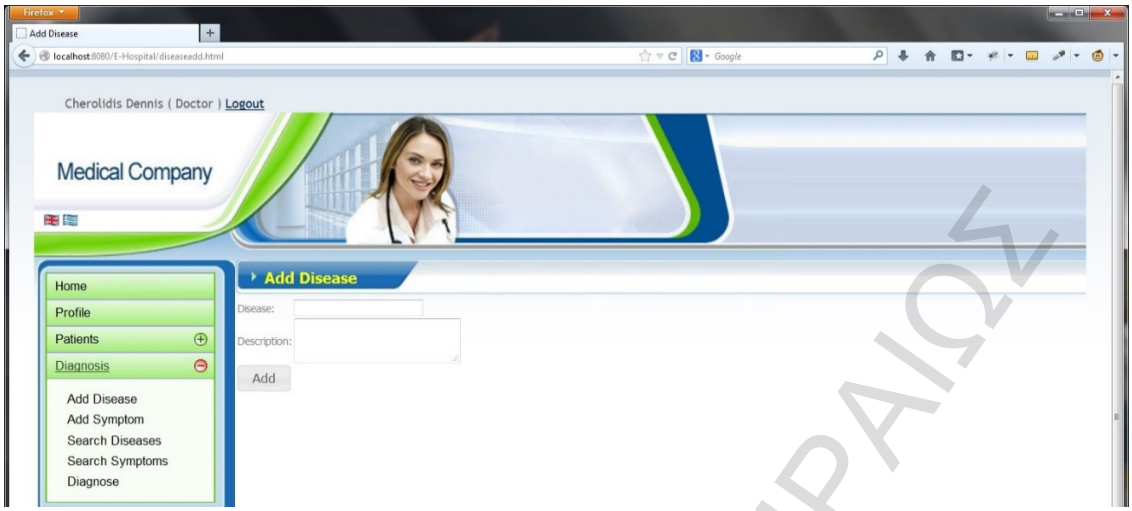
Εικόνα 29 Αναζήτηση ασθένειας



Εικόνα 30 Αναζήτηση συμπτώματος

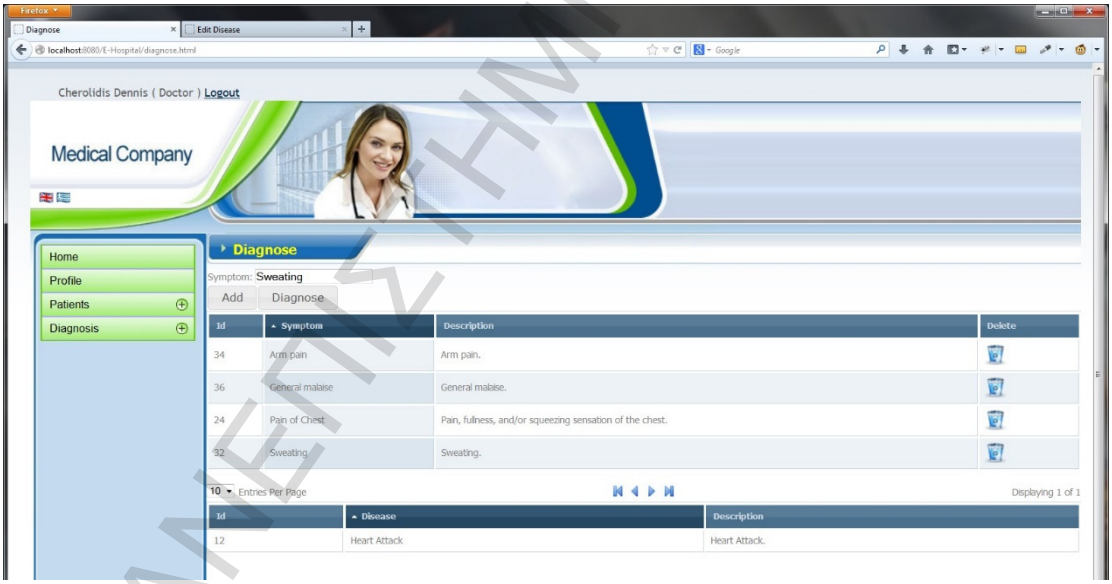


Εικόνα 31 Προσθήκη συμπτώματος



Εικόνα 32 Προσθήκη Ασθένειας

Ο λόγος που δημιουργούμε αυτή την βιβλιοθήκη είναι ώστε να μπορέσουμε να κάνουμε ηλεκτρονικά βοηθητικές διαγνώσεις σε ασθενείς με βάση τα συμπτώματα που παρουσιάζουν. Στην εικόνα 33 βλέπουμε την λειτουργεία αυτού του εργαλείου το οποίο ανοίγει από το μενού 'Diagnose'. Εδώ ο ιατρός προσθέτει τα συμπτώματα που παρουσιάζει ο ασθενής και μόλις επιλέξει 'Diagnose' θα ξεκινήσει η λειτουργία αντιστοίχησης των συμπτωμάτων με τις ασθένειες που έχουμε στην βάση δεδομένων μας και θα παρουσιαστούν όλες οι πιθανές ασθένειες που παρουσιάζουν όλα τα συμπτώματα που επέλεξε προηγουμένως ο ιατρός.

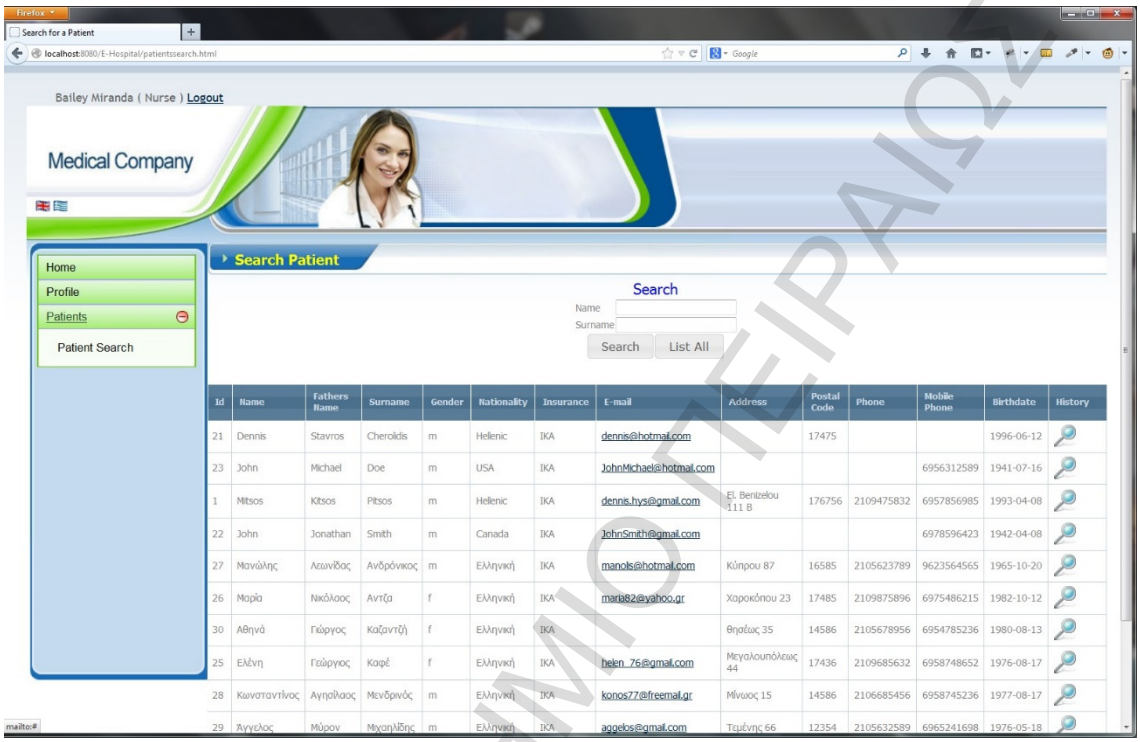


Εικόνα 33 Διάγνωση ασθένειας με βάση τα συμπτώματα

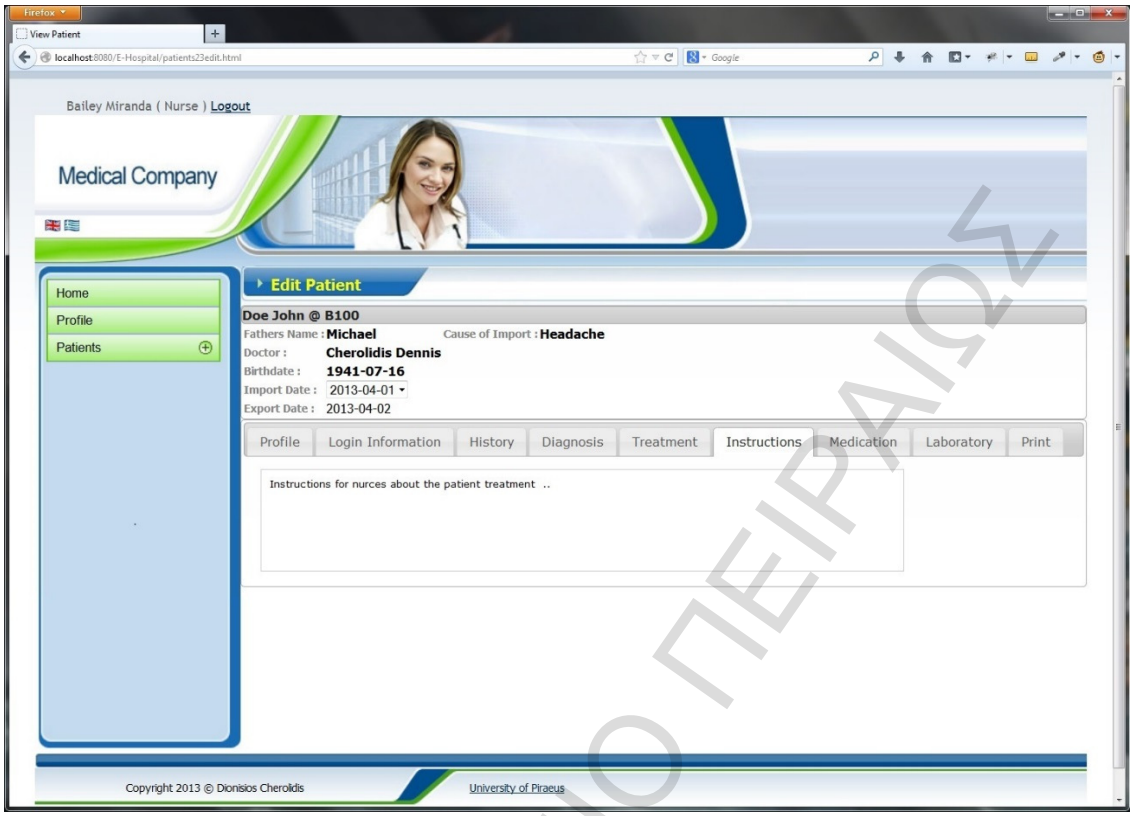
5.4 Νοσοκόμος-Νοσηλευτής

Οι νοσοκόμοι και οι νοσηλευτές έχουν την ευθύνη να φροντίζουν τους ασθενείς και να εκτελούν τις οδηγίες του ιατρού. Στην εφαρμογή μας οι οδηγίες αυτές καταγράφονται ηλεκτρονικά στον φάκελο του ασθενή από τον ιατρό και όταν χρειαστεί ο υπεύθυνος νοσοκόμος η νοσηλευτής αναζητά τον ασθενή από το μενού 'PatientSearch' όπως βλέπουμε και στην εικόνα 34. Αφού εντοπιστεί ο ασθενής επιλέγοντας το εικονίδιο στην στήλη 'History' ανοίγουμε το ιστορικό του ασθενή το οποίο όμως δεν είναι πλήρως διαθέσιμο και δεν δίνεται η δυνατότητα τροποποίησης του περιεχομένου μιας και αυτό είναι αρμοδιότητα

του ιατρού. Το τμήμα του ιστορικού που είναι διαθέσιμο μπορεί να διαφέρει από νοσοκομείο σε νοσοκομείο και αυτό μπορεί να οριστεί εύκολα προγραμματιστικά χάρις το πλαίσιο 'SpringSecurity'. Στις καρτέλες που περιέχονται μπορούμε να δούμε διάφορες πληροφορίες στις οποίες έχουμε πρόσβαση και αφορούν τον ασθενή μας και στην καρτέλα 'Instructions' (εικόνα 35) έχουμε τις οδηγίες του ιατρού για τον συγκεκριμένο ασθενή.



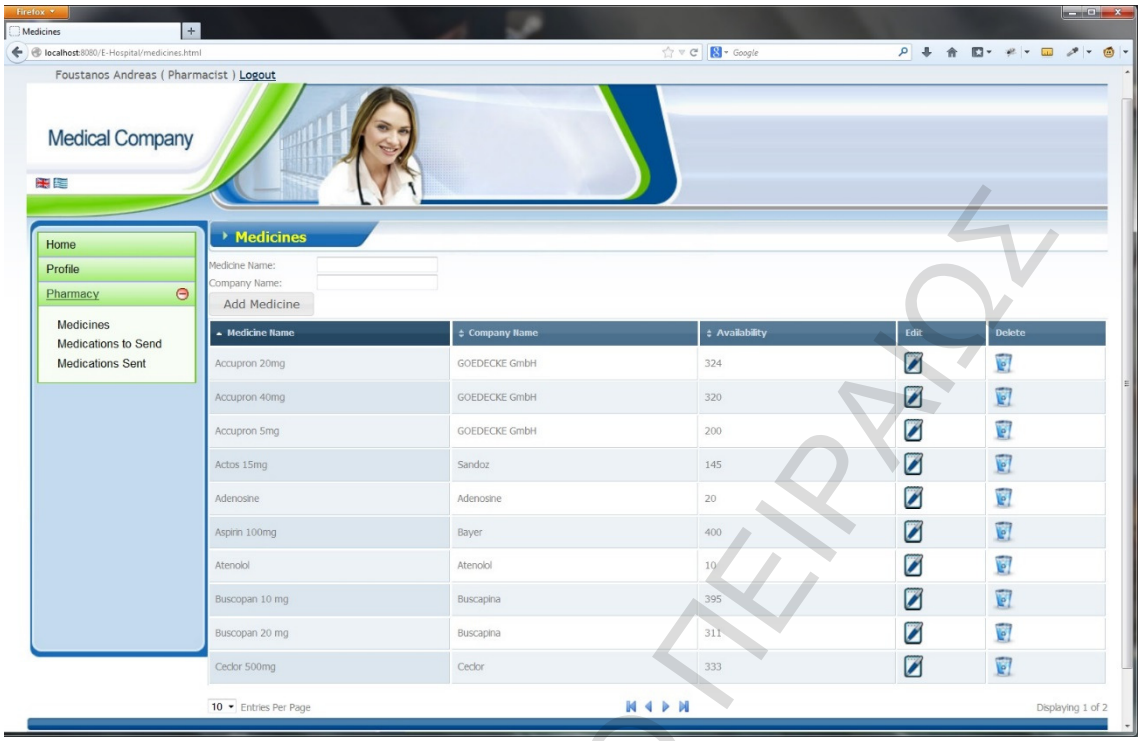
Εικόνα 34 Αναζήτηση ασθενή από νοσοκόμο ή νοσηλευτή.



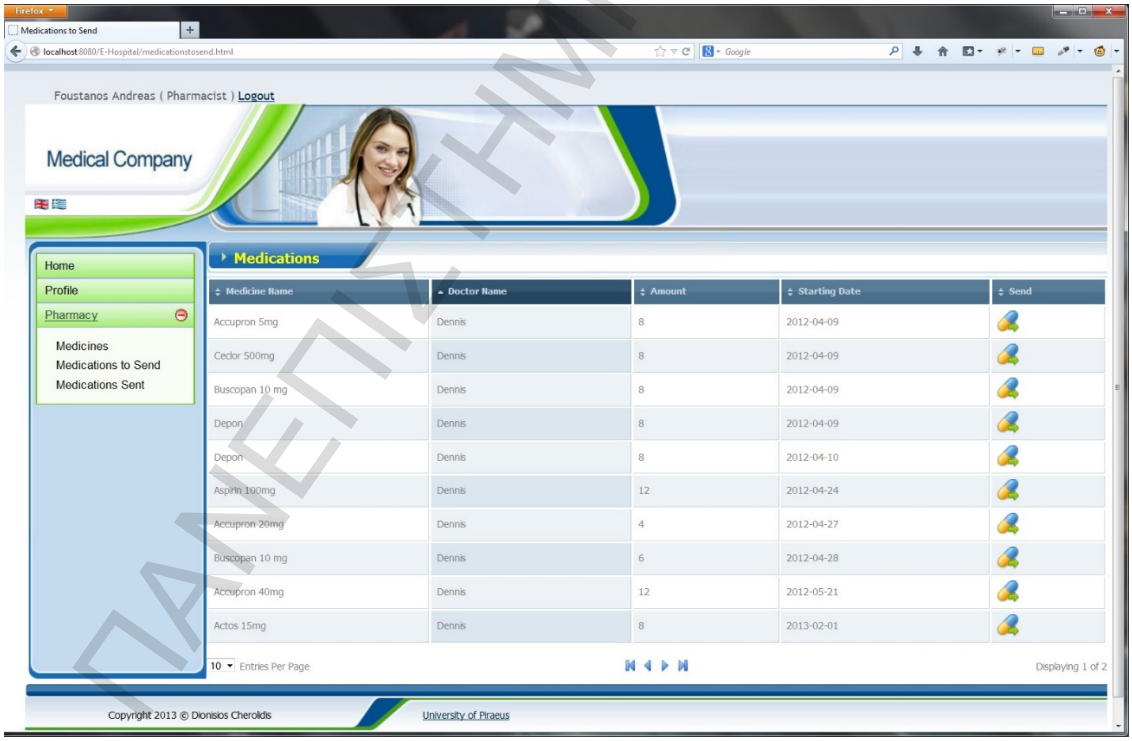
Εικόνα 35 Οδηγίες ιατρού προς τον υπεύθυνο νοσοκόμο η νοσηλευτή του ασθενή.

5.5 Φαρμακοποιός

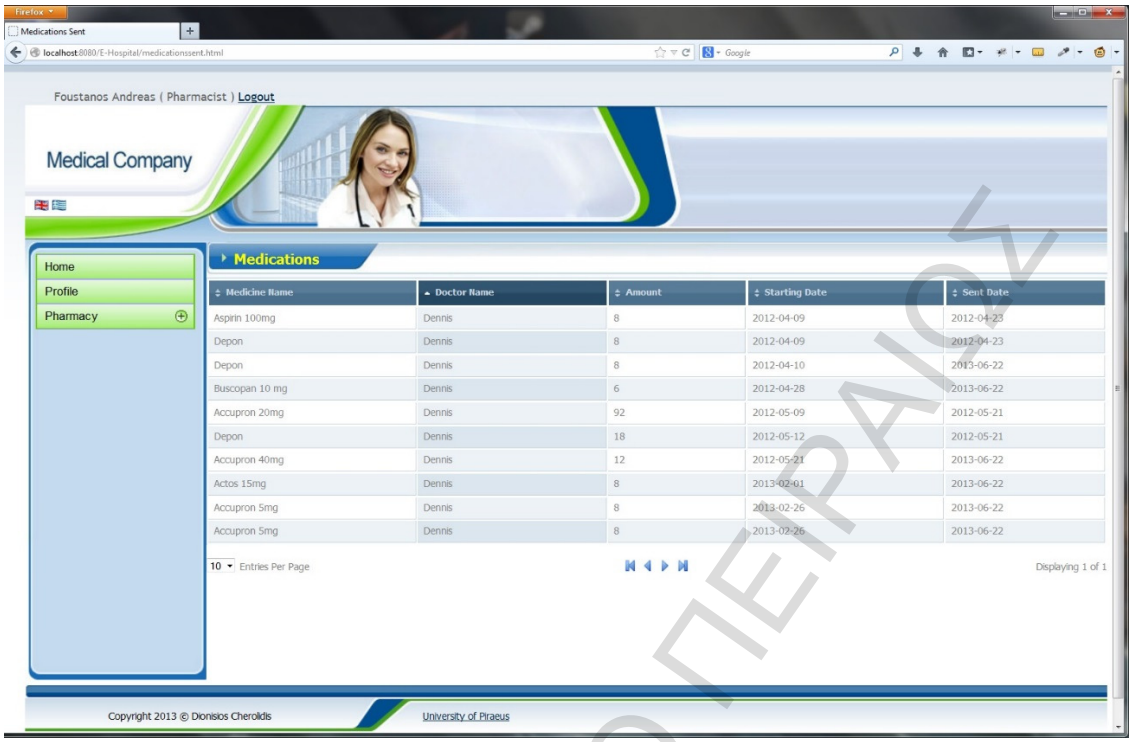
Ο φαρμακοποιός έχει την ευθύνη της διαχείρισης των φαρμάκων που διαθέτει το νοσοκομείο.Από το προσωπικό του μενού 'Pharmacy' έχει την δυνατότητα να μεταβαίνει στις διάφορες σελίδες που τον αφορούν. Με την επιλογή 'Medicines' μεταβαίνουμε στην σελίδα διαχείρισης των φαρμάκων (εικόνα 36), εδώ ο φαρμακοποιός μπορεί να προσθέσει και να διαγράψει φάρμακα από την λίσταή να επεξεργαστεί τα φάρμακα που ήδη υπάρχουν αλλάζοντας την διαθέσιμη ποσότητα τους.Στην συνέχεια έχουμε την επιλογή 'MedicationstoSend'όπου έχουμε όλες τις αιτήσεις για φάρμακα που έχουν γίνει από τους ιατρούς (εικόνα 37), εδώ ο φαρμακοποιός αφού προετοιμάσει την δοσολογία που χρειάζεται επιλέγει την φωτογραφία από την στήλη 'Send' ώστε να δηλώσει ότι η αποστολή του φαρμάκου έγινε και αυτομάτως η ποσότητα του συγκεκριμένου φαρμάκου που στάλθηκε αφαιρείτε από την συνολική διαθέσιμη ποσότητα ώστε η διαθεσιμότητα του κάθε φαρμάκου να είναι ενημερωμένη συνέχεια. Τελευταία επιλογή που έχει ο φαρμακοποιός είναι το 'MedicinesSent' (εικόνα 38) όπου έχουμε μία λίστα με τα φάρμακα που έχουν ήδη αποσταλεί στους ασθενείς ώστε να διατηρείται ένα αρχείο με όλη την κίνηση των φαρμάκων.



Εικόνα 36Διαχείριση Φαρμάκων.



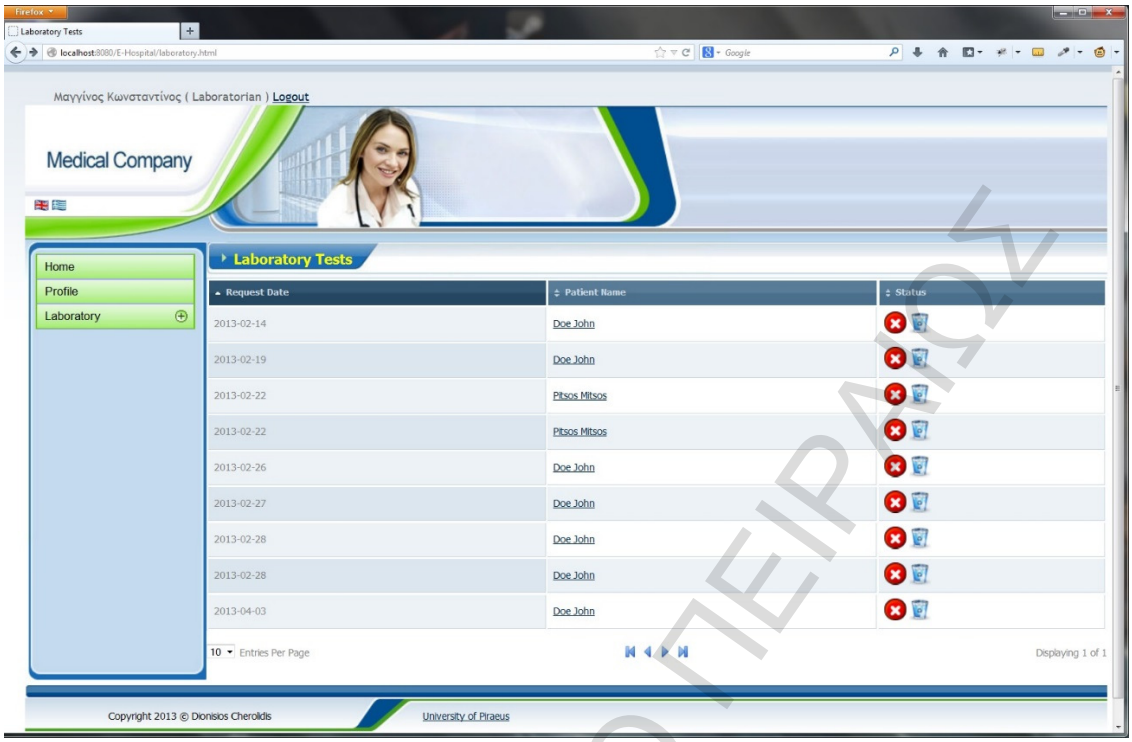
Εικόνα 37 Αποστολή Φαρμάκων.



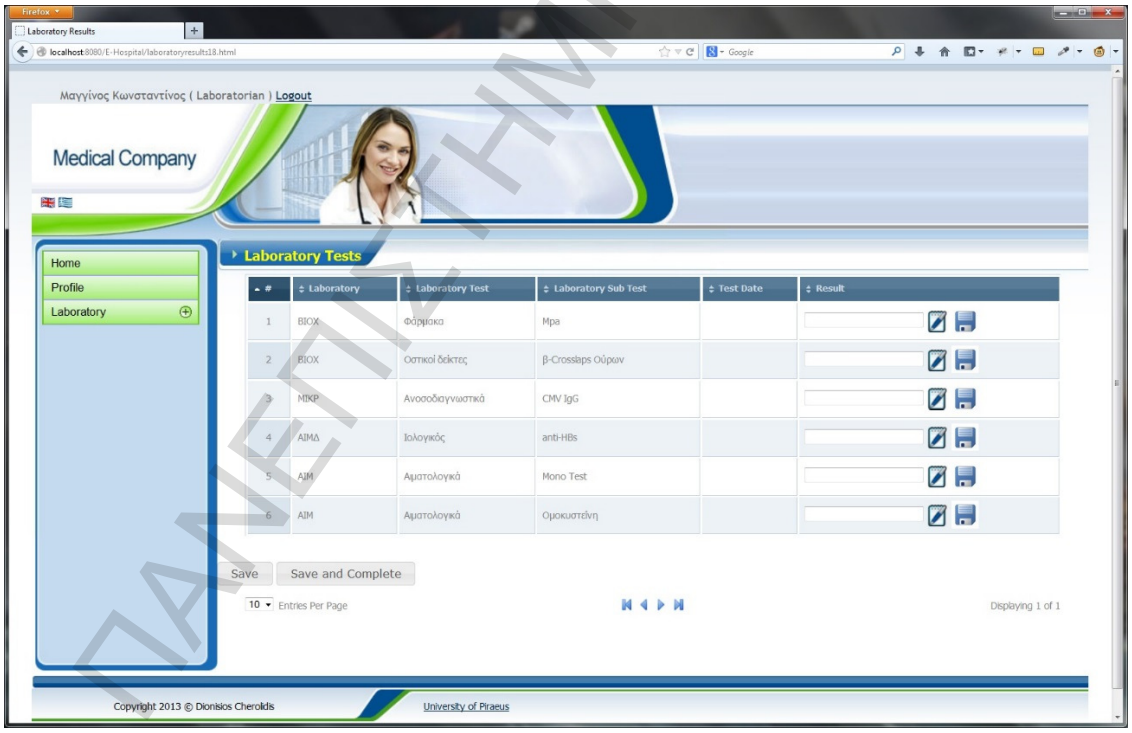
Εικόνα 38Απεσταλμένα Φάρμακα.

5.6 Εργαστηριακός Ιατρός

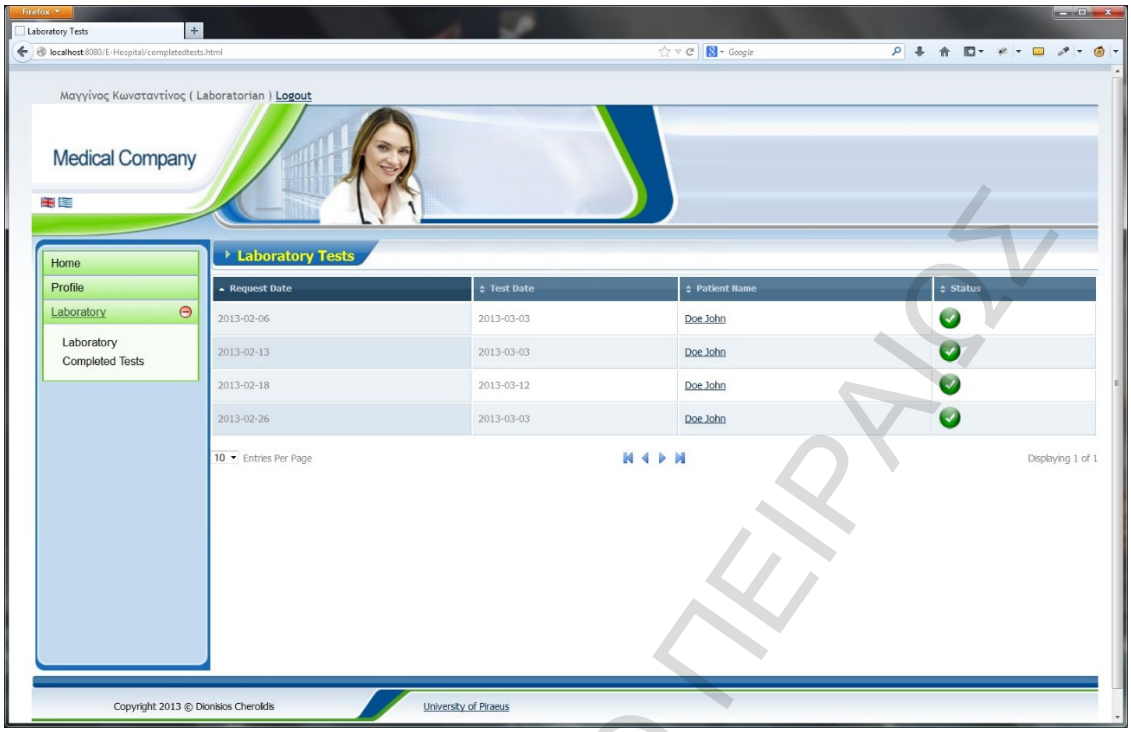
Ο εργαστηριακός ιατρός είναι υπεύθυνος για τις εργαστηριακές εξετάσεις που πρέπει να γίνουν σε κάποιον ασθενή και για την καταχώριση των αποτελεσμάτων στην βάση δεδομένων. Η καταχώριση θα γίνει μέσω του συστήματός από το μενού ‘Laboratory’ (εικόνα 39) όπου έχουμε μία λίστα με όλα τα εργαστηριακά τεστ που έχουν ζητηθεί να γίνουν από τους ιατρούς του νοσοκομείου. Επιλέγοντας κάποιον ασθενή θα δούμε ακριβώς όλες τις εξετάσεις που πρέπει να γίνουν για τον συγκεκριμένο ασθενή καθώς και την ημερομηνία που πρέπει να γίνουν (εικόνα 40). Αφού πραγματοποιηθούν αυτές οι εξετάσεις στην ίδια σελίδα καταγράφουμε τα αποτελέσματα και επιλέγουμε είτε ‘Save’για προσωρινή αποθήκευση χωρίς όμως να καταγραφούν τα αποτελέσματα στο ιστορικό του ασθενή. Εάν είναι έτοιμα όλα τα αποτελέσματα και γίνει η καταχώρισή τους τότε επιλέγουμε ‘SaveandComplete’ ώστε να καταχωριστούν τα τελικά αποτελέσματα στο ιστορικό του ασθενή όπου πλέον ο ιατρός μπορεί να τα δει. Η ημερομηνία που έγιναν τα τεστ καταγράφεται αυτόματα στο σύστημα ώστε ο ιατρός να γνωρίζει πότε ακριβώς πραγματοποιήθηκαν. Το μενού ‘CompletedTests’ παρουσιάζει μία λίστα με όλες τις προηγούμενεςεξετάσεις που έχουν πραγματοποιηθεί ώστε να υπάρχει μία πλήρης καταγραφή για το πότε ζητήθηκαν και πότε πραγματοποιήθηκαν οι εξετάσεις (εικόνα 41).



Εικόνα 39Προγραμματισμένα εργαστηριακές εξετάσεις.



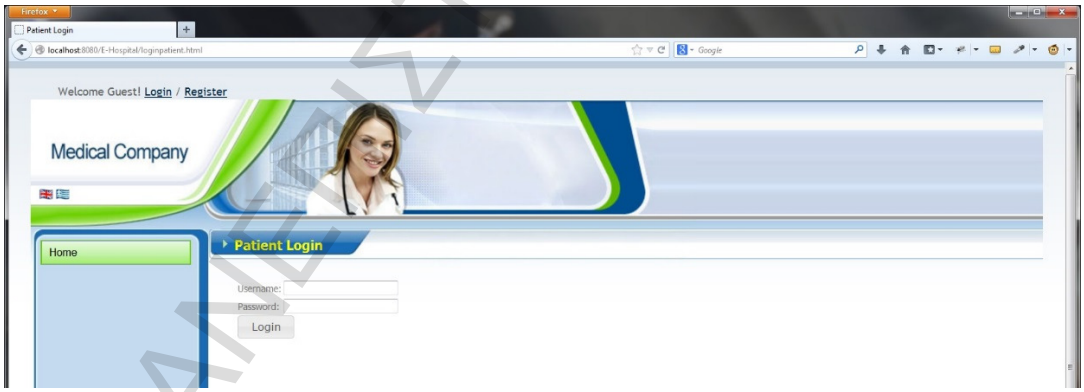
Εικόνα 40Καταχώριση αποτελεσμάτων εξετάσεων.



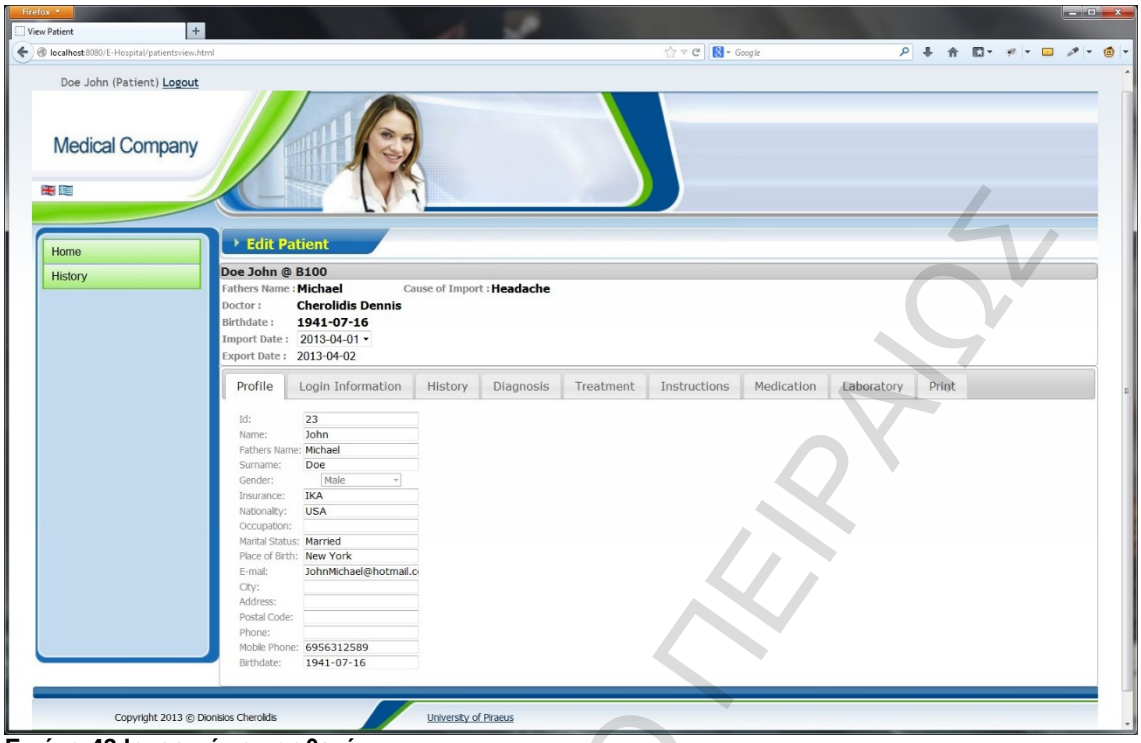
Εικόνα 41Εξετάσεις που έχουν πραγματοποιηθεί.

5.7 Ασθενής

Ο ασθενής για να συνδεθεί στο σύστημα πρέπει στην σελίδα εισόδου 'login'να επιλέξει την επιλογή 'For PatientsPleaseLog in fromhere!' (εικόνα 17). Έτσι θα μεταφερθεί σε μία νέα όμοια φόρμα (εικόνα 42) όπου συμπληρώνει το όνομα χρήστη και τον κωδικό πρόσβασης. Αφού γίνει η είσοδος στο σύστημα ο ασθενής έχει την δυνατότητα να δει το ιστορικό του αλλά φυσικά μόνο τα τμήματα και τις πληροφορίες που επιτρέπει το κάθε νοσοκομείο ξεχωριστά (εικόνα 43).



Εικόνα 42 Είσοδος στο σύστημα για ασθενείς.

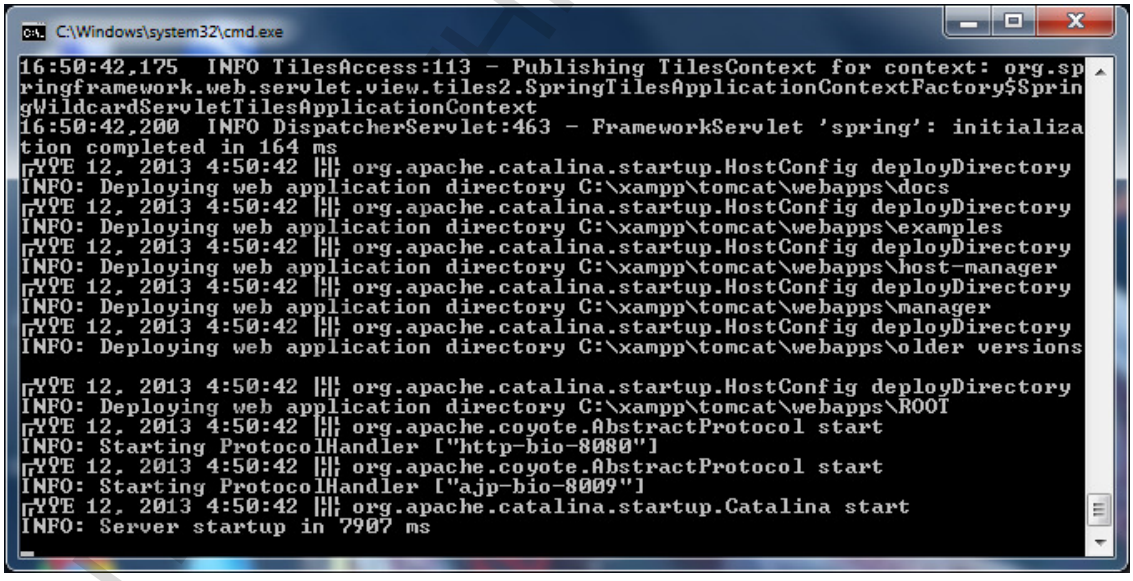


Εικόνα 43 Ιστορικό του ασθενή.

6. Εγχειρίδιο προγραμματιστή

Για να εγκατασταθεί το παρόν σύστημα σε ένα νοσοκομείο η κάποια κλινική πρέπει να διαθέτουμε έναν διακομιστήο οποίος θα φιλοξενήσει την εφαρμογή και τα δεδομένα και ένα εσωτερικό δίκτυο ενσύρματο ή ασύρματο για να έχουν πρόσβαση οι χρήστες στο Porta.Επίσης είναι απαραίτητη η ύπαρξη τουλάχιστον ενός ακόμα διακομιστή ο οποίος θα λειτουργεί σαν 'back-upserver'καθώς διαχειριζόμαστε ευαίσθητα δεδομένα μεγάλου όγκου και πρέπει να εξασφαλίσουμε την ασφάλειά τους και την διαθεσιμότητά τους. Έτσι ώστε εάν συμβεί κάτι στον πρώτο διακομιστή να είναι διαθέσιμος ο δεύτερος για να συνεχίσει την λειτουργία του συστήματος[46]. Γενικότερα, υπάρχουν πολλές μέθοδοι δημιουργίας backupσυστημάτων και ιδιαίτερα όταν έχουμε να κάνουμε με θέματα προσωπικών δεδομένων και νοσοκομειακών αρχείων πρέπει να είμαστε πολύ προσεκτικοί.

Για να εγκαταστήσουμε την εφαρμογή πρέπει ο διακομιστής μας καταρχήν να διαθέτει 'tomcatserver'**Error! Reference source not found.**έκδοση 6.0 ή νεότερη. Η εφαρμογή μας βρίσκεται στο αρχείο 'E-Hospital.war' το οποίο περιέχει τον πηγαίο κώδικα της εφαρμογής 'sourcecode'των αρχείωνJava, όλα τα .jsp, javascript, .cssκαι .xmlαρχεία που δημιουργήσαμε,αρχεία εικόνωνκαι όλα τα απαραίτητα αρχεία βιβλιοθήκης ('.jar') της Javaπου χρειαζόμαστε. Τα αρχεία βιβλιοθήκης είναι απαραίτητα ώστε να λειτουργήσει η εφαρμογή καθώς περιέχουν όλες της κλάσεις που απαιτούνται τα 'πλαίσια' που χρησιμοποιήσαμε αλλά και κάποιες βοηθητικές κλάσεις που χρησιμοποιούμε για λειτουργίες όπως κωδικοποίηση δεδομένων και κωδικών ασφαλείας, δημιουργία αρχείων καταγραφής 'logfiles', διαχείριση σελίδων λάθους και άλλα. Το αρχείο που περιέχει την εφαρμογή μας 'E-Hospital.war'πρέπει να τοποθετηθεί στον φάκελο 'webapps' του tomcatή σε οποιονδήποτε άλλο φάκελο έχει οριστεί από τον tomcat. Αφού τοποθετηθεί εκεί ανοίγοντας τον tomcatservern εφαρμογή μας θα γίνει deploy αυτόματα και θα δημιουργηθεί ένας φάκελος με το όνομα 'E-Hospital' που θα περιέχει την εφαρμογή μας. Πρέπει να ελέγξουμε τα logsκαι να βεβαιωθούμε ότι δεν υπάρχει κάποιο errorκαι ότι η εφαρμογή μας έγινε delpoy και ότι ο tomcatservermας ξεκίνησε χωρίς προβλήματα (Εικόνα 44).Πλέον θα μπορούμε να χρησιμοποιήσουμε την εφαρμογή μας πληκτρολογώντας την διεύθυνση 'http://localhost:8080/E-Hospital/' στον περιηγητή-browsermας.

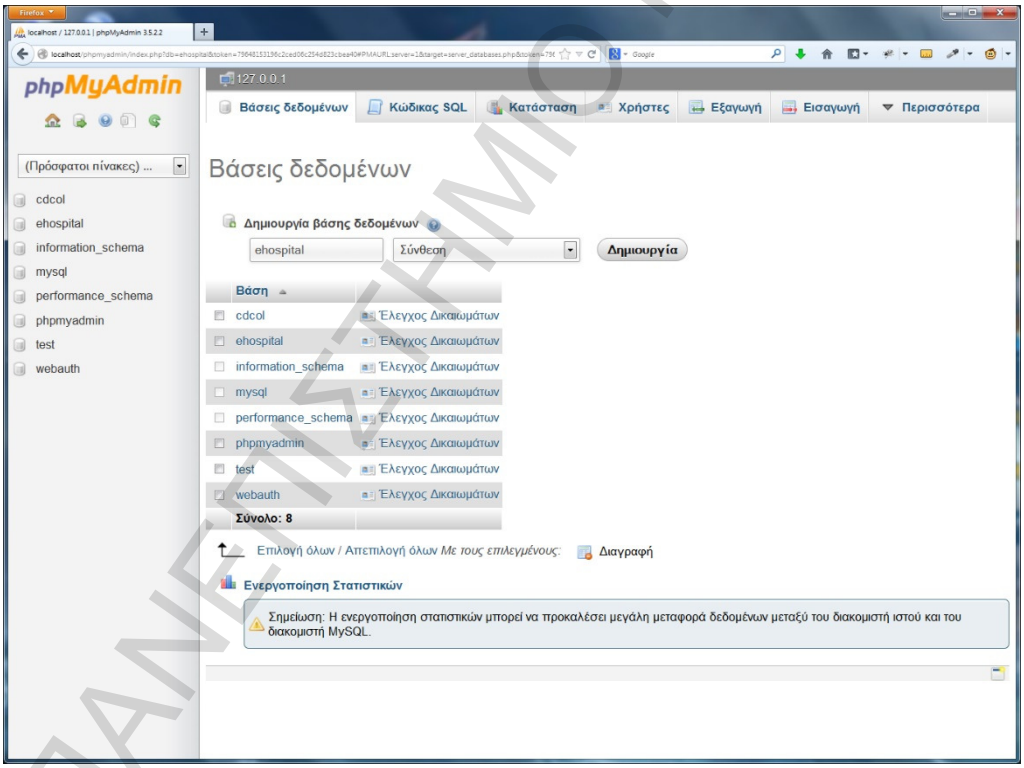


Εικόνα 44 Εκκίνηση του tomcatservermας.

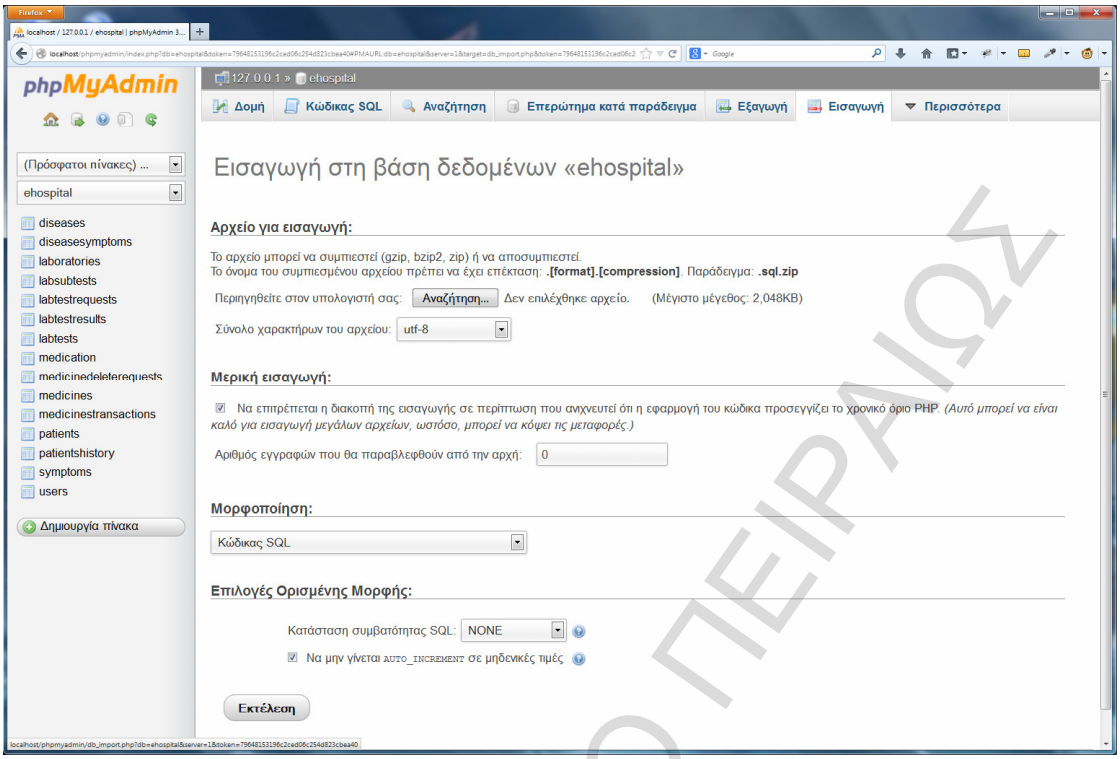
Με την παραπάνω διαδικασία εγκαθιστούμε την εφαρμογή.Όμως αυτή είναι άδεια χωρίς δεδομένα και χωρίς δυνατότητα αποθήκευσης των δεδομένων καθώς δεν έχουμε ορίσει κάποια βάση δεδομένων. Οι βάσεις δεδομένων που μπορεί να υποστηρίξει το σύστημά μας είναι πολλές και έχουν αναφερθεί στο κεφάλαιο που παρουσιάσαμε το FrameworkHibernate. Εμείς στην περίπτωσή μας

χρησιμοποιήσαμε την MySQLκαι έχουμε δημιουργήσει το αρχείο ‘ehospital.sql’το οποίο περιέχει τον .sqlκώδικα που απαιτείται για να δημιουργηθούν οι απαραίτητοι πίνακες και να εισαχθούν τα αρχικά δεδομένα που χρησιμοποιήθηκαν και για την δοκιμή της εφαρμογής. Για να δημιουργήσουμε την βάση εύκολα μπορούμε να χρησιμοποιήσουμε το phpMyAdmin**Error! Reference source not found.**τοοποίοαπαιτείύπαρξηapacheserver**Error! Reference source not found.**για να τρέξει.Πληκτρολογώντας την διεύθυνση ‘localhost/phpmyadmin’ στην γραμμή διευθύνσεων του περιηγητή μας θα μεταφερθούμε στην εφαρμογή phpMyAdminόπου μπορούμε να δημιουργήσουμε μία νέα βάση δεδομένων. Αυτό γίνεται επιλέγοντας ‘Βάσεις δεδομένων’και πληκτρολογώντας το όνομα της βάσης μας ‘ehospital’στο αντίστοιχο πεδίο όπως βλέπουμε στην εικόνα 45 και επιλέγοντας ‘Δημιουργία’ για να δημιουργηθεί η βάση μας. Στη συνέχεια επιλέγοντας την καινούρια βάση ‘ehospital’ από το αριστερό μενού εμφανίζεται η επιλογή ‘Εισαγωγή’ πάνω δεξιά στην οθόνη μας από όπου και μπορούμε να εισάγουμε το αρχείο μας το οποίο περιέχει τους πίνακες και τα δεδομένα. Συγκεκριμένα στο πεδίο ‘Αρχείο για εισαγωγή’ επιλέγουμε αναζήτηση και στην συνέχεια βρίσκουμε το αρχείο ‘ehospital.sql’ πατάμε άνοιγμα και στην συνέχεια εκτέλεση για να εκτελεστούν οι mysqlεντολές που διαθέτει (Εικόνα 46). Έτσι κατασκευάζεται η δομή της βάσης και εισάγονται κάποια αρχικά δεδομένα που χρησιμοποιήθηκαν κατά τον έλεγχο της εφαρμογής.

Τώρα πια με την βάση δεδομένων έτοιμη και τον tomcatserverανοιχτό μπορούμε πλέον να χρησιμοποιήσουμε κανονικά την εφαρμογή δημιουργώντας νέους λογαριασμούς χρήστη και κάνοντας είσοδο στο σύστημα ‘Login’. Ο αρχικός διαχειριστής του συστήματος ονομάζεται administrator και έχει όνομα χρήστη ‘administrator’ και κωδικό ‘administrator’.Με αυτόν τον χρήστη μπορούμε να μπούμε για να ενεργοποιήσουμε τους νέους λογαριασμούς που δημιουργούμε και φυσικά το όνομα χρήστη του και ο κωδικός του θα πρέπει να αλλάξουν μέσω του μενού ‘Profile’ για λόγους ασφαλείας.



Εικόνα 45Δημιουργία νέας βάσης ‘ehospital’με το phpMyAdmin



Εικόνα 46 Εισαγωγή πινάκων και δεδομένων

7. Όρυξη Δεδομένων

Εδώ και πολλά χρόνια χρησιμοποιούνται υπολογιστικά συστήματα για να αποθηκεύονται ηλεκτρονικά δεδομένα ασθενών που αφορούν ασθένειες, εργαστηριακές εξετάσεις και κάθε είδους πληροφορία που αφορά την υγεία. Έτσι τα μεγάλα νοσοκομείαλόγω των πολλών ασθενών που τα επισκέπτονται παράγουν μεγάλο όγκο και θα ήταν λάθος να μείνει ανεκμετάλλευτος μιας και θα μπορούσαν να προκύψουν χρήσιμα στατιστικά στοιχεία που θα βοηθούσαν στις μελλοντικές θεραπείες και διαγνώσεις αρκεί να καταφέρουμε να τα εντοπίσουμε και να τα εξαγάγουμε.Αυτή η διαδικασία ανάλυσης δεδομένων που έχουμε αποθηκεύσει σε ηλεκτρονικές βάσεις δεδομένων ώστε να εξορύξουμε χρήσιμα στοιχεία και συμπεράσματα ονομάζεται datamining και παρόλο που είναι μια πολύπλοκη και χρονοβόρα εργασία αξίζει και με το παραπάνω να γίνει[50]. Εκτός από ιατρικά δεδομένα ένα πληροφοριακό σύστημα νοσοκομείου μπορεί να διαθέτει και λογιστικές πληροφορίες διαχείρισης του νοσοκομείου και των υπαλλήλων και μπορούν και αυτές να φανούν χρήσιμες για την βελτίωση της διοίκησης και μείωση του κόστους λειτουργείας του νοσοκομείου[51].

Έτσι ξεκίνησαν μελέτες και έρευνες για το πώς θα μπορούσαμε να χρησιμοποιήσουμε όλη αυτή την γνώση προς όφελός μας. Μία τέτοια έρευνα δημοσιεύτηκε το 2003 όπου μελετώντας ασθενείς με νεφρική ανεπάρκεια εντόπισαν πάνω από πενήντα παραμέτρους που αφορούν κάποιον ασθενή και η γνώση των οποίων θα μπορούσε να διευκολύνει την θεραπεία του. Εξετάζοντας ασθενείς αυτής της πάθησης από το παρελθόν και παρακολουθώντας διάφορες παραμέτρους οι ερευνητές κατάφεραν να κατανοήσουν την επιρροή τους στον τρόπο και την μέθοδο θεραπείας που θα ήταν η καλύτερή για να εφαρμοστεί σε κάθε περίπτωση. Έτσι υποστηρίζουν στην δημοσίευσή τους ότι με την χρήση αλγορίθμων εξόρυξης δεδομένων κατάφεραν να εξάγουν την γνώση που χρειάζονταν για να καταλήξουν στο κατά πόσο επηρεάζει η κάθε παράμετρος την θεραπευτική αγωγή που πρέπει να ακολουθηθεί[52]. Παρόμοιες προσεγγίσεις μπορεί να γίνουν για κάθε είδους ασθένεια για την οποία θα θέλαμε να μάθουμε παραπάνω πληροφορίες ή δεν έχουμε τις απαιτούμενες γνώσεις για την πρόβλεψή και την καταπολέμηση της. Παρόμοια μελέτη άντλησης δεδομένων έγινε και για την ασθένεια της φυματίωσης που προκαλείτε από επικίνδυνα βακτήρια που προσβάλλουν τον άνθρωπο. Στην έρευνα αυτή μελετούν τα συμπτώματα που παρουσιάζονται πριν και μετά την ασθένεια και προσπαθούν να βρουν συσχετίσεις που θα βοηθήσουν στην πρόληψη και την καλύτερη και γρηγορότερη αντιμετώπιση της ασθένειας [53].

Μία άλλη πιο εξιδεικευμένη και δυσκολότερη περίπτωση είναι η εξαγωγή χρήσιμων πληροφοριών από ακτινογραφίες ή αξονικές και μαγνητικές τομογραφίες. Εδώ πλέον δεν έχουμε μετρήσιμα δεδομένα αλλά σχήματα τα οποία πρέπει να τα επεξεργαστούμε και να τα αναλύσουμε. Έτσι εάν διαθέτουμε μία μεγάλη βάση με εικόνες θα μπορούσαμε να κάνουμε συγκρίσεις χρησιμοποιώντας διάφορους μηχανισμούς επεξεργασίας και ανάλυσης εικόνας. Αυτό υλοποιήθηκε σε μαγνητικές εγκεφάλου που διέθετε το πανεπιστημιακό κέντρο υγείας ‘Leiden University’ για να εξαχθούν χρήσιμα συμπεράσματα για νευροεκφυλιστικές ασθένειες όπως ‘πάρκινσον’ και το ‘αλτσχαίμερ’ [54]. Για να συγκρίνουν τις εικόνες οι ερευνητές υλοποίησαν κατάλληλες μεθόδους με τις οποίες παρατηρούσαν και σημείωναν τυχόν διαφορές και ανωμαλίες στην δομή του εγκεφάλου. Στην συνέχεια συγκρίνοντας τα δεδομένα που εξήγαγαν για την κάθε ασθένεια και παρατηρώντας την πορεία της ασθένειας προσπάθησαν να βγάλουν χρήσιμα συμπεράσματα. Με την καλύτερη κατανόηση αυτών των ασθενειών έχουν πλέον ισχυρό πλεονέκτημα σε σχέση με το παρελθόν για την αντιμετώπιση μελλοντικών περιπτώσεων και την πρόβλεψη της πορείας της ασθένειας σε νέους ασθενείς.

Η εφαρμογή της μεθόδου εξόρυξης δεδομένων σε ηλεκτρονικά υπολογιστικά συστήματα νοσοκομείων πρέπει να γίνει με ιδιαίτερη προσοχή και από εξιδεικευμένο προσωπικό χρησιμοποιώντας εξιδεικευμένο λογισμικό που απαιτείται για αυτήν την διαδικασία. Επίσης πάντα όλο αυτό πρέπει να γίνεται με σεβασμό στα δεδομένα που επεξεργαζόμαστε καθώς πρόκειται για προσωπικά δεδομένα ασθενών. Η σωστή εφαρμογή εξόρυξης δεδομένων μπορεί να αποφέρει πληροφορίες που θα βοηθήσουν στην περίθαλψη των ασθενών και σε κάποιες περιπτώσεις μπορεί να σώσει και ανθρώπινες ζωές.

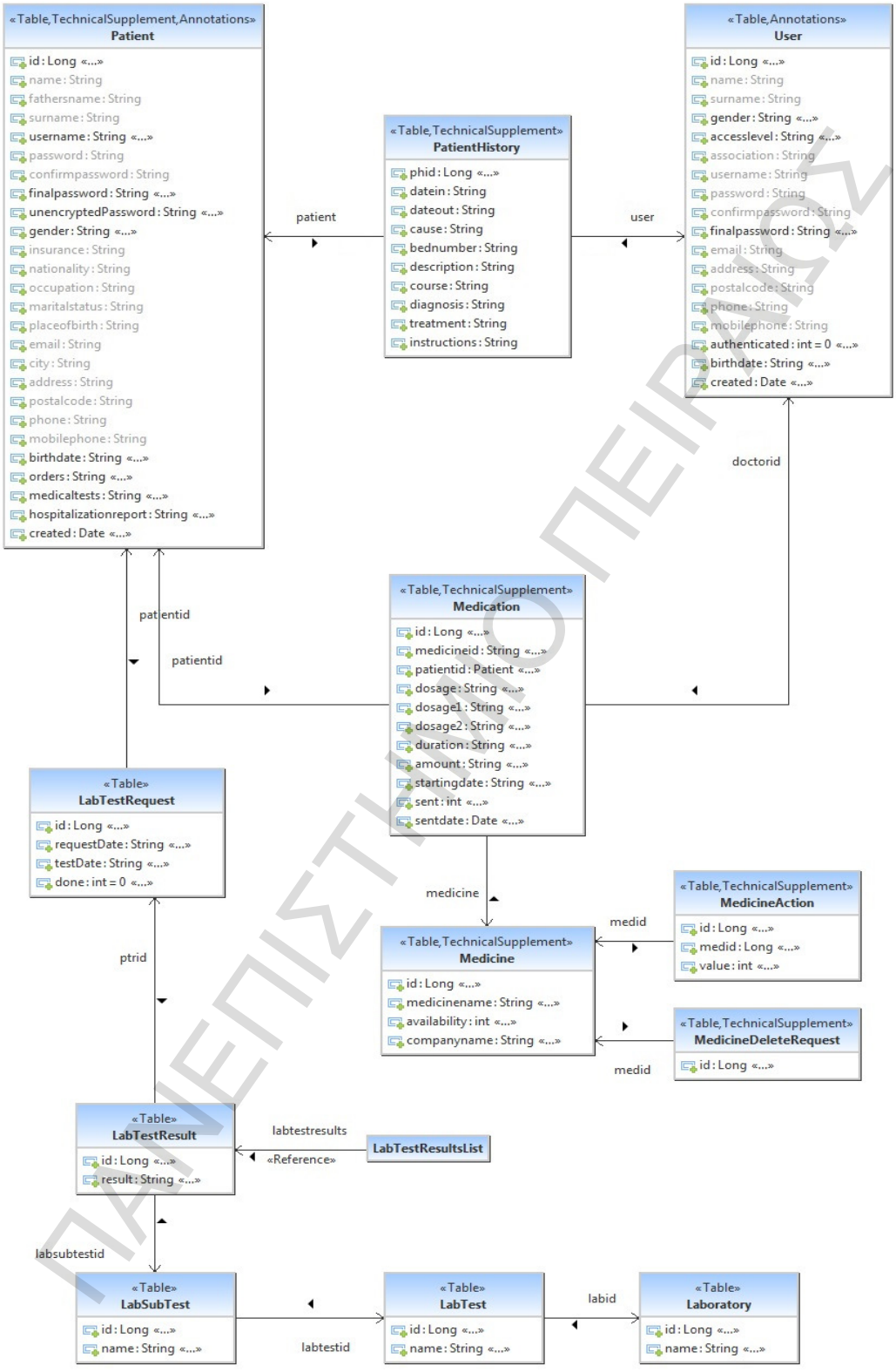
8. Συμπεράσματα

Με την υλοποίηση ενός πολύπλοκου πληροφοριακού συστήματος και την εφαρμογή του σε κέντρα υγείας βελτιώνουμε την ποιότητα παροχής υπηρεσιών προς τους ασθενείς, διευκολύνουμε την εσωτερική λειτουργία του κέντρου και εξοικονομούμε οικονομικούς πόρους μειώνοντας τις δαπάνες και αυξάνοντας την αποτελεσματικότητα. Αντιλαμβανόμαστε λοιπόν ότι είναι απαραίτητη η εφαρμογή παρόμοιων συστημάτων σε όλα τα κέντρα υγείας για να αναβαθμιστούν ποιοτικά και να γίνουν πιο παραγωγικά. Οι δυσκολίες που παρουσιάζονται είναι πολλές φορές λόγω οικονομικού κόστους και δυσκολιών προσαρμογής και εκπαίδευσης στην χρήση του λογισμικού από τους υπαλλήλους. Ένα από τα συμπεράσματα που μπορούν να προκύψουν από την παρούσα διατριβή είναι ότι σε καμία περίπτωση δεν μπορούν να δικαιολογηθούν τα τεράστια ποσά που επενδυθήκαν για την ανάπτυξη παρόμοιων συστημάτων στην Ελλάδα. Μιας και αφού είναι δυνατή η δημιουργία ενός τέτοιου συστήματος στα πλαίσια μιας διατριβής, ακόμα και αν σε καμία περίπτωση το σύστημα αυτό δεν μπορεί να θεωρηθεί πλήρες και ικανοποιητικό, μία πιο μεγάλη ομάδα προγραμματιστών σε συνεργασία με ιατρούς του δημοσίου θα μπορούσαν να δημιουργήσουν ένα αξιόλογο πληροφοριακό σύστημα για νοσοκομεία. Επίσης το σύστημα αυτό θα μπορούσε να εφαρμοστεί σε όλα τα δημόσια νοσοκομεία της Ελλάδος με τις όποιες παραλλαγές και μετατροπές θα απαιτούνταν για το κάθε νοσοκομείο και όχι να ανατίθεται το κάθε νοσοκομείο σε διαφορετική εταιρία πληροφορικής πολλαπλασιάζοντας το κόστος κάλυψης όλων των κέντρων.

Σημαντική παρατήρηση που μπορούμε να εξαγάγαμε από την εμπειρία που αποκτήσαμε κατά την υλοποίηση της εφαρμογής είναι ότι για να πετύχει δεν επαρκούν μόνο η εμπειρία και οι γνώσεις του προγραμματιστή. Απαιτείται στενή συνεργασία με προσωπικό που δουλεύει σε νοσοκομεία και κέντρα υγείας όπως με ιατρούς, νοσηλευτές, φαρμακοποιούς κτλ. Η συνεργασία και η επικοινωνία μεταξύ των δύο πλευρών πρέπει να ξεκινήσει από το στάδιο του σχεδιασμού της εφαρμογής ώστε ο υπάλληλος να μεταφέρει της εμπειρίες και τις γνώσεις του στον προγραμματιστή και να εντοπίσουν τα σημεία στα οποία μπορεί η τεχνολογία να βελτιώσει την λειτουργία του νοσοκομείου. Η συνεργασία αυτή πρέπει να συνεχιστεί κατά την διάρκεια της υλοποίησης για τυχόν διευκρινίσεις, λεπτομέρειες και διορθώσεις που θα χρειαστούν και φυσικά θα πρέπει να παραμείνει και μετά την τελική έκδοση της εφαρμογής καθώς όπως και σε κάθε εφαρμογή έτσι και σε αυτήν είναι σχεδόν σίγουρο ότι θα χρειαστούν κάποιες επιδιορθώσεις ή προσθήκες για την βελτίωσή της.

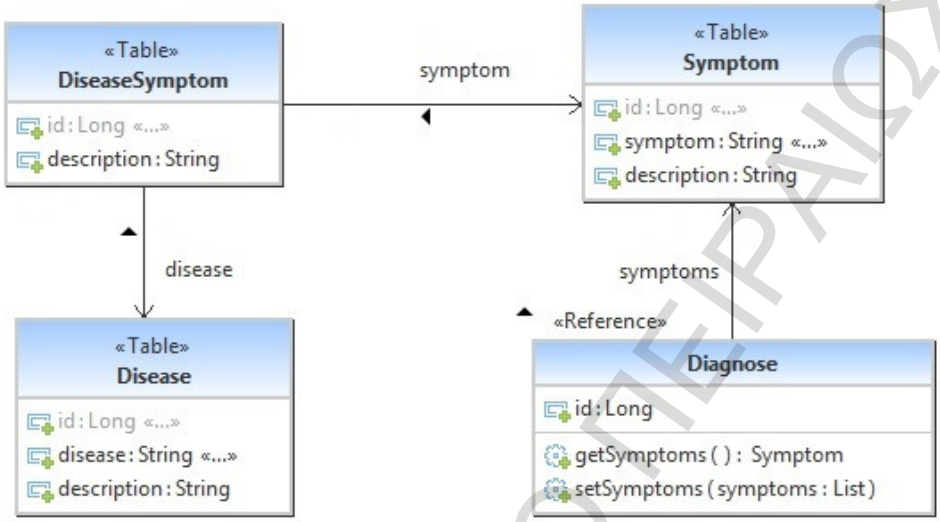
Παράρτημα - Δομή κώδικα

Στο κεφάλαιο αυτό θα μελετήσουμε και θα παρουσιάσουμε την δομή των Javaκλάσεων που δημιουργήσαμε για την εφαρμογή μας. Στην πρώτη εικόνα που βλέπουμε παρακάτω έχουμε όλες τις κλάσεις με τις οποίες δημιουργούμε τα αντικείμενα τα οποία θα χρησιμοποιήσουμε για να αποθηκεύουμε τα δεδομένα τα οποία επεξεργαζόμαστε. Παρατηρώντας την εικόνα 47 με τις κλάσεις θα δούμε ότι είναι όμοια με το UMLδιάγραμμα από το κεφάλαιο 3.6 της παρούσας διατριβής. Αυτό είναι λογικό καθώς ό,τι δεδομένα χρησιμοποιούμε τα αποθηκεύουμε στην βάση δεδομένων μας και για να γίνει αυτή η αποθήκευση και η περαιτέρω επεξεργασία στο μέλλον γίνεται χρήση αντικειμένων της Javaτα οποία έχουν ιδιότητες, οι οποίες αντιπροσωπεύουν τις στήλες των πινάκων της βάσης μας. Έτσι για να γίνει οποιαδήποτε αποθήκευση στην βάση τα δεδομένα πρώτα περνάνε στο αντικείμενο μας και μετά με την βοήθεια του framework 'hibernate' αποθηκεύονται στην αντίστοιχη βάση μας, ενώ όταν θέλουμε να ανακτήσουμε δεδομένα από την βάση μας αυτά αποθηκεύονται στα αντικείμενα που έχουμε και μετά από εκεί τα χρησιμοποιούμε είτε για επεξεργασία είτε για εμφάνιση στον χρήστη. Στην εφαρμογή μας αυτές οι κλάσεις ονομάζονται 'views' διότι μέσω αυτών 'βλέπουμε' τα δεδομένα μας.

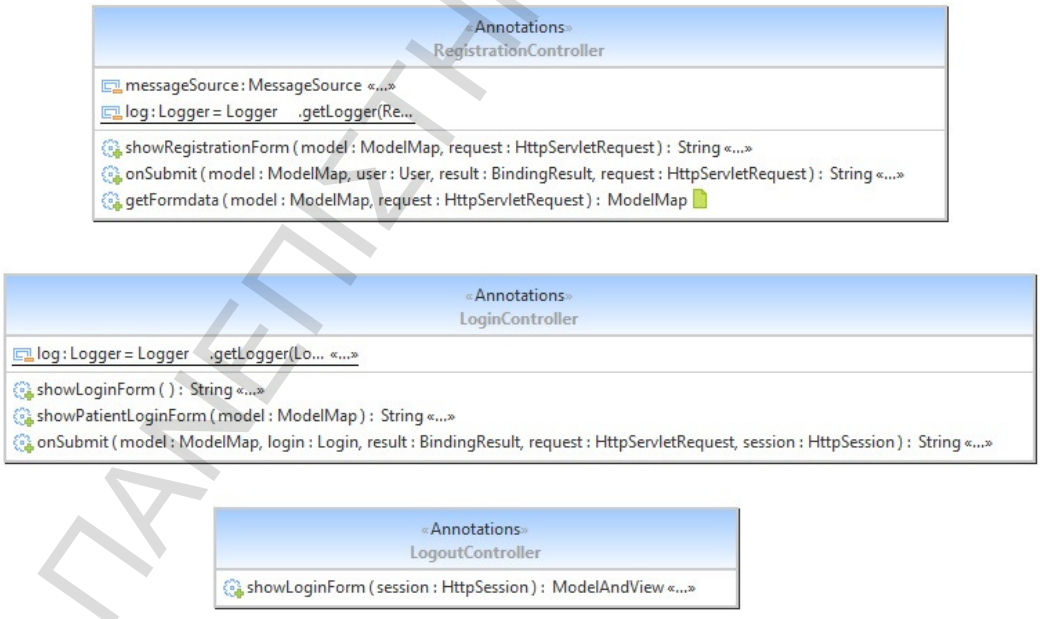


Εικόνα 47 Κλάσεις (Views).

Στην συνέχεια έχουμε κι άλλες παρόμοιες κλάσεις (Views)που δεν έχουν καμία συσχέτιση με τις προηγούμενες διότι αφορούν την εφαρμογή της διάγνωσης ασθενειών που έχουν οι ιατροί και η οποία λειτουργεί ανεξάρτητα. Η λειτουργία τους είναι όπως και πρίν για την αποθήκευση και μεταφορά δεδομένων. Σε αυτές τις κλάσεις δεν γίνεται επεξεργασία δεδομένων κάτι που συμβαίνει παρακάτω σε άλλες κλάσεις.



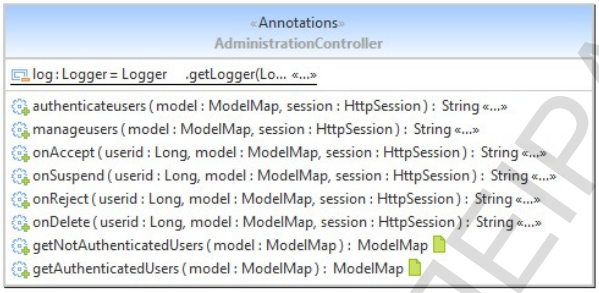
Εικόνα 48Κλάσεις (Views) για την διάγνωση ασθενειών.

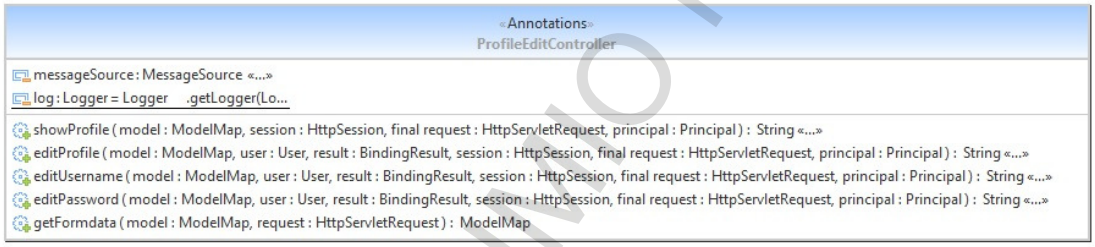


Εικόνα 49ControllersΕγγραφής Εισόδου και Εξόδου στο σύστημα.

Οι Controllers όπως αναφέραμε σε προηγούμενο κεφάλαιο είναι Java κλάσεις οι οποίες δέχονται τα ‘requests’ από τον χρήστη και τα ‘εξυπηρετούν’. Είναι οι κύριες κλάσεις της εφαρμογής μας και στην ουσία κάνουν όλες τις διεργασίες που πρέπει να γίνουν για να διαβάσουμε, να τροποποιήσουμε και να

προσθέσουμε στοιχεία στην βάση δεδομένων μας. Η χρήση τους δεν περιορίζεται μόνο στα δεδομένα της βάσης. Ένας 'controller'μπορεί να δεχτεί ένα 'request'και να κατευθύνει τον χρήστη στην αντίστοιχη σελίδα που απαιτείται χωρίς απαραιτήτως να επικοινωνήσει με την βάση. Αυτό μπορεί να γίνει όταν ζητηθεί μια στατική σελίδα που δεν απαιτεί δεδομένα ή εάν εντοπιστεί κάποιο λάθος και πρέπει να ενημερωθεί ο χρήστης με το κατάλληλο μήνυμα. Στην εικόνα 49 έχουμε τρείς controllersοι οποίοι δέχονται τα αιτήματα εγγραφής στο σύστημα (registration), εισόδου στο σύστημα (login)και εξόδου από το σύστημα (logout). Αφού λάβει την αίτηση ελέγχονται τα δεδομένα που έχει και εκτελούνται οι ενέργειες που απαιτούνται για να γίνει η λειτουργία που ζητήθηκε.



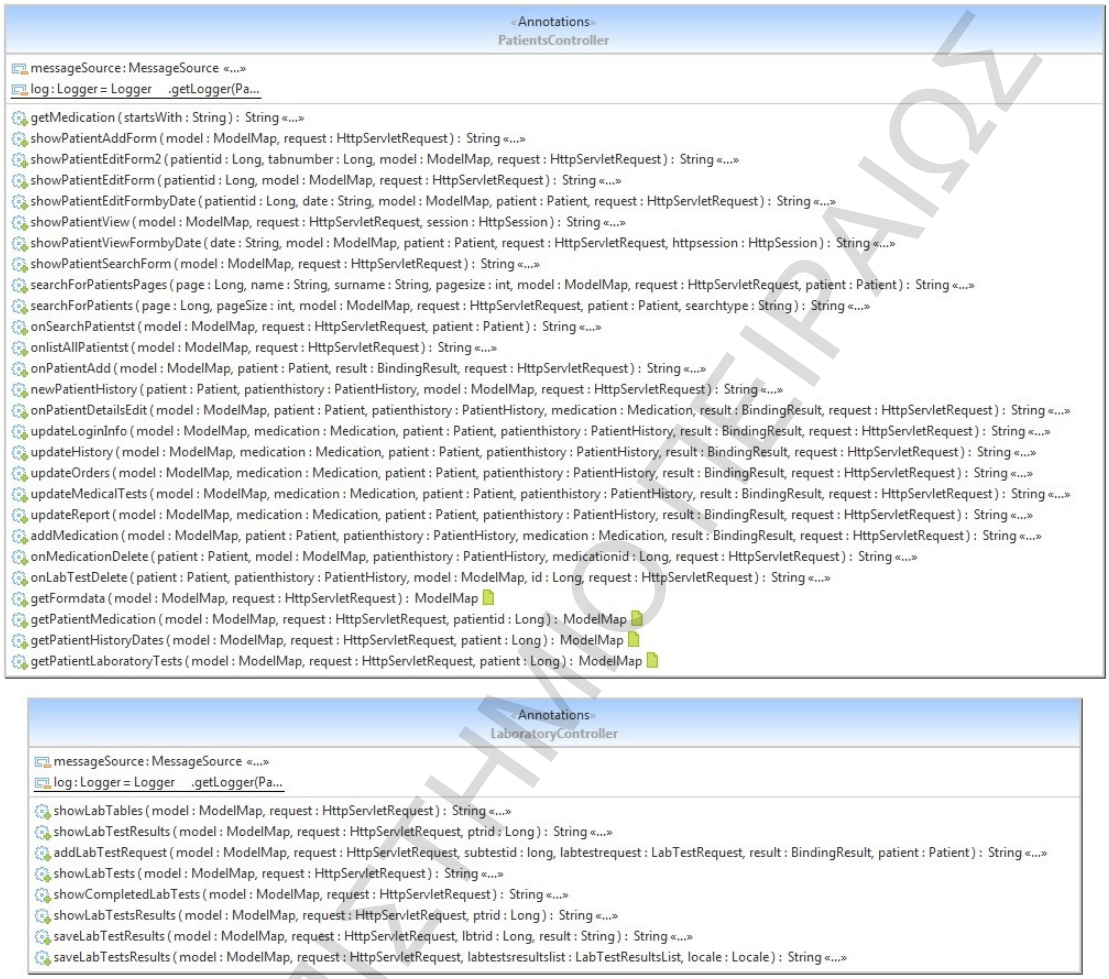


ΕικόναError! Reference source not found.**ControllersΔιαχειριστή και επεξεργασίας προφίλ.**

Στην εικόνα 50 βλέπουμε τους controllers'Administration' και 'ProfileEdit' καθώς και τις μεθόδους που χρησιμοποιούν μαζί με τις παραμέτρους τους. Οι μέθοδοι που διαθέτουν καλούνται για να εκτελέσουν κάποια εργασία. Για παράδειγμα ο controller 'Administration' αφορά ενέργειες που μπορεί να κάνει ο διαχειριστής του συστήματος όπως αποδοχή χρηστών στο σύστημα (onAccept), απόρριψη εγγραφής χρήστη (onReject), διαγραφή χρήστη (onDelete), μπλοκάρισμα χρήστη (onSuspend). Επίσης μπορεί να ζητήσει την λίστα με τους εγγεγραμμένους χρήστες (getAuthenticatedUsers)ή την λίστα με τους χρήστες που έχουν κάνει αίτηση για εγγραφή (getNotAuthenticatedUsers). Για όλες αυτές τις ενέργειες υπάρχει η αντίστοιχη μέθοδος που εκτελείτε. Το ίδιο ισχύει και για τον controller'ProfileEdit'ο οποίος περιέχει μεθόδους αλλαγής ονόματος χρήστη, κωδικού και στοιχείων των χριστών της εφαρμογής μας.

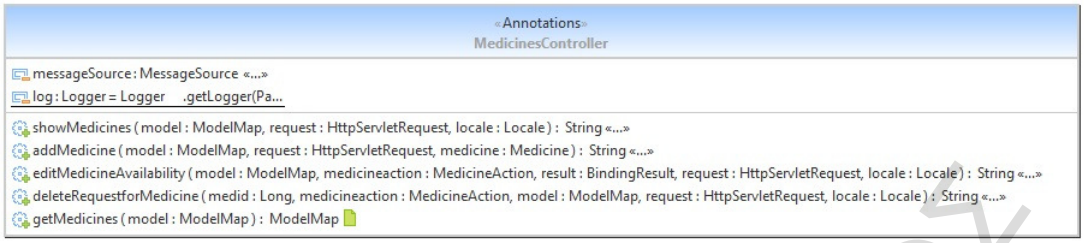
Στην εικόνα 51 έχουμε έναν από τους κυριότερους controller της εφαρμογής μας τον 'PatientsController'ο οποίος είναι υπεύθυνος για την διαχείριση των περισσοτέρων πληροφοριών που αφορούν τους ασθενείς. Χρησιμοποιείται για την αναζήτηση ασθενών, προβολή στοιχείων ασθενή, προβολή ιστορικού ασθενή, προσθήκη νέων ασθενών, τροποποίηση των στοιχείων και του ιστορικού και πολλά άλλα. Οι λειτουργίες αυτές είναι διαθέσιμες για τους ιατρούς της εφαρμογής αλλά όχι μόνο. Κάποιες από τις μεθόδους μπορεί να χρησιμοποιηθούν και από άλλες ομάδες χρηστών όπως οι νοσηλευτές οι οποίοι μπορούν και αυτοί να αναζητήσουν ασθενείς και να δουν πληροφορίες που τους αφορούν για την περίθαλψη του ασθενή. Για τη διασφάλιση της ακεραιότητας των δεδομένων ώστε να μην μπορούν να παραβιαστούν γίνεται χρήση του πλαισίου springsecurity,με το οποίο μπορούμε εσωτερικά στον κώδικα να ορίσουμε, για κάθε μέθοδο αλλά και μέρος του κώδικα κάποιας μεθόδου ξεχωριστά το ποια ομάδα χρηστών μπορεί να έχει πρόσβαση και ποια όχι. Για παράδειγμα στις μεθόδους που αφορούν τροποποίηση στοιχείων του ιστορικού του ασθενή έχει πρόσβαση μόνο η ομάδα χριστών των ιατρών και καμία άλλη.

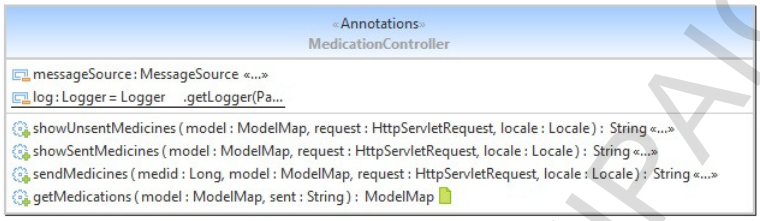
Ένας τέτοιος διαχωρισμός συμβαίνει και στο 'LaboratoryController' το οποίο διαχειρίζεται τις εργαστηριακές εξετάσεις των ασθενών. Εδώ έχει πρόσβαση σε κάποιες μεθόδους ο ιατρός ώστε να μπορεί να ζητήσει τις εξετάσεις που θέλει και σε κάποιες άλλες μεθόδους έχει πρόσβαση και ο εργαστηριακός ιατρός ο οποίος θα δει τις εξετάσεις που ζητήθηκαν και όταν αυτές ολοκληρωθούν θα περάσει τα αποτελέσματα στην βάση για να τις δει ο ιατρός.



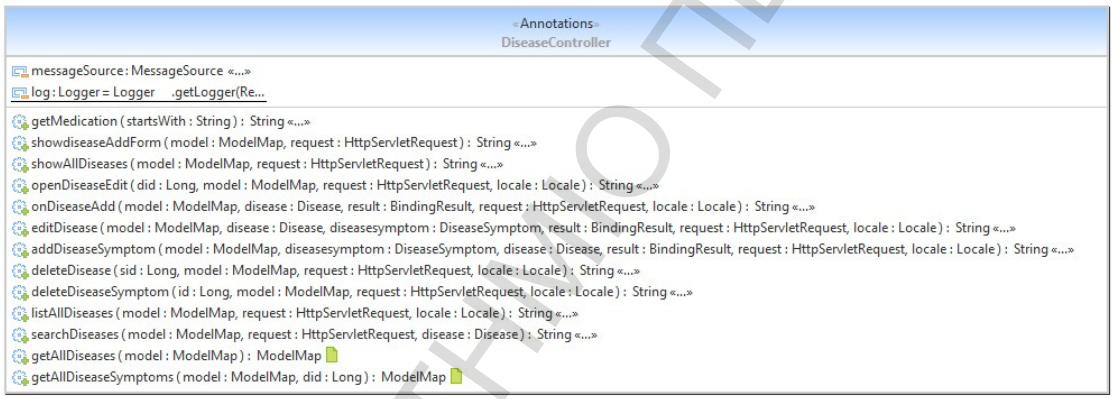
ΕικόναError! Reference source not found.**ControllersΑσθενή και Εργαστηρίου.**

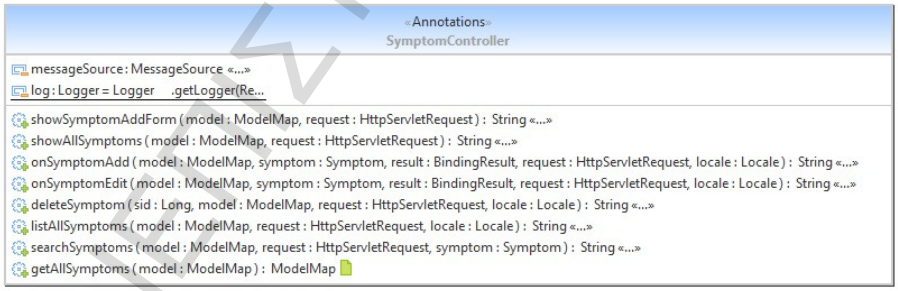
Στηνπερίπτωση της εικόνας 52 οιMedicines και Medicationχειρίζονται τα διαθέσιμα φάρμακα (προσθήκη νέων φαρμάκων, τη διαγραφή παλαιών, τηντροποποίηση διαθέσιμων φαρμάκων) αλλά και την αποστολή των φαρμάκων που ζητήθηκαν από τους ιατρούς στους ασθενείς. Οπότε χρησιμοποιείται από τον φαρμακοποιό και έτσι αυτός είναι και ο μόνος που έχει πρόσβαση στις μεθόδους αυτών των δύο controller.

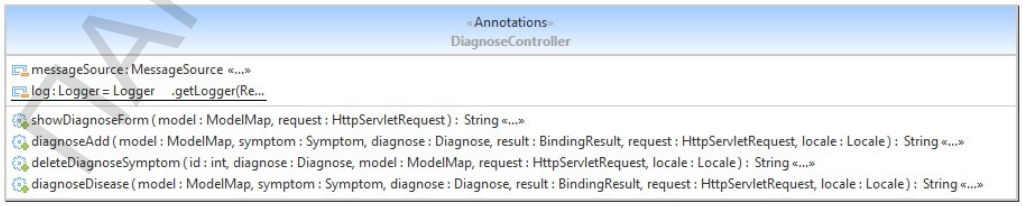




Εικόνα 50ControllersΦαρμάκων και Φαρμακευτικής αγωγής.

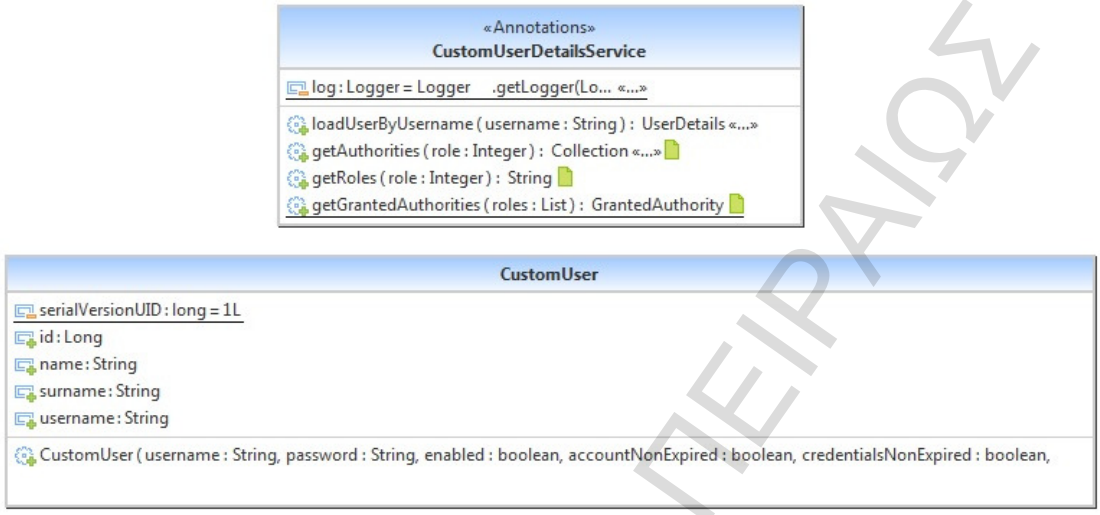






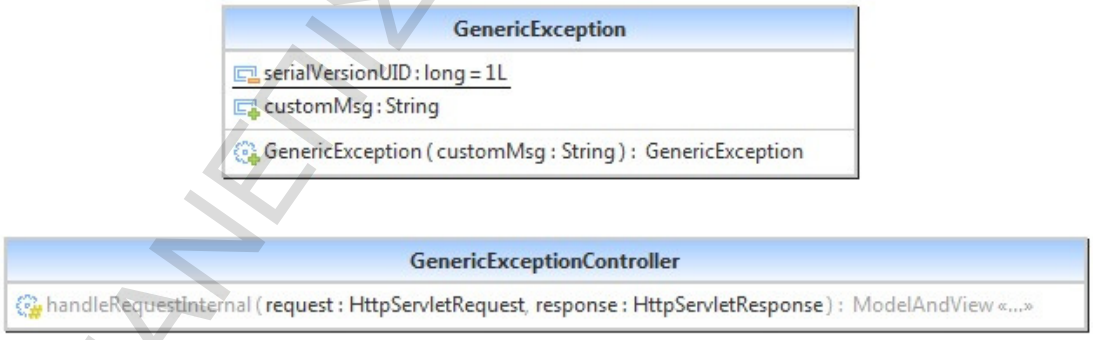
ΕικόναError! Reference source not found.**ControllersΑσθένειας,Συμπτώματος και Διάγνωσης.**

Στην εικόνα 53 έχουμε τους τρείς ακόμαcontrollersτης εφαρμογής.Οι οποίοι χρησιμοποιούνται για την λειτουργία της διάγνωσης που διαθέτουν οι ιατροί. Ο Diseaseαφορά την προσθαφαίρεση και τροποποίηση των ασθενειών, ο Symptomτην προσθαφαίρεση και τροποποίηση των συμπτωμάτων και ο Diagnoseτην λειτουργία της διάγνωσης των ασθενειών με βάση τα συμπτώματα.



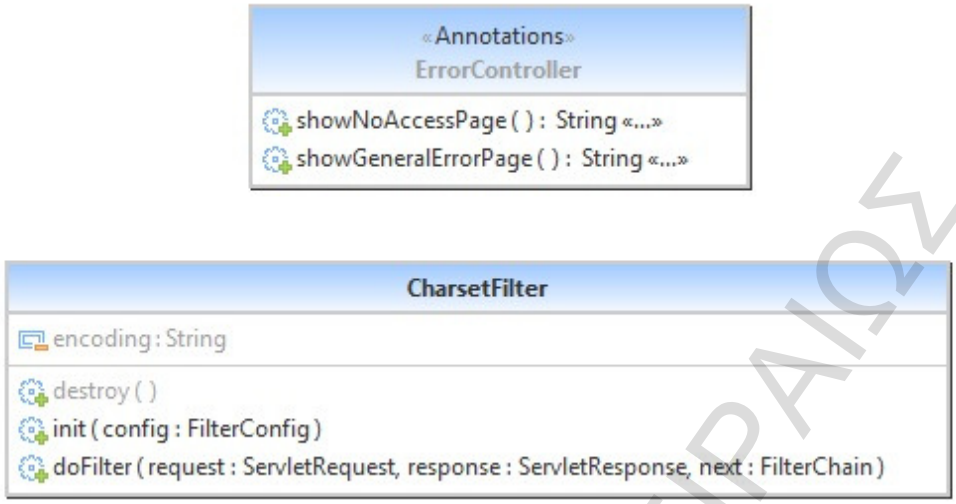
Εικόνα 51Javaκλάσεις αυθεντικοποίησης χρήστη.

Οι δύο κλάσεις της εικόνας 54 δημιουργήθηκαν για να προσαρμόσουμε την προκαθορισμένη λειτουργία της αυθεντικοποίησης του χρήστη η οποία έχει οριστεί από το ‘springFramework’. Η κλάση ‘CustomUser’κάνει ‘extend’δηλαδή επεκτείνει μία άλλη υπάρχουσα κλάση την ‘User’ έτσι κληρονομεί όλα τα στοιχεία, τις μεταβλητές και τις συναρτήσεις από την υπάρχουσα κλάση και μας δίνεται η δυνατότητα να προσθέσουμε και δικά μας στοιχεία. Έτσι προσθέτουμε επιπλέον μεταβλητές για το όνομα χρήστη, το id, το όνομα και το επίθετο τα οποία χρειαζόμαστε στην εφαρμογή μας αλλά δεν υπήρχαν. Η κλάση ‘CustomUserDetailsService’ αντικαθιστά την αντίστοιχη κλάση που υπάρχει από το ‘springFramework’ την οποία τροποποιούμε ελάχιστα για να προσθέσουμε τα στοιχεία που αναφέραμε προηγουμένως και για να ορίσουμε τους ρόλους που θα έχουμε (διαχειριστής, ιατρός, νοσοκόμος, φαρμακοποιός, εργαστηριακός ιατρός),έτσι ώστε να μπορεί το σύστημα να διαχωρίζει τους χρήστες.



Εικόνα 52Javaκλάσεις για Exceptions

Για τα exceptionsτο πλαίσιόμας έχει κάποια προκαθορισμένη συμπεριφορά με προκαθορισμένα μηνύματα. Εμείς μπορούμε να παρέμβουμε σε αυτά και να αλλάξουμε είτε τα μηνύματα λάθους είτε την διαδικασία που ακολουθείται και τις σελίδες που εμφανίζονται. Εμείς με τις κλάσεις της εικόνας 55 ορίζουμε ότι για κάθε γενικό λάθος που συμβαίνει να εμφανίζεται κάποια συγκεκριμένη σελίδα με ένα μήνυμα που θέσαμε εμείς. Με αυτή την αλλαγή όταν συμβεί κάποιο λάθος στην εκτέλεση του κώδικά μας αυτό να μην εμφανίσει στον χρήστη κάποιο πολύπλοκο μήνυμα λάθους αλλά το δικό μας μήνυμα. Φυσικά το κανονικό μήνυμα λάθους καταγράφεται στα αρχεία καταγραφής που διαθέτουμε.



Εικόνα 53 Βοηθητικές κλάσεις.

Ένα συχνό πρόβλημα που παρουσιάζεται σε ιστοσελίδες και δικτυακές εφαρμογές είναι η χρήση διαφορετικής κωδικοποίησης χαρακτήρων καθώς ο χρήστης μπορεί να επιλέξει από τον σελιδοδείκτη του οποιαδήποτε κωδικοποίηση. Εάν επιλέξει λανθασμένη κωδικοποίηση και προσπαθήσει να αποθηκεύσει κάποιο κείμενο στα Ελληνικά, αυτό υπάρχει πιθανότητα να αποθηκευτεί λανθασμένα στην βάση μας. Για αυτόν τον λόγω δημιουργήσαμε την κλάση 'CharsetFilet'η οποία ελέγχει όλες τις επικοινωνίες με τον χρήστη και ορίζει σαν κωδικοποίηση χαρακτήρων την 'UTF-8'ώστε να μην έχουμε πρόβλημα με τους Ελληνικούς χαρακτήρες. Η κλάση 'errorController' είναι βοηθητική κλάση και έχουμε ορίσει δύο προκαθορισμένες σελίδες που θα εμφανίζεται η μία όταν υπάρξει πρόβλημα προσβασιμότητας και η άλλη για οποιοδήποτε άλλο πρόβλημα βρει το 'springSecurityFramework'.

Εκτός από αυτές έχουμε κι άλλες βοηθητικές κλάσεις για διάφορες λειτουργίες όπως είναι η 'HibernateUtil' η οποία έχει οριστεί από το 'HibernateFramework' που χρησιμοποιούμε. Το περιεχόμενό της δίνεται από το πλαίσιοκαι εμείς απλά δημιουργούμε την κλάση και την καλούμε κάθε φορά που θέλουμε να επικοινωνήσουμε με την βάση χρησιμοποιώντας το 'HibernateFramework'.Επίσης έχουμε την κλάση 'UserValidator'την οποία δημιουργήσαμε για να επαληθεύουμε το όνομα χρήστη και τον κωδικό πρόσβασης κατά την διάρκεια της εγγραφής του χρήστη. Ελέγχουμε καταρχήν εάν οι δύο κωδικοί που εισήγαγε ο χρήστης είναι όμοιοι και στην συνέχεια βλέπουμε εάν το όνομα χρήστη που επέλεξε υπάρχει ήδη στην βάση μας, εάν περάσουμε τους ελέγχους συνεχίζεται η διαδικασία εγγραφής ενώ εάν υπάρξει κάποιο πρόβλημα εμφανίζουμε ένα μήνυμα λάθους για να ενημερώσουμε τον χρήστη. Τέλος υπάρχουν και επιπλέον controllersγια τις στατικές σελίδες που έχουμε καθώς όπως είπαμε κάθε 'request'του χρήστη περνάει από 'controller'ο οποίος καθορίζει την σελίδα που θα εμφανίσουμε στον χρήστη. Τέτοιες σελίδες είναι η κεντρική σελίδα, η φόρμα εγγραφής, η φόρμα εισαγωγής στο σύστημα και διάφορες άλλες.

Βιβλιογραφία

[1] Βικιπαίδεια, «Πληροφοριακά συστήματα,» [Ηλεκτρονικό]. Available: http://el.wikipedia.org/wiki/Πληροφοριακά_συστήματα. [Πρόσβαση Μάρτιος 2013].

[2] Βικιπαίδεια, «Υπολογιστής Ταμπλέτα,» [Ηλεκτρονικό]. Available: http://el.wikipedia.org/wiki/Υπολογιστής_ταμπλέτα. [Πρόσβαση 2013 Μάρτιος].

[3] H. Gottinger, «Computers in hospital care: a qualitative assessment. Human Systems Management,» 1984:324–345.

[4] «ΑΕΜΥ,» Ανώνυμη Εταιρεία Μονάδων Υγείας (ΑΕΜΥ Α.Ε.), [Ηλεκτρονικό]. Available: http://www.aemy.gr/web/guest/72. [Πρόσβαση Ιούνιος 2012].

[5] Νοσηλευτική, «ρόλος των Πληροφοριακών Συστημάτων Υγείας στην οργάνωση και διεκπεραίωση της νοσηλευτικής πρακτικής,» 2008. [Ηλεκτρονικό]. Available: http://www.hjn.gr/actions/get_pdf.php?id=68. [Πρόσβαση Μάρτιος 2013].

[6] «Ηλεκτρονική Συνταγογράφηση,» Γενική Γραμματεία Κοινωνικών Ασφαλίσεων, [Ηλεκτρονικό]. Available: http://www.e-syntagografisi.gr/.

[7] Α. Βαγγελάτος και Ι. Σαριβουγιούκας, «Πληροφοριακό Σύστημα Νοσοκομείου: Απαραίτητη υποδομή στο σύγχρονο Νοσοκομείο,» Υπουργείο Υγείας και Πρόνοιας.

[8] Γ. Πάγκαλος, «Κέρκυρα 2008, Πληροφοριακά Συστήματα Υγείας,» [Ηλεκτρονικό]. Available: http://www.ionio.gr/depts/cs/index.php?option=com_docman&task=doc_download&gid=91&Itemid [Πρόσβαση Απρίλιος 2013].

[9] «Ευρωπαικά Χρηματοδοτούμενα Αναπτυξιακά Προγράμματα Στην Ελλάδα,» Υπουργείο Οικονομίας και Οικονομικών, 2005.

[10] Intracom, «Ολοκληρωμένο Πληροφοριακό Σύστημα του Νοσοκομείου Παίδων,» Intracom, [Ηλεκτρονικό]. Available: http://www.intracom.gr/. [Πρόσβαση Απρίλιος 2013].

[11] Wikipedia, «Multitier architecture,» [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/Multitier_architecture. [Πρόσβαση Απρίλιος 2013].

[12] Datamed, «Systems Integration & Consulting Services,» [Ηλεκτρονικό]. Available: http://www.datamed.gr/.

[13] Apollo, «Information Technologies,» [Ηλεκτρονικό]. Available: http://www.apollo.gr/dev/Orion/Concerto.asp. [ΠρόσβασηΑπρίλιος 2013].

[14] «Techology Evaluation Centers,» BT Business Solutions, [Ηλεκτρονικό]. Available: http://www.technologyevaluation.com/research/company/BT-Business-Solutions.html. [Πρόσβαση Απρίλιος 2013].

[15] «CTG,» Healthcare Providers, [Ηλεκτρονικό]. Available: http://www.ctg.com/industries/healthcare-providers/. [Πρόσβαση Απρίλιος 2013].

[16] «Yasasii,» Yasasii, [Ηλεκτρονικό]. Available: http://www.yasasiisys.com/. [Πρόσβαση Απρίλιος 2013].

[17] «Web Based Hospital Information Management System,» Quanta , [Ηλεκτρονικό]. Available: http://www.quanta-his.com/QuantaOverview.html. [Πρόσβαση Απρίλιος 2013].

[18] «LHP,» HP, [Ηλεκτρονικό]. Available: http://h71028.www7.hp.com/enterprise/cache/425098-0-0-

14-121.html. [Πρόσβαση Απρίλιος 2013].

[19] «Βικιπαίδεια,» wikipedia, [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/Agile_software_development. [Πρόσβαση Απρίλιος 2013].

[20] Volgainfotech, «Methodologies,» [Ηλεκτρονικό]. Available: http://volgainfotech.com/Methodologies.aspx. [Πρόσβαση Απρίλιος 2013].

[21] C. Dewan, «BASICS OF WEB APP ATTACK,» [Ηλεκτρονικό]. [Πρόσβαση Μάρτιος 2013].

[22] Oracle, «Java Documentation,» [Ηλεκτρονικό]. Available: http://docs.oracle.com/javaee/6/api/javax/servlet/Servlet.html. [Πρόσβαση Μάρτιος 2013].

[23] Oracle, «Java EE,» [Ηλεκτρονικό]. Available: http://www.oracle.com/technetwork/java/javaee/overview/index.html. [Πρόσβαση Μάρτιος 2013].

[24] Wikipedia, «Relational database management system,» [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/Relational_database_management_system. [Πρόσβαση Μάρτιος 2013].

[25] C. Springsource, Springsource, [Ηλεκτρονικό]. Available: http://www.springsource.org/. [Πρόσβαση March 2013].

[26] Apache, «Licenses,» [Ηλεκτρονικό]. Available: http://www.apache.org/licenses/. [Πρόσβαση Μάρτιος 2013].

[27] Struts, Apache, [Ηλεκτρονικό]. Available: http://struts.apache.org/. [Πρόσβαση Μάιος 2013].

[28] Springsource, «Spring Security,» [Ηλεκτρονικό]. Available: http://static.springsource.org/spring-security/site/index.html.

[29] GNU. [Ηλεκτρονικό]. Available: http://www.gnu.org/copyleft/lesser.html.

[30] «Red Hat,» [Ηλεκτρονικό]. Available: http://www.redhat.com/. [Πρόσβαση Μαίος 2013].

[31] Apache, «Apache Tiles,» [Ηλεκτρονικό]. Available: http://tiles.apache.org/framework/. [Πρόσβαση Μαίος 2013].

[32] Tutorialzine, «framewarp,» [Ηλεκτρονικό]. Available: http://tutorialzine.com/2012/07/framewarp-jquery-plugin/.

[33] TinyMCE. [Ηλεκτρονικό]. Available: http://www.tinymce.com/. [Πρόσβαση Ιούλιος 2013].

[34] JQuery. [Ηλεκτρονικό]. Available: http://jquery.com/. [Πρόσβαση Μάιος 2013].

[35] Wikipedia, «BarCamp,» [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/BarCamp. [Πρόσβαση Μάιος 2013].

[36] Wikipedia, «Ajax (programming),» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Ajax_%28programming%29.

[37] Y. Haiyan, L. Jingsong, C. Huan, Z. Xiaoguang, T. Yu και Y. Yibing, «Performance Evaluation of Post-Relational Database in Hospital Information Systems,» Wuhan, 2010.

[38] «Strategy Management Group,» [Ηλεκτρονικό]. Available: http://www.balancedscorecard.org/. [Πρόσβαση August 2013].

[39] S. Mitropoulos, «A Simulation-based Approach for IT and Business Strategy Alignment and Evaluation».

[40] D. R. S. Kaplan και D. P. Norton, The Balanced Scorecard: Translating Strategy into Action, Boston : Harvard Business School Press, 1996.

[41] M. S., O. D. και H. D., «The balanced scorecard: a necessary good or an unnecessary evil?,» Τόμ. %1 από %217 No5 pp 481-491.

[42] C. E. Schneier, D. G. Shaw, R. W. Beatty και L. S. Baird, Performance Measurement, Management, and Appraisal Sourcebook, Amherst Massachusetts: HRD Press, 1995.

[43] J. Steele, «Transforming the Balanced Scorecard into Your Strategy Execution System,» τόμ. 53, αρ. 1: 22-24.

[44] W. Zelman, G. Pink και C. Matthias, «Use of the Balanced Scorecard in Health Care,» τόμ. 29, αρ. 4, 2003.

[45] P. Littlejohns, J. C. Wyatt και L. Garvican, «Evaluating computerised health information systems: hard lessons still to be learnt,» *BMJ ,* τόμ. 860, 2003.

[46] B. J. Liu, S. S. Chao, J. Documet, J. Lee, M. Lee, I. Topic και L. Williams, «Implementation of an ASP model offsite backup archive for clinical images utilizing Internet 2,» 2005.

[47] Apache, «Tomcat,» [Ηλεκτρονικό]. Available: http://tomcat.apache.org/. [Πρόσβαση July 2013].

[48] «phpMyAdmin,» [Ηλεκτρονικό]. Available: http://www.phpmyadmin.net/home_page/index.php. [Πρόσβαση July 2013].

[49] Apache, «Apache Server,» [Ηλεκτρονικό]. Available: http://www.apache.org/. [Πρόσβαση July 2013].

[50] S. Tsumoto, «Medical knowledge discovery in hospital information systems,» 2001.

[51] S. Tsumoto και Y. Tsumoto, «Mining hospital management data,» 2006.

[52] S. Shah, A. Kusiak και B. Dixon, «Data mining in predicting survival of kidney dialysis patients,» 2003.

[53] A. T., N. S. και M. K. N. B., «Association-rule-based tuberculosis disease diagnosis,» 2010.

[54] A. Ekin, R. Jasinschi, J. v. d. Grond και M. v. Buchem, «Mining databases to better understand neurodegenerative diseases,» 2007. [Ηλεκτρονικό]. Available: http://spie.org/documents/Newsroom/Imported/0721/0721-2007-04-13.pdf. [Πρόσβαση Ιούλιος 2013].