ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ
ΠΛΗΡΟΦΟΡΙΚΗ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΪΑΤΡΙΚΗ

ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΚΑΤΑΝΕΜΗΜΕΝΗΣ ΑΠΟΘΗΚΕΥΣΗΣ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑΣ ΙΑΤΡΙΚΩΝ ΕΙΚΟΝΩΝ ΣΕ ΠΕΡΙΒΑΛΛΟΝ ΠΛΕΓΜΑΤΟΣ (IMAGE-GRID)

Υπαπαντή Θερμού

Υπεύθυνος
Μαγκλογιάννης Ηλίας
Επίκουρος Καθηγητής

ΛΑΜΙΑ 2010

ΕΥΧΑΡΙΣΤΙΕΣ

Πριν προχωρήσω στην παρουσίαση της πτυχιακής αυτής εργασίας οφείλω να ευχαριστήσω θερμά ορισμένους ανθρώπους που χωρίς την βοήθεια και την συμπαράσταση τους δεν θα μπορούσα να ολοκληρώσω.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα της εργασίας μου, Επίκουρο Καθηγητή Ηλία Μαγκλογιάννη που μου έδωσε τη δυνατότητα να ασχοληθώ με το τόσο ενδιαφέρον ερευνητικό θέμα των εφαρμογών σε τεχνολογίες πλέγματος καθώς επίσης και την ιδιαίτερα πολύτιμη καθοδήγηση του. Επίσης, θα ήταν παράληψη να μην ευχαριστήσω τους υποψήφιους διδάκτορες Ιωάννη Κανάρη και Χαράλαμπο Δούκα που με το ενδιαφέρον τους, τις εύστοχες παρατηρήσεις και συμβουλές τους με βοήθησαν αρκετές φορές κατά τη διάρκεια εκπόνησης της εργασίας.

Τέλος, οφείλω ένα πολύ μεγάλο ευχαριστώ στην καλύτερη οικογένεια που θα μπορούσε ένα παιδί να έχει, τους γονείς μου Ειρήνη και Επαμεινώνδα, τον θείο μου Λευτέρη και τα αδέρφια μου Σπύρο και Δημήτρη. Με την αγάπη και τη συμπαράσταση τους με βοήθησαν να φτάσω ως εδώ και με παράδειγμα τη ζωή τους να καταλάβω την αξίας της γνώσης ως μέσο για να βελτιώσουμε τον εαυτό μας.

ΠΕΡΙΕΧΟΜΕΝΑ

Ευρετήριο Εικόνων

Ευρετήριο Πινάκων

ΠΕΡΙΛΗΨΗ

Οι επιστήμες ζωής βρίσκονται στο επίκεντρο των ερευνητών με μια συνεχώς αυξανόμενη ένταση τις τελευταίες δεκαετίες στον τομέα της ανάλυσης των ιατρικών και/ή βιοϊατρικών δεδομένων. Τα δεδομένα που παράγονται παγκοσμίως καταλαμβάνουν έναν ιδιαίτερα μεγάλο αποθηκευτικό χώρο. Ο λόγος αυτό οδήγησε στην αναζήτηση έξυπνων τρόπων για την αποδοτική διαχείρισή τους. Υπέρ αυτής της κατεύθυνσης κινούνται οι νέες τεχνολογίες. Το πλέγμα προσφέρει μια εναλλακτική προσέγγιση στην επεξεργασία (παράλληλος υπολογισμός) και διαχείριση των δεδομένων αφού μπορεί να τα κατανείμει με ασφάλεια σε υπολογιστικές και αποθηκευτικές μονάδες με κατανεμημένο τρόπο. Τα ιατρικά δεδομένα και συγκεκριμένα οι ιατρικές εικόνες με τις οποίες ασχολείται αυτή η εργασία βρίσκουν πρόσφορο έδαφος στις τεχνολογίες πλέγματος καθώς τα SEs (Storage Elements) μπορούν να λειτουργήσουν ως μια τεράστια βάση δεδομένων. Στην εργασία αυτή παρουσιάζεται η διεπαφή IMAGEGRID που λειτουργεί ως συνδετικός κρίκος μεταξύ του χρήστη και της υποδομής του πλέγματος (Grid Infrastructure). Οι υπηρεσίες ιστού (web services) αποτελούν το μαγικό κλειδί που αναλαμβάνει να κάνει αυτή την επικοινωνία απολύτως εφικτή. Η εφαρμογή λειτουργεί συνολικά ως ένα περιορισμένο πληροφοριακό σύστημα που κρατά στοιχεία για τους ασθενείς και τους ιατρούς που εμπλέκονται σε κάθε εξέταση άρα και σε κάθε ιατρικό αρχείο. Κεντρικό σημείο αυτής της υλοποίησης είναι ότι ο τελικός χρήστης μπορεί χρησιμοποιήσει το πλέγμα με έναν τρόπο ιδιαίτερα φιλικό αφού αποκτά πρόσβαση δημιουργώντας έναν απλό λογαριασμό στη βάση δεδομένων. Έτσι, χρησιμοποιεί το πλέγμα όπως θα χρησιμοποιούσε οποιοδήποτε άλλο σύστημα αλλά επωφελείται του πλεονεκτήματος ότι έχει μεγάλο αποθηκευτικό χώρο για τα αρχεία του βλέποντας τα με έναν ενοποιημένο τρόπο. Για την υλοποίηση του συστήματος θεωρείται σημαντικός ο συνδυασμός διαφορετικών στοιχείων όπως το λογισμικό gLite που παρέχεται από το Grid, οι τεχνολογίες διαδικτύου και στοιχεία βάσεων δεδομένων. Ο τρόπος με τον οποίο εμπλέκονται όλα τα παραπάνω στοιχεία δεν αφορά μόνο τα τεχνικά τους χαρακτηριστικά αλλά και τη φιλοσοφία που βρίσκεται πίσω τους και το σύνολο κανόνων που τα διέπει.

ABSTRACT

Life sciences are in the center of scientific research with an ongoing growing intensity especially in the field of biological and/or biomedical data analysis. The produced data worldwide need a great amount of storage resources. In this way, clever solutions are searched in order to efficiently manage all these data. New technologies are in favor of this approach. The grid offers an alternative approach at the process (parallel programming) and data management because it is able to "scatter" all these data in a great number of computing and storage elements in a safety way. Medical data and, especially, medical images which play a key role in this project, can use the grid technologies, as SEs are able to behavior like a huge database. In this project, IMAGEGRID interface is presented and it links the final user with the grid. Web services are a magic key that undertake to do this connection absolutely implemented. Totally, the implementation simulates a limited information system that stores records for all involved patients and doctors for each examination and each file as follows. A central point of this implementation is that final user can use the grid with a friendly way by just creating an account at the database. Therefore, users use the grid as using every other system, but benefits from the advantage of having «infinite» storage space viewing his files with a consolidated way. To implement such a system the combination of various significant factors including gLite software provided by the Grid, Internet technologies and database data is of major importance. The way that engages all the above components is not only their technical characteristics but also the philosophy that supports them.

1. ΕΙΣΑΓΩΓΗ

Η τεχνολογία των υπολογιστών παρουσίασε αλματώδη πρόοδο τις δύο τελευταίες δεκαετίες. Εξίσου σημαντική ήταν και η αύξηση των απαιτήσεων σε όλους τους τομείς. Βρισκόμαστε σε μια εποχή που η επιστήμη των υπολογιστών και η τεχνολογία έχουν παγιωθεί και παίζουν καθοριστικό ρόλο τόσο στην πρόοδο της επιστήμης όσο και την βελτίωση της καθημερινότητας. Μέθοδοι και πρακτικές που άλλοτε αποτελούσαν αντικείμενο μόνο κάποιων εξειδικευμένων επιστημόνων σήμερα αντιμετωπίζονται ως μια συνήθης και τετριμμένη εργασία. Η εξέλιξη αυτή όπως είναι φυσικό κάλυψε τις ήδη υφιστάμενες ανάγκες αλλά δημιούργησε και νέες στοχεύοντας στην συνεχώς καλύτερη και ποιοτικότερη βελτίωση της ζωής. Νέες ιδέες εμφανίστηκαν στο προσκήνιο, υλοποιήθηκαν και εφαρμοστήκαν σε όλους τους τομείς από τους οποίους δεν θα μπορούσε να λείπει και η περιοχή των επιστημών ζωής. Η στάθμη υγείας ενός πληθυσμού αν και αποτελείται από πλήθος παραγόντων, αρκετές φορές ανεξάρτητων μεταξύ τους, μπορεί να βελτιωθεί σημαντικά χρησιμοποιώντας εργαλεία και μέσα της τεχνολογίας και της επιστήμης των υπολογιστών όχι μόνο στην έρευνα για αποτελεσματικότερες διαγνώσεις και την ανάπτυξη νέων θεραπειών αλλά και την οργάνωση και διαχείριση των ιατρικών δεδομένων[1]. Πέρα, όμως, από τον υλικοτεχνικό εξοπλισμό των υπηρεσιών υγείας και την ανάπτυξη δικτύων, το λογισμικό που χρησιμοποιείται και η ανάπτυξη εφαρμογών είναι ιδιάζουσας σημασίας. Πλήθος πρωτοκόλλων και εφαρμογών έχουν δημιουργηθεί για την διαχείριση, ανταλλαγή και επεξεργασία ιατρικών δεδομένων. Όλο και περισσότερες τεχνολογίες και εργαλεία τροποποιούνται ώστε να καλυφθούν οι ανάγκες στον τομέα της υγείας.

[1] Τα ιατρικά δεδομένα αποτελούνται από αποτελέσματα εξετάσεων, στοιχεία ασθενών και ιατρών, στατιστικά στοιχεία, ιστορικό ασθενών, πολυμεσικά δεδομένα, κα.

1.1 Ορισμός προβλήματος

Ο Παγκόσμιος Ιστός διαμοιράζει πληροφορίες αποθηκευμένες σε διαφορετικές γεωγραφικές περιοχές, το Πλέγμα, από την πλευρά του, υπολογιστική ισχύ και αποθήκευση δεδομένων κατανεμημένα σε όλο τον πλανήτη, τουλάχιστον σε θεωρητικό επίπεδο, στοχεύοντας στη συλλογή γεωγραφικά κατανεμημένων ετερογενών πόρων. Οι πόροι αυτοί αναφέρονται σε υπολογιστικές μονάδες, αποθηκευτικές μονάδες, αισθητήρες, οπτικοακουστικά εργαλεία και λογισμικό. Η τάση που επικρατεί σήμερα προσανατολίζεται στην μηχανογράφηση, τον αυτοματισμό και την επιτάχυνση διεργασιών για εξοικονόμηση χρόνου. Η χρήση του Διαδικτύου και των δικτύων γενικότερα, αξιοποιώντας τα διαθέσιμα εργαλεία και δυνατότητες που υπάρχουν, μπορεί να λειτουργήσει ως μια Βάση Γνώσης για την στήριξη αποφάσεων στην καθημερινή πρακτική και σε μελέτες.

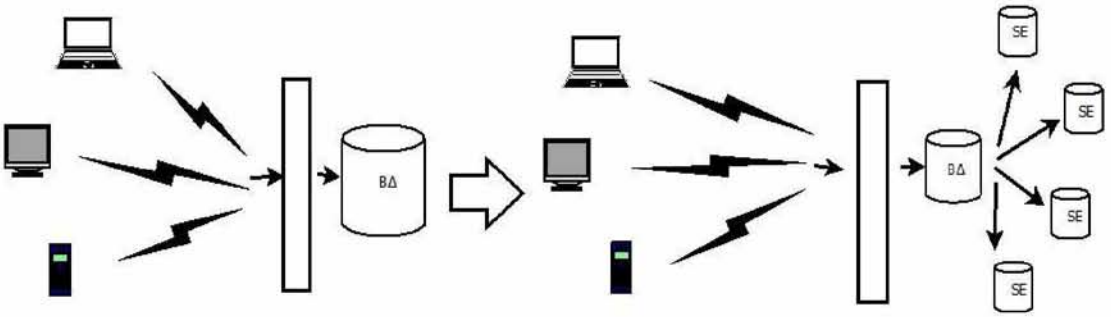
Η περιοχή των επιστημών ζωής βρίσκεται όλο και περισσότερο στο επίκεντρο των ερευνητών καθώς θέματα όπως η διαχείριση και επεξεργασία του μεγάλου όγκου των ιατρικών και βιοϊατρικών δεδομένων σε συνδυασμό με την ευαισθησία της συγκεκριμένης κατηγορίας δεδομένων καθιστούν αναγκαία την αναζήτηση και εφαρμογή νέων τεχνολογιών και μεθόδων, στοχεύοντας στην βελτίωση των παρερχομένων υπηρεσιών. Οι αυξανόμενες απαιτήσεις τόσο σε υπολογιστική ισχύ όσο και χωρητικότητα αποθήκευσης συνάντησαν τις τεχνολογίες πλέγματος (Grid Technologies), που έχουν ήδη εφαρμοστεί σε περιοχές όπως η φυσική, η πρόγνωση των κλιματικών αλλαγών και η μοντελοποίηση, παρέχοντας τη δυνατότητα συνεργασίας κατανεμημένων πηγών πέρα από τα όρια χωρών και ηπείρων. Το πλέγμα αποτελεί το επόμενο βήμα στον τομέα των κατανεμημένων συστημάτων καθώς εισάγει την έννοια της κοινής χρήσης και διαμοίρασης πόρων. Βέβαια, σε αντίθεση με τους παραδοσιακούς τομείς εφαρμογής τα ιατρικά δεδομένα και υπηρεσίες όπως έχει ήδη αναφερθεί απαιτούν ένα υψηλότερο βαθμό ασφάλειας, αποδοτικότητας και αξιοπιστίας[1].

Σκοπός αυτής της εργασίας είναι η δημιουργία μιας εφαρμογής βασισμένης στον ιστό (web-based) που θα λειτουργεί ως διεπαφή (portal)- πλατφόρμα που θα αποθηκεύει ιατρικά αρχεία και θα παρουσιάζει ιατρικές εικόνες με τα μεταδεδομένα τους με απλό τρόπο στον χρήστη. Οι εικόνες, φυσικά, και η ύπαρξή τους εντάσσονται σε ένα ευρύτερο σύστημα εφόσον αποθηκεύονται δημογραφικά στοιχεία των ασθενών στον οποίο ανήκουν, ποιος είναι ο υπεύθυνος ιατρός, ποια είναι τα στοιχεία της εξέτασης και το ίδιο το αρχείο που περιέχει το σύνολο της εξέτασης. Πέρα από αυτό, εξάγονται στοιχεία της επικεφαλίδας DICOM του αρχείου τα οποία παρέχουν επιπλέον πληροφορίες για τα χαρακτηριστικά του. Ο χρήστης από την πλευρά του θα μπορεί να χρησιμοποιεί την εφαρμογή και ως βιβλιοθήκη εικόνων, που λειτουργούν ενδεικτικά ενός αρχείου, στην οποία θα μπορεί να ανατρέχει ανά πάσα στιγμή και από οπουδήποτε αρκεί να έχει πρόσβαση στο διαδίκτυο και να έχει αποκτήσει τα κατάλληλα δικαιώματα χρήσης, να διαβάσει

τεχνικές πληροφορίες για το αρχείο και στοιχεία για τον ιδιοκτήτη του. Αυτό προϋποθέτει ότι ο χρήστης του συστήματος θα είναι ένα εξουσιοδοτημένο άτομο όπως για παράδειγμα ο χρήστης ενός Πληροφοριακού Συστήματος. Οι επιμέρους στόχοι που τίθενται προς υλοποίηση είναι η εισαγωγή, παρουσίαση, αποθήκευση και ανάκτηση των δεδομένων μιας εικόνας και των μετα-δεδομένων της, όπως τα δημογραφικά στοιχεία των ασθενών, πληροφορίες ή παρατηρήσεις που αφορούν την εικόνα και θεωρήθηκαν σημαντικά για την αρχική διάγνωση. Επίσης, θα παρέχονται πληροφορίες για τα χαρακτηριστικά του αρχείου όπως το μέγεθος, ο τύπος της εικόνας, το είδος επεξεργασίας στο οποίο υποβλήθηκε η εικόνα αν έχει υποστεί επεξεργασία και οποιαδήποτε σχόλια ή παρατηρήσεις που προστέθηκαν είτε από το χειριστή του μηχανήματος από το οποίο παράχθηκε η εικόνα είτε από τον υπεύθυνο γιατρό.

1.2 Αναγκαιότητα

Μια από τις προκλήσεις που αντιμετωπίζουν οι πάροχοι υπηρεσιών στο χώρο της επιστήμης των υπολογιστών αφορά τον καθορισμό των μεταξύ τους σχέσεων, καθώς η ανάγκη για πρόσβαση σε πλήθος υπηρεσιών δεν συνεπάγεται ότι όλοι οι χρήστες θα χρησιμοποιούν την ίδια πλατφόρμα. Παραδοσιακά, οι προγραμματιστές που ασχολούνται με την ανάπτυξη εφαρμογών συνήθως εργάζονται σε ένα ομογενές, αξιόπιστο, ασφαλές και κεντρικά ελεγχόμενο περιβάλλον[8]. Τίθεται, λοιπόν, το ερώτημα γιατί είναι απαραίτητη η κατανεμημένη επεξεργασία και, συνεπώς, γιατί χρειαζόμαστε τη χρήση του πλέγματος (grid) στην επεξεργασία, οργάνωση και διαχείριση των ιατρικών δεδομένων. Όπως έχει ήδη ειπωθεί σε γενικές γραμμές, οι σύγχρονες ανάγκες επιτάσσουν την αποτελεσματικότερη και αποδοτικότερη διαχείριση των ιατρικών δεδομένων, κάτι που μπορεί να πραγματοποιηθεί άμεσα με κατανεμημένα συστήματα και τη χρήση του διαδικτύου παρέχοντας στους χρήστες τη δυνατότητα πρόσβασης σε μια βάση δεδομένων και σε αρχεία δεδομένων από οπουδήποτε συνδυάζοντας, όμως, και την δυνατότητα αποθήκευσης αυτών οπουδήποτε δίνοντας έτσι τη ευκαιρία στους ποικίλους παρόχους υπηρεσιών υγείας να επικοινωνούν μεταξύ τους για την προαγωγή της υγείας των ασθενών εξαλείφοντας τους γεωγραφικούς περιορισμούς. Στην Εικόνα 1 παρατηρούμε τη διαφορά μεταξύ των δύο προσεγγίσεων (απλή αποθήκευση σε μια Βάση Δεδομένων και αποθήκευση στο Πλέγμα). Έτσι, η αλληλεπίδραση συστημάτων και οργανισμών κινείται στην κατεύθυνση “ευφιών” δικτύων αδυνατώντας να παραβλέψουν τον παράγοντα κόστος.



Εικόνα 1: Η ανάγκη αποθήκευσης μεγάλου όγκου δεδομένων οδήγησε στην αναζήτηση πιο αποδοτικών λύσεων.

Ακόμη, η χρήση του διαδικτύου δίνει τη δυνατότητα της ανάκτησης δεδομένων, χρήσιμα για την αξιολόγηση ή/και υποβοήθηση μιας διάγνωσης ή την προβολή των ραδιολογικών εξετάσεων ενός ασθενή, χωρίς απαραίτητα να βρίσκεται στο χώρο όπου παρήχθησαν οι εξετάσεις ή σε μια τοπική βάση δεδομένων που πιθανότατα είναι αποθηκευμένες γνωρίζοντας τους φυσικούς περιορισμούς καθώς ο μεγάλος όγκος δεδομένων αποτελεί ανασταλτικό παράγοντα. Για παράδειγμα, πολλές φορές τα δεδομένα των ασθενών διαγράφονται μετά από κάποιο χρονικό διάστημα ή κρατώνται μόνο κάποιες εγγραφές και όχι λεπτομερή στοιχεία. Ένα ακόμα μειονέκτημα που καλείται να αντιμετωπιστεί είναι η μεταφερσιμότητα των δεδομένων αυτών που συνήθως γίνεται κατόπιν συνεννοήσεως των ειδικών (γιατροί, ραδιολόγοι, κα) ή μέσω των ίδιων των ασθενών. Τέλος, πρέπει να αναφερθεί ότι η ύπαρξη ενός portal ιατρικών εικόνων μπορεί να υποστηρίξει την πραγματοποίηση συγκριτικών μελετών καθώς οι ερευνητές θα μπορούν να αντλήσουν στοιχεία βάσει κάποιων χαρακτηριστικών ή παρατηρήσεων όπως η σχέση δημογραφικών στοιχείων (φύλο, ηλικία) με την εμφάνιση συγκεκριμένων νόσων. Είναι εμφανές ότι ένα τέτοιο σύστημα μπορεί να ενσωματώσει ή να συνδεθεί με τον ηλεκτρονικό ιατρικό φάκελο του ασθενή αφού παρέχει το σύνολο των δεδομένων που αφορούν τον ασθενή και το γιατρό του, καθώς και τις σχετικές με την απεικόνιση εξετάσεις.

1.3 Θεματικές Περιοχές

Η υλοποίηση αυτής της εργασίας προϋποθέτει την γνώση των ακόλουθων θεματικών περιοχών:

- ✓ Βάσεις Δεδομένων
- ✓ Διαδικτυακός Προγραμματισμός
- ✓ Μεσεισμικό Λογισμικό (Middleware Software)
- ✓ Ιατρικη Πληροφορική

- Βάσεις Δεδομένων

Το βασικό χαρακτηριστικό των σύγχρονων εφαρμογών είναι η απαίτηση για αποτελεσματική και αποδοτική διαχείριση της αποθηκευμένης

πληροφορίας. Σε περιπτώσεις όπου ο όγκος των δεδομένων είναι σχετικά μεγάλος και η πολυπλοκότητα της πληροφορίας αυξάνει, η διαχείριση της από το λειτουργικό σύστημα δεν είναι αρκετή. Η πολυπλοκότητα διαχείρισης, για παράδειγμα, των αριθμητικών δεδομένων δεν ισοδυναμεί με την αντίστοιχη μιας εφαρμογής που στοχεύει στην οργάνωση μιας βάσης, όπου απαιτείται η αναζήτηση εικόνων με βάση το όνομα του ασθενή ή την ημερομηνία της εξέτασης. Η δημιουργία μιας βάσης δεδομένων δίνει τη δυνατότητα της αποδοτικής διαχείρισης στην περίπτωση που η δομή της πληροφορίας είναι περίπλοκη (πχ. ένας ασθενής μπορεί να έχει κάνει περισσότερες από μια εξετάσεις), παρέχει ευελιξία στην εισαγωγή νέων στοιχείων και την ανανέωση των δεδομένων και, τέλος, χρησιμοποιούνται μηχανισμοί ταυτόχρονης προσπέλασης της εφαρμογής από διαφορετικούς χρήστες. Πέρα, λοιπόν, από τις κλασικές εφαρμογές των συστημάτων διαχείρισης βάσεων δεδομένων (κρατήσεις θέσεων, τραπεζικές συναλλαγές, διαχείριση εταιρικών δεδομένων) οι σύγχρονες ανάγκες όπως η υποστήριξη νέων τύπων δεδομένων, όπως ήχο, εικόνα, βίντεο, και μεθόδων επεξεργασίας οδήγησαν στην ανάπτυξη εφαρμογών που καλύπτουν αυτές τις ιδιαιτερότητες (πολυμεσικές εφαρμογές, διαχείριση γεωγραφικών δεδομένων, αποθήκες δεδομένων, εξόρυξη δεδομένων, ανάκτηση πληροφορίας από το διαδίκτυο, διαχείριση κινούμενων αντικειμένων). Η υλοποίηση μιας βάσης δεδομένων απαιτεί τη χρήση μιας γλώσσας αποτελούμενη από δύο τμήματα: τη γλώσσα ορισμού των δεδομένων και τη γλώσσα χειρισμού των δεδομένων. Η πρώτη αφορά τον ορισμό των οντοτήτων και των μεταξύ τους σχέσεων και η δεύτερη την εισαγωγή, την επεξεργασία, την ενημέρωση και την διαγραφή δεδομένων [15]. Η SQL (Structured Query Language) είναι σύμφωνα και με το όνομα της μια δομημένη γλώσσα ερωτημάτων και επιτελεί τις ανωτέρω λειτουργίες. Σε αυτή την εργασία η βάση δεδομένων που λειτουργεί υποστηρικτικά της εφαρμογής υλοποιήθηκε κάνοντας χρήσης της SQL και αφορά τις λειτουργίες της δημιουργίας και ενημέρωσης πινάκων, επιλογής και εισαγωγής στοιχείων σε αυτούς.

- Διαδικτυακός Προγραμματισμός και Τεχνολογίες Ιστού

Το διαδίκτυο περιλαμβάνει πέντε βασικές εφαρμογές[2] μεταξύ των οποίων είναι η απομακρυσμένη σύνδεση, όπου με τη βοήθεια του Telnet, του Secure Shell (SSH) ή κάποιου συναφούς προγράμματος ένας χρήστης μπορεί να αποκτήσει πρόσβαση σε οποιοδήποτε άλλο μηχάνημα έχει λογαριασμό και η μεταφορά αρχείων με τη χρήση ενός προγράμματος FTP (File Transfer Protocol). Η πλέον γνωστή εφαρμογή του είναι ο Παγκόσμιος ιστός που χρησιμοποιεί υπερ-συνδέσεις (hyperlinks) για να συνδέσει το περιεχόμενο

[2] Ηλεκτρονικό ταχυδρομείο, απομακρυσμένη σύνδεση, FTP, www, ομάδες νέων

του ιστού. Η γλώσσα που χρησιμοποιείται στον ιστό για την κατασκευή ιστοσελίδων είναι η HTML (HyperText Markup Language) που δίνει την δυνατότητα δημοσίευσης εγγράφων με επικεφαλίδες, κείμενο, πίνακες, λίστες, φόρμες και ανάκτησης πληροφοριών μέσω υπερσυνδέσεων. Μια καλά συνεργαζόμενη γλώσσα κατασκευής δυναμικών ιστοσελίδων με την HTML είναι η PHP [16]. Οι δυναμικές ιστοσελίδες χρησιμοποιούν εφαρμογές που εκτελούνται στην πλευρά του εξυπηρετητή και λαμβάνουν δεδομένα που εισάγει ο χρήστης, τα επεξεργάζονται ή ανατρέχουν σε μια βάση δεδομένων (που είναι εγκατεστημένη στον εξυπηρετητή) εμφανίζοντας μια ιστοσελίδα προσαρμοσμένη στις ανάγκες της εφαρμογής. Όταν, δηλαδή, κάποιος χρήστης επισκεφθεί μια ιστοσελίδα που περιέχει κώδικα PHP, ο εξυπηρετητής εκτελεί τον κώδικα και αυτό που επιστρέφει στον χρήστη είναι μια σελίδα HTML. Η χρήση της PHP είναι, επίσης, κατάλληλη εξαιτίας της συμβατότητας της με την SQL οπότε δίνει τη δυνατότητα επικοινωνίας με τη βάση δεδομένων. Στα πλαίσια των διαδικτυακών εφαρμογών εντάσσονται και οι Υπηρεσίες Ιστού ή Web Services παρέχοντας τη δυνατότητα εκτέλεσης και ενσωμάτωσης ανεξάρτητα γραμμένων προγραμμάτων σε μια εφαρμογή κάνοντας τις κατάλληλες τροποποιήσεις και ενημερώσεις, ώστε να ταιριάζουν με τις ανάγκες μιας εφαρμογής.

- Μεσεισμικό Λογισμικό (Middleware Software)

Ο όρος gLite αναφέρεται στο λογισμικό που χρησιμοποιείται στους υπολογισμούς σε περιβάλλον πλέγματος και παρέχει ένα πλαίσιο για την ανάπτυξη εφαρμογών. Οι υπηρεσίες που προσφέρει το gLite, μέχρι σήμερα έχει υιοθετηθεί από 250 και πλέον Υπολογιστικά Κέντρα και χρησιμοποιούνται από τουλάχιστον 15.000 ερευνητές στην Ευρώπη και τον κόσμο. Η πρώτη φάση υλοποίησης ξεκίνησε από το 2004-2005 με την έκδοση του LCG-2, αλλά η τελική έκδοση ήρθε το Μάιο του 2006 όταν το gLite 3.0 έγινε το επίσημο λογισμικό του EGEE. Πιο συγκεκριμένα, το λογισμικό αυτό δίνει τη δυνατότητα στο χρήστη να αποκτήσει πρόσβαση σε πόρους και συστήματα και να παρακολουθεί την δραστηριότητα στο πλέγμα μέσω μιας διεπαφής χρήστη που αποτελεί και τον τρόπο επικοινωνίας και πρόσβασης στην υποδομή. Ο χρήστης μπορεί να χρησιμοποιήσει εντολές υψηλού επιπέδου για να επιτελέσει τις λειτουργίες αποστολής εργασιών στο πλέγμα και ανάκτησης των αποτελεσμάτων μετά το πέρας της επεξεργασίας, να διαχειριστεί τα δεδομένα και τα αντίγραφα τους. Θεωρείται ότι αποτελεί την επόμενη γενιά λογισμικού στην περιοχή των υπολογισμών σε πλέγμα (grid computing). Το λογισμικό gLite περιέχει εντολές υψηλού και χαμηλού επιπέδου (command-line) για την εκτέλεση διάφορων λειτουργιών όπως περιγράφονται στην επίσημη ιστοσελίδα του Hellas Grid και, ουσιαστικά, παρέχουν ένα πλαίσιο για την υλοποίηση εφαρμογών εκμεταλλευόμενοι την δύναμη του κατανεμημένου υπολογισμού και πλήθος αποθηκευτικών πηγών μέσω του Διαδικτύου.

Περισσότερες πληροφορίες σχετικά με το gLite, τις λειτουργίες του και πως χρησιμοποιείται υπάρχουν στο τρίτο κεφάλαιο.

- Ιατρική Πληροφορική

"Η Ιατρική Πληροφορική είναι ο κλάδος που ασχολείται με τη συστηματική επεξεργασία των δεδομένων, της πληροφορίας και της γνώσης. Το πεδίο της καλύπτει υπολογιστικές και πληροφοριακές πλευρές των διαδικασιών και των δομών της φροντίδας υγείας." [18] Η μελέτη των γενικών αρχών της επεξεργασίας των δεδομένων, της πληροφορίας και της γνώσης στοχεύουν στην εύρεση λύσης στους σχετικούς τομείς. Η Ιατρική Πληροφορική χρησιμοποιεί πρότυπα για την ανάλυση, κατασκευή, διατήρηση και ανάπτυξη συστημάτων σε συνδιασμό με προγράμματα υπολογιστών και εφαρμογές. Ο Van Bemmel [18] διακρίνει τα ακόλουθα επίπεδα ιατρικής πληροφορικής:

1. Επικοινωνία και καταγραφή
2. Αποθήκευση και ανάκτηση δεδομένων
3. Υπολογισμός
4. Αναγνώριση και διάγνωση
5. Θεραπεία και έλεγχος
6. Έρευνα και μοντελοποίηση

Η Ιατρική Πληροφορική καλύπτει αρκετές περιοχές θεμάτων όπως συστήματα επικοινωνίας και δίκτυα, Πληροφοριακά Συστήματα Υγείας, Ηλεκτρονικός Φάκελος Ασθενή, επεξεργασία σημάτων και εικόνων και βάσεις γνώσεων [19]. Από τα παραπάνω, εστιάζοντας στην επεξεργασία εικόνας, μεσώ τεχνικών που επιτρέπουν τη διαχείριση και την μετατροπή τους εξάγονται πληροφορίες και χαρακτηριστικά που δεν ειναι εμφανή και αντιληπτά.

Ο μεγάλος όγκος των δεδομένων που προκύπτουν από την ψηφιοποίηση των εικόνων ήταν αδύνατο να διαχειριστούν με τα υπάρχοντα Νοσοκομειακά Πληροφοριακά Συστήματα (Hospital Information Systems - HIS) και Ραδιολογικά Πληροφοριακά Συστήματα (Radiological Information Systems – RIS) [20]. Για αυτό το λόγο θεωρήθηκε απαραίτητη η ανάπτυξη συστημάτων ψηφιακής αρχειοθέτησης και επικοινωνίας εικόνων με σημαντικότερα τα παρακάτω μοντέλα: PACS (Picture Archiving and Communication System) και ISACS (Image Save and Carry System). Το σύστημα PACS συνδιάζει την παρακολούθηση ιατρικών εικόνων από σταθμούς εργασίας καθώς και αρχειοθέτηση των εικόνων σε ηλεκτρονικά μέσα αποθήκευσης με δυνατότητα επικοινωνίας και ανταλλαγής μέσω δικτύων. Η αποθήκευση των δεδομένων από ένα σύστημα PACS διακρίνεται σε τρία βασικά επίπεδα, στο τρίτο από τα οποία επιχειρούμε να επέμβουμε και ειναι τα ακόλουθα:

1. Σταθμός εργασίας, όπου πραγματοποιείται η πρώτη διάγνωση και παρακολούθηση

2. Κεντρική Βάση Δεδομένων, όπου πραγματοποιείται συγκριση με παλαιότερες εξετάσεις
3. Οπτικοί δίσκοι, όπου γίνεται η μακροχρόνια αποθήκευση των εικόνων

1.4 Δομή της εργασίας

Η εργασία αποτελείται από πέντε κεφάλαια των οποίων το περιεχόμενο περιγράφεται συνοπτικά σε αυτή την ενότητα.

Το πρώτο κεφάλαιο αποτελεί την εισαγωγή στο θέμα της εργασίας και παρουσιάζεται το πρόβλημα που καλείται να αντιμετωπίσει. Επισημαίνει ποιες ανάγκες οδήγησαν στην αναζήτηση μιας υλοποίησης κατανεμημένης εργασίας και σε ποια προβλήματα αναζητείται λύση. Στη συνέχεια, προσδιορίζεται ο σκοπός της εργασίας, καθορίζονται οι επιμέρους στόχοι και τα στοιχεία που καθιστούν αναγκαία την υλοποίηση της εφαρμογής που παρουσιάζεται στα επόμενα κεφάλαια. Παρουσιάζονται, συνοπτικά, οι θεματικές περιοχές που συνέβαλαν στην υλοποίηση και ολοκλήρωση της εργασίας μεταξύ των οποίων είναι οι βάσεις δεδομένων, ο διαδικτυακός προγραμματισμός ή γενικότερα οι τεχνολογίες ιστού και το λογισμικό gLite και, τέλος, παρουσιάζεται η δομή που ακολουθεί η εργασία περιγράφοντας περιληπτικά το περιεχόμενο κάθε κεφαλαίου.

Το κεφάλαιο 2 καλύπτει την ανάγκη μιας πρώτης επαφής με τις έννοιες που θα χρησιμοποιηθούν σε αυτή την εργασία όπως η έννοια του πλέγματος, η εξέλιξή του και ο βαθμός απήχησης του στον βιομηχανικό και ακαδημαϊκό κόσμο. Διευκρινίζεται τι εννοούμε όταν χρησιμοποιούνται οι όροι παράλληλη και κατανεμημένη επεξεργασία. Παρουσιάζονται οι εναλλακτικοί τρόποι που μπορούμε να κάνουμε παραλληλισμό, δηλαδή παραλληλισμός δεδομένων, παραλληλισμός εντολών και παραλληλισμός προγράμματος, καθώς και ποιος είναι ο ρόλος των συστάδων υπολογιστών (clusters). Επιπλέον, αναφέρονται οι διαφορές και οι ομοιότητες των Clusters και των Grids. Στη συνέχεια του κεφαλαίου, γίνεται μια ανασκόπηση των ερευνών που έχουν λάβει χώρα και εφαρμογών που έχουν πραγματοποιηθεί σχετικά με το υπό μελέτη πρόβλημα υποδεικνύοντας το εύρος και την ποικιλία των υλοποιήσεων. Παρουσιάζονται έρευνες καθώς και ήδη υλοποιημένες εφαρμογές οι οποίες συνδυάζουν τις ανάγκες των υπηρεσιών υγείας όσο και τον τρόπο με τον οποίο εκμεταλλευτήκαν τις δυνατότητες και τα εργαλεία της τεχνολογίας όπως το πλέγμα ή οι κατανεμημένες βάσεις δεδομένων. Μερικές από τις εργασίες που παρουσιάζονται σε αυτό το κεφάλαιο είναι τα MediGrid, MAGIC-5, AGIR, EKTORAS [7] και DataGrid [3] παρατηρώντας ότι κάθε μια από αυτές χρησιμοποιεί την υποδομή με διαφορετικό τρόπο και για διαφορετικό σκοπό. Η πλατφόρμα του MediGrid [1] αποτελεί την υλοποίηση του Γερμανικού έργου εφαρμογής του πλέγματος στις επιστήμες ζωής, περιγράφει την εφαρμογή του

πρωτοκόλλου GRID-DICOM και τα συμπεράσματα που εξήχθησαν από την εκτέλεση συγκεκριμένων σεναρίων. Η εφαρμογή EKTORAS [7] δίνει την δυνατότητα στον τελικό χρήστη μέσω μιας πλατφόρμας με εύκολο και απλό τρόπο να εκτελέσει πειράματα μικροσυστοιχιών DNA χρησιμοποιώντας την υποδομή πλέγματος, τόσο για την εκτέλεση παράλληλων εργασιών όσο και για την αποθήκευση των αποτελεσμάτων σε αυτό. Ο σκοπός του έργου MAGIC-5 [8] είναι η ανάπτυξη λογισμικού CAD (computer aided detection: ανίχνευση υποβοηθούμενη από υπολογιστές) για ιατρικές εφαρμογές σε κατανεμημένες βάσεις δεδομένων εκμεταλλευόμενοι τη σύνδεση με την υποδομή πλέγματος. Έχει ήδη εφαρμοστεί επιτυχώς σε εφαρμογές στη μαστογραφία και αναπτύσσονται εφαρμογές για την ανίχνευση κακοήθους όγκου στους πνεύμονες από εικόνες αξονικού τομογράφου ενώ εφαρμόζεται και η ανάλυση εικόνων PET για τη νόσο Alzheimer στο πλέγμα. Σε αυτή την εργασία δίνεται έμφαση σε θέματα επεξεργασίας και διαχείρισης δεδομένων. Η εφαρμογή AGIR δίνει έμφαση στη δημιουργία ενός πλαισίου ικανού να καλύψει την ανάπτυξη μιας ιατρικής εφαρμογής. Το DataGrid μπορεί να θεωρηθεί ως μια ξεχωριστή κατηγορία εφαρμογών αφού εστιάζεται στις λειτουργίες της αποθήκευσης και ανάκτησης των δεδομένων. Γίνεται μια προσπάθεια να καθοριστούν τα δικαιώματα των χρηστών και ο βαθμός πρόσβασής τους στα δεδομένα προτείνοντας μια αρχιτεκτονική διαχείρισης των ιατρικών δεδομένων [3]. Τέλος, οι Chun-Hsi Huang et. Al [4] παρουσιάζουν την κατάσταση που επικρατεί σήμερα παγκοσμίως στον τομέα των επιστημών ζωής υπό το πρίσμα της τεχνολογίας, ποιες βελτιώσεις έχουν γίνει και ποιοι ενδοιασμοί είναι ακόμη παρόντες, κάνοντας αναφορά από ηθικά και νομικά σε τεχνικά ζητήματα υπό την ευρύτερη έννοια του Biogrid.

Το κεφάλαιο 3 τιτλοφορείται “προτεινομένη αντιμετώπιση” και παρουσιάζει ποιες είναι οι κατευθυντήριες γραμμές, τα βήματα και τα μέσα που θα χρησιμοποιηθούν για την υλοποίηση της παρούσας εργασίας. Γίνεται αναφορά στα συστατικά στοιχεία του πλέγματος, τις δυνατότητες που παρέχει και τον τρόπο με τον οποίο θα χρησιμοποιηθούν. Πιο συγκεκριμένα, παρουσιάζονται οι βασικές αρχές υποβολής εργασιών στο πλέγμα και τα βασικά στοιχεία του λεξιλογίου της γλώσσας JDL. Η σύνδεση της εφαρμογής με το πλέγμα αποτελεί ένα από τα βασικά σημεία της υλοποίησης για αυτό το λόγο περιγράφονται αρκετά αναλυτικά οι διαδικασίες για την απόκτηση πρόσβασης στην υποδομή, υποβολής εργασιών και διαχείρισης των δεδομένων. Ακολουθεί μια σύντομη εισαγωγή στο πρότυπο DICOM εφόσον τα δεδομένα που καλούνται να διαχειριστούν σε αυτή την εργασία είναι τύπου DICOM. Για τον χειρισμό των εικόνων όπως η μετατροπή τους από DICOM σε JPEG μορφή (κυρίως για λόγους συμπίεσης αποτελώντας προαιρετική λειτουργία) και την εξαγωγή πληροφοριών από τα αρχεία DICOM χρησιμοποιούνται ήδη υλοποιημένα πακέτα με κώδικα PHP αφού τροποποιηθούν κατάλληλα για να είναι δυνατή η ενσωμάτωσή τους στο σύστημα. Ακόμη, γίνεται αναφορά στην λειτουργία και συνεργασία της PHP με την MySQL και ο τρόπος με τον οποίο ο συνδυασμός τους

συνεισφέρει στην κατασκευή της δυναμικής μας ιστοσελίδας. Παρουσιάζεται λεπτομερώς πως έχει σχεδιαστεί η εκτέλεση των λειτουργιών της εφαρμογής όπως η εισαγωγή, αποθήκευση, παρουσίαση και ανάκτηση των δεδομένων και ποιες είναι οι δυνατότητες αναζήτησης στη βάση δεδομένων. Η βάση δεδομένων που δημιουργήθηκε περιγράφεται σχηματικά από τους πίνακες που την αποτελούν και από τις μεταξύ τους σχισιές. Ακόμη, γίνεται μια σύντομη εισαγωγή και περιγραφή των Web Services και εξηγείται ο τρόπος με τον οποίο έχουν χρησιμοποιηθεί. Ουσιαστικά, αυτό το κεφάλαιο κλείνει παρουσιάζοντας τη συνολική αρχιτεκτονική του συστήματος και πως τελικά ο συνδυασμός όλων των επιμέρους στοιχείων οδήγησε στην εξαγωγή του τελικού αποτελέσματος.

Το 4ο κεφαλαίο παρουσιάζει αναλυτικά την λειτουργία του συστήματος ακολουθώντας βήμα προς βήμα τον τρόπο με τον οποίο ο χρήστης αποκτά πρόσβαση και κάνει χρήση των υπηρεσιών που προσφέρονται. Εν συντομία, ο χρήστης αφού πληκτρολογήσει το "όνομα" και τον "κωδικό" του και αναγνωριστεί με επιτυχία από τη βάση δεδομένων, μπορεί να καταχωρήσει έναν ασθενή ή έναν γιατρό συμπληρώνοντας απλά μια φόρμα υποβολής στοιχείων, να καταχωρήσει μια εξέταση επιλέγοντας τον ασθενή ή τον γιατρό αν οι ίδιοι δεν έχουν απομνημονεύσει τον κωδικό τους (ID) και, τέλος, μπορούν να αντιγράψουν τα επιθυμητά αρχεία στο πλέγμα. Ακόμη, δίνεται η δυνατότητα αναζήτησης στοιχείων σε όλη τη βάση δεδομένων και προβολή μιας συλλογής αρχείων τύπου JPEG τα οποία λειτουργούν αντιπροσωπευτικά των αρχικών αρχείων που μπορούν να κατεβούν τοπικά. Το κεφάλαιο αυτό μπορεί, λοιπόν, να λειτουργήσει ως οδηγός χρήσης για τον τρόπο λειτουργίας της εφαρμογής καθώς, επίσης, και των δυνατοτήτων που προσφέρει.

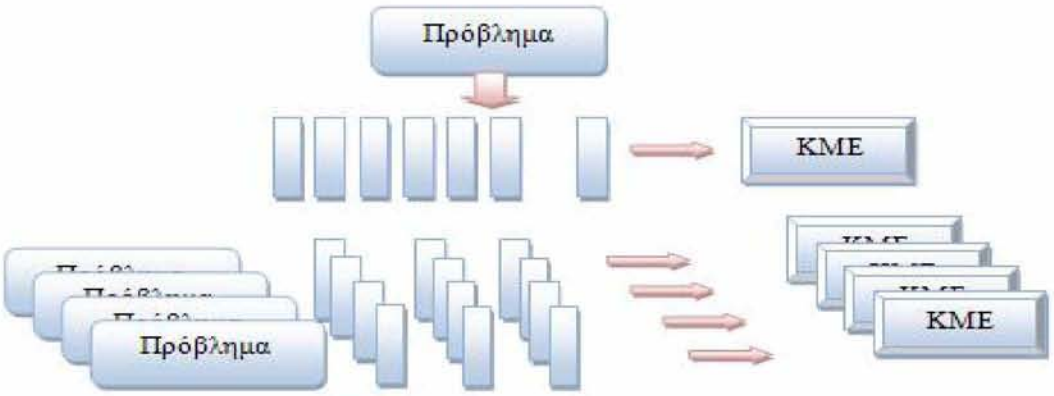
Το 5ο κεφάλαιο κλείνει την εργασία αναφέροντας όσες παρατηρήσεις έγιναν κατά την υλοποίηση, ποιοι προβληματισμοί και εμπόδια υπήρξαν (και πως αξιολογείται το τελικό αποτέλεσμα). Αποτελεί ένα πρώτο βήμα στον κόσμο της απεικόνισης ιατρικών εικόνων μέσα από την διαχείρισή τους με έναν τρόπο ξεχωριστό κάνοντας χρήση των νέων τεχνολογιών που αναπτύσσονται και βελτιώνονται καθημερινά, δίνοντας ώθηση σε ακόμα πιο φιλόδοξους στόχους όπως είναι η επεξεργασία και ανάλυση των ιατρικών δεδομένων όχι μόνο αποκλειστικά σε αρχεία εικόνων (CT, MRI, PET, US) αλλά και επεξεργασία βίντεο με παράλληλο ή κατανεμημένο τρόπο.

2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ ΚΑΙ ΜΕΛΕΤΕΣ ΕΦΑΡΜΟΓΩΝ ΠΕΡΙΠΤΩΣΗΣ (CASE STUDIES)

Η διαφορετικότητα αποτελεί στόχο για πολλούς ανθρώπους και στοιχείο προβληματισμού για πολλούς άλλους. Πολλές φορές στοιχεία που με μια πρώτη ματιά μοιάζουν ετερογενή και εντελώς ξένα μεταξύ τους αν εξεταστούν από μια συγκεκριμένη οπτική γωνία μπορούν να οδηγήσουν σε ιδιαίτερα ενδιαφέρουσες παρατηρήσεις και συμπεράσματα. Ακόμη, η συνεργασία διαφορετικών στοιχείων, πιο περιορισμένη ή πιο εκτεταμένη, μπορεί να χτίσει νέα καινοτόμα και πρωτότυπα εργαλεία, εφαρμογές και θεωρίες. Σε αυτό το κεφάλαιο παρουσιάζονται συστήματα που αναπτύχθηκαν έχοντας παρόμοιους στόχους αλλά ακολούθησαν διαφορετικές προσεγγίσεις και μεθοδολογίες. μεταξύ των οποίων είναι οι MediGrid, AGIR, DataGrid, HECTOR. Κάθε σύστημα από αυτά που παρουσιάζονται στη συνέχεια χρειάστηκε να συνδυάσει ποικιλία στοιχείων και μονάδων για την ανάπτυξη λειτουργικών εφαρμογών όσο αταίριαστα κι αν έμοιαζαν σε πρώτη φάση. Εδώ περιγράφονται κάποια από τα βασικά τους χαρακτηριστικά αφού προηγείται μια θεωρητική εισαγωγή στην παράλληλη ή κατανεμημένη επεξεργασία δεδομένων, στις έννοιες των Grid και Cluster και των μεταξύ τους διαφορών. Το Πλέγμα αποτελεί ένα από τα βασικά στοιχεία αυτής της εργασίας για αυτό το λόγο γίνεται μια πρώτη γνωριμία με αυτό, μέσα από μια ιστορική αναδρομή και την περιγραφή κάποιων ιδιοτήτων του.

2.1 Παράλληλη και κατανεμημένη επεξεργασία δεδομένων

Ο κόσμος γύρω μας είναι γεμάτος από παραδείγματα παράλληλης επεξεργασίας αλλά σε κάθε μας δραστηριότητα υιοθετούμε έναν σειριακό τρόπο αντιμετώπισης και δράσης. Έτσι και η ανάπτυξη λογισμικού, παραδοσιακά, γίνεται με σειριακό τρόπο, δηλαδή μια μονάδα επεξεργασίας, που εκτελεί ένα διακριτό πλήθος εντολών, την μια μετά την άλλη, και έχει το χαρακτηριστικό ότι κάθε στιγμή εκτελείται μόνο μια εντολή. Αντίθετα, η παράλληλη επεξεργασία χρησιμοποιεί πολλές μονάδες επεξεργασίας ή μια πολύ-πύρηνη μονάδα επεξεργασίας, όπου η καθεμία εκτελεί διακριτά τμήματα εντολών για την επίλυση ενός προβλήματος ταυτόχρονα. Δηλαδή κάθε τμήμα χωρίζεται σειριακά σε εντολές και αυτές εκτελούνται την ίδια στιγμή. Οι κυριότεροι λόγοι που στρέφουμε το βλέμμα στην παράλληλη επεξεργασία είναι για εξοικονόμηση χρόνου και χρήματος, για επίλυση αρκετά μεγάλων προβλημάτων, για την προώθηση του ταυτοχρονισμού, την χρήση μη τοπικών πηγών και την άρση των περιορισμών της σειριακής επεξεργασίας. Η φιλοσοφία εκτέλεσης μιας παράλληλης εργασίας συγκρινόμενη με την σειριακή παρουσιάζεται στην Εικόνα 2. Στην περίπτωση της παράλληλης επεξεργασίας το πρόβλημα χωρίζεται σε υπό-προβλήματα καθένα από τα οποία εκτελείται σειριακά σε κάποια μονάδα επεξεργασίας. Όταν, όμως, αναφερόμαστε σε παραλληλισμό κάνουμε διάκριση μεταξύ δεδομένων και επεξεργαστών.



Εικόνα 2: Η διαφορά μεταξύ σειριακής και παράλληλης επεξεργασίας και αντιμετώπισης ενός προβλήματος.

Υπάρχουν τέσσερις διάφοροι τρόποι ταξινόμησης των παράλληλων επεξεργαστών, ένας, όμως, δείχνει να είναι ιδιαίτερα δημοφιλής και αναφέρομαι στην ταξινόμηση (ή ταξινομία) του Flynn [6] και παρουσιάζεται στη συνέχεια.

Πίνακας 1: Οι 4 διαφορετικοί τρόποι ταξινόμησης των παράλληλων επεξεργαστών

SISD Single Instruction, Single Data	SIMD Single Instruction, Multiple Data
MISD Multiple Instruction, Single Data	MIMD Multiple Instruction, Multiple Data

Όταν αναφερόμαστε σε SISD, ουσιαστικά, μιλάμε για έναν σειριακό επεξεργαστή, όπου σε κάθε κύκλο ρολογιού εκτελείται μόνο μια εντολή ή εισάγεται ως είσοδος μόνο μια ομάδα οδηγιών και ακολουθεί ντετερμινιστική εκτέλεση. Η δεύτερη κατηγορία (SIMD) αποτελεί ένα είδος παράλληλης επεξεργασίας, καθώς κάθε επεξεργαστής εκτελεί την ίδια ομάδα εντολών σε διαφορετικά δεδομένα. Ο συγκεκριμένος τρόπος εκτέλεσης βρίσκει εφαρμογή και στην ανάλυση εικόνων. Η τρίτη κατηγορία (MISD) αφορά την επεξεργασία των δεδομένων τα οποία υπόκεινται σε διαφορετική επεξεργασία από διαφορετικό επεξεργαστή και, τέλος, η MIMD που αποτελεί τον πιο κοινό τύπο παράλληλων επεξεργαστών με κάθε επεξεργαστή να δουλεύει εκτελώντας διαφορετικές οδηγίες σε διαφορετικά δεδομένα. Παραδείγματα τέτοιων εφαρμογών αποτελούν οι υπέρ - υπολογιστές, τα "clusters" και τα "grids" μεταξύ άλλων. Άρα, όταν κάνουμε αναφορά σε παράλληλη επεξεργασία γνωρίζουμε ότι μπορούμε να εκτελέσουμε πολλές εργασίες ταυτόχρονα έχοντας εξαλείψει ή ρυθμίσει τις όποιες εξαρτήσεις των δεδομένων καθώς τμήματα κώδικα ίσως χρειάζονται αποτελέσματα άλλων εντολών για να συνεχίσουν. Στις περισσότερες περιπτώσεις παράλληλης επεξεργασίας τα δεδομένα αποθηκεύονται σε μια κοινά διαμοιραζόμενη μνήμη από την οποία όλοι οι επεξεργαστές αντλούν τα δεδομένα εισόδου τους. Η επικοινωνία και ο συγχρονισμός μεταξύ των διαφορετικών υπό-εργασιών είναι από τα μεγαλύτερα εμπόδια που καλούνται να αντιμετωπιστούν για την επίτευξη καλής απόδοσης. Από την άλλη πλευρά, η κατανεμημένη επεξεργασία θα μπορούσε να θεωρηθεί ως μια γενίκευση της παράλληλης επεξεργασίας και αφορά την εκτέλεση διαφορετικής ομάδας εντολών ή δεδομένων όχι απαραίτητα στον ίδιο χρόνο και χωρίς η συνέχιση της εκτέλεσης μιας εργασίας να απαιτεί την ολοκλήρωση κάποιας άλλης. Κάθε επεξεργαστής χρησιμοποιεί τη δική του μνήμη (μια τοπική μνήμη). Σε αυτή την περίπτωση, ο προγραμματιστής είναι υπεύθυνος για τον τρόπο οργάνωσης της επικοινωνίας μεταξύ των επεξεργαστών. Ο όρος κατανεμημένο σύστημα αναφέρεται τόσο στο υλικό (επεξεργαστές, μνήμες, δίκτυο) όσο και το λογισμικό (λειτουργικό σύστημα, εφαρμογές) που είναι απαραίτητα για την υλοποίηση μιας εφαρμογής δίνοντας την εικόνα ενός ενιαίου συστήματος ακόμα κι αν κάποιοι κόμβοι του συστήματος δεν χρησιμοποιούνται. Παράλληλη ή κατανεμημένη

επεξεργασία μπορούμε να έχουμε σε ομάδες επεξεργαστών (Cluster Computing) και σε πλέγματα επεξεργαστών (Grid Computing).

Ένα cluster είναι ένα σύνολο/σύμπλεγμα συνδεδεμένων υπολογιστών που συνεργάζονται ώστε να δίνουν την εντύπωση ότι λειτουργούν ως μεμονωμένοι υπολογιστές. Τα συστατικά ενός cluster είναι συνδεδεμένα με ένα γρήγορο τοπικό δίκτυο και συνήθως αναπτύσσονται για να βελτιώσουν την ταχύτητα και την απόδοση συγκριτικά με έναν υπολογιστή. Συχνά, οι συστάδες υπολογιστών χρησιμοποιούνται για υπολογιστικούς σκοπούς παρά για τον χειρισμό διεργασιών εισόδου/εξόδου όπως για παράδειγμα σε βάσεις δεδομένων ή υπηρεσίες ιστού (web services) και οι κόμβοι τους πιθανότατα να είναι ομογενείς. Τα Grids μπορούν να θεωρηθούν ως συστάδες υπολογιστών με την διαφορά ότι τα συστατικά στοιχεία του πλέγματος παρουσιάζουν ετερογένεια, είναι γεωγραφικά κατανεμημένα και μερικές φορές η διαχείριση τους παρέχεται από διαφορετικούς οργανισμούς.

2.2 Το πλέγμα (GRID)

Το πλέγμα ή όπως είναι ευρέως γνωστό με τον αγγλικό όρο Grid βρίσκει εφαρμογή σε ερευνητικά προγράμματα, τα οποία έχουν αναδείξει τα πλεονεκτήματα της δομημένης και οργανωμένης κοινής χρήσης κάποιων διαθέσιμων πόρων. Αυτά τα πλεονέκτημα λειτουργούν ως κινητήριος δύναμη και προάγγελοι εφαρμογών του πλέγματος στη βιομηχανία, την ακαδημαϊκή κοινότητα και ευρύτερα την κοινωνία. Διάφοροι βιομηχανικοί τομείς μεταξύ των οποίων η φαρμακοβιομηχανία, οι τηλεπικοινωνίες, οι οικονομικές υπηρεσίες (e-banking, e-commerce) και η πληροφορική επωφελούνται άμεσα από τις υψηλές υπολογιστικές δυνατότητες που προσφέρει, χωρίς να παραβλέπουμε το γεγονός ότι αποτελούν έναν καλύτερο τρόπο οργάνωσης και διαχείρισης των πληροφοριών. Η ερευνητική και ακαδημαϊκή κοινότητα από την πλευρά της χρησιμοποίησε αρχικά το πλέγμα για την υλοποίηση εφαρμογών πολύ υψηλού υπολογιστικού κόστους όπως είναι οι βιοϊατρικές επιστήμες (πχ στην ανάλυση του ανθρώπινου γονιδιώματος).

Με τον καιρό οι συμπράξεις και οι συνεργασίες που πραγματοποιήθηκαν στους διάφορους τομείς εφαρμογής στοχεύοντας στην επίλυση προβλημάτων που κάποτε θεωρούταν αδύνατο να αντιμετωπιστούν σε συνδυασμό με την εξέλιξη της τεχνολογίας μετέβαλλαν τον αρχικό ορισμό του πλέγματος. Σύμφωνα με τον Ian Foster [2], που θεωρείται από τους εμπνευστές των τεχνολογιών πλέγματος, ο όρος Πλέγμα εμφανίστηκε στα μέσα της δεκαετίας του '90 για να προσδιορίσει μια κατανεμημένη υπολογιστική υποδομή προσανατολισμένη στην ανώτερη επιστήμη και μηχανική. Σήμερα όταν χρησιμοποιούμε τον όρο δεν αναφερόμαστε μόνο στην υποδομή υλικού και λογισμικού που παρέχει αξιοπιστία και χαμηλού κόστους πρόσβαση σε υπολογιστικές δυνατότητες υψηλών απαιτήσεων, αλλά δίνουμε μια ευρύτερη ερμηνεία που περιλαμβάνει κοινωνικά ζητήματα και πολιτικές στο πλαίσιο των εικονικών οργανισμών (Virtual Organizations). Ένας εικονικός

οργανισμός είναι ένα δυναμικό σύνολο ιδιωτών ή/και ινστιτούτων που διέπονται από καθορισμένους κανόνες και συνθήκες. Το σύνολο των εικονικών οργανισμών μοιράζονται κάποια κοινά χαρακτηριστικά συμπεριλαμβανομένων κοινών θεωρήσεων και απαιτήσεων, αλλά ίσως διαφέρουν στο μέγεθος, το πεδίο εφαρμογών, την διάρκεια και την δομή τους. Τίθεται, λοιπόν, το ερώτημα ποια είναι τα χαρακτηριστικά εκείνα που ανταποκρίνονται στον ορισμό και την εξέλιξη του.

Το πλέγμα, λοιπόν, αποτελεί μια δομή με οργανωμένη, κοινή χρήση πόρων για την επίλυση προβλημάτων στο πλαίσιο δυναμικών δομών που σχετίζονται από τη συνάθροιση ή την συνεργασία πολλαπλών οργανισμών που χαρακτηρίζονται από τρία βασικά, κοινά χαρακτηριστικά[5]:

✓συνάθροιση πόρων που δεν υπόκεινται σε κεντρικό έλεγχο

✓τη χρήση προκαθορισμένων διεπαφών μεταξύ των σημείων του πλέγματος

✓την παροχή υπηρεσιών υψηλής ποιότητας

Η βασική ιδιότητα του είναι ότι μπορεί να αναπαράγει χρήσιμες ποσότητες υπολογιστικής ισχύς στους χρήστες του. Η αρχιτεκτονική του ακολουθεί το πρότυπο των επιπέδων διακρίνοντας το επίπεδο δικτύου, το επίπεδο πόρων (υπολογιστές, ηλεκτρονικοί κατάλογοι δεδομένων, αισθητήρες) που συνδέονται στο δίκτυο, το επίπεδο λογισμικού middleware (εργαλεία και στοιχεία για την επικοινωνία με το πλέγμα) και, τέλος, το επίπεδο εφαρμογών. Τα διαφορετικά μοντέλα και προεκτάσεις του περιγράφονται από τις Πέντε Βασικές Ιδέες Πίσω από το Πλέγμα και αφορούν τα εξής:

✓Διαμοίραση πόρων (εικονικοί οργανισμοί)

✓Ασφάλεια πρόσβασης (πολιτικές πρόσβασης, αυθεντικοποίηση και ταυτοποίηση)

✓Χρήση πόρων

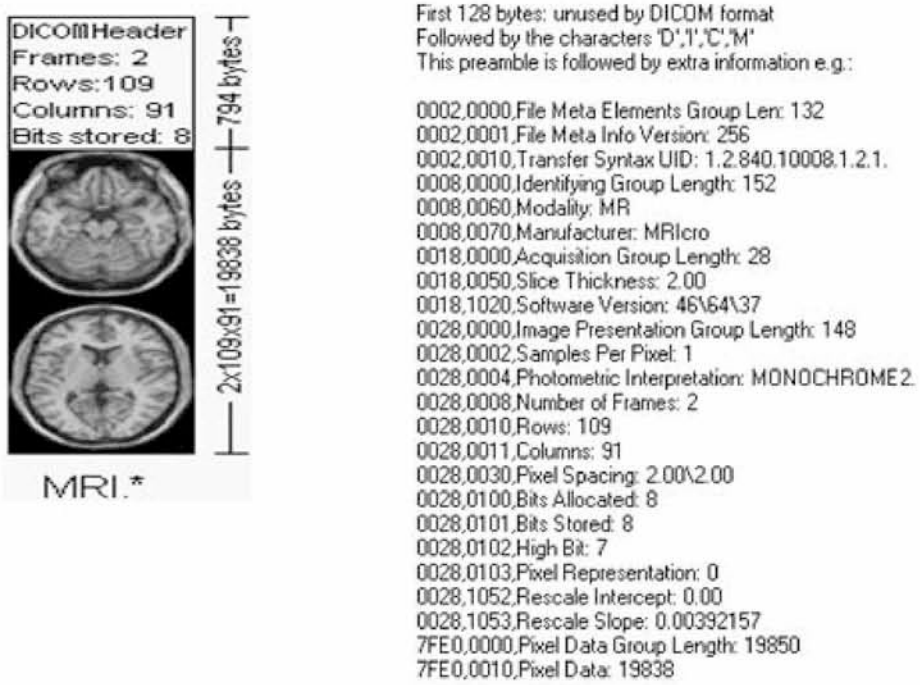
✓Το τέλος των αποστάσεων (ανάπτυξη δικτύων υψηλών ταχυτήτων)

✓Ανοιχτά πρότυπα (χρήση των ίδιων πρωτοκόλλων)

Όλα τα παραπάνω έχουν ως απώτερο σκοπό ο τελικός χρήστης να βλέπει το πλέγμα ως ένα μεγάλο υπερ-υπολογιστή, ακριβώς όπως ένας χρήστης του διαδικτύου βλέπει όλο το περιεχόμενό του ενοποιημένο και αυτά τα χαρακτηριστικά είναι που θα επωφεληθούμε σε αυτή την εργασία.

2.3 Το πρότυπο DICOM

Αναφέρθηκε στην εισαγωγή ότι η Ιατρική Πληροφορική χρησιμοποιεί διάφορα πρότυπα για την ανάλυση και ανάπτυξη συστημάτων και ένα από αυτά είναι το πρότυπο DICOM . Το πρότυπο DICOM (Digital Imaging and Communications in Medicine) δημιουργήθηκε από την NEMA (National Electric Manufacturers Associations) για να καλύψει την περιοχή της διαχείρισης, αποθήκευσης, εκτύπωσης και μετάδοσης των ιατρικών εικόνων. Αποτελεί το πιο κοινό πρότυπο για την λήψη τομών από ένα νοσοκομείο, και η ανταλλαγή δεδομένων τύπου DICOM απαιτεί την ύπαρξη ενός DICOM Viewer και στις δύο πλευρές (αποστολέας – παραλήπτης). Ένα απλό αρχείο τύπου DICOM περιέχει μια επικεφαλίδα που περιλαμβάνει πληροφορίες για τον ασθενή, τον τύπο της εικόνας/τομής, τις διαστάσεις της εικόνας, σχόλια, κ.ο.κ και τα δεδομένα της εικόνας. Ένα αρχείο εικόνας DICOM θα μπορούσε να έχει τη μορφή που παρουσιάζεται στην εικόνα 3 (αριστερά) ενώ το πλήθος των bytes της επικεφαλίδας ποικίλει ανάλογα με το ποσό πληροφορίας που αποθηκεύει. Μια πιο λεπτομερής παρουσίαση των στοιχείων μιας επικεφαλίδας φαίνεται στην εικόνα 3 (δεξιά). Πρακτικά, οι DICOM Viewers δεν ελέγχουν για την ύπαρξη όλων αυτών των στοιχείων αλλά χρησιμοποιούν μόνο τις απαραίτητες για αυτούς πληροφορίες, όπως για παράδειγμα το μέγεθος της εικόνας ή αν έχει γίνει συμπίεση με ή χωρίς απώλειες.



Εικόνα 3: Αριστερα απεικονίζεται θεωρητικά ένα αρχείο DICOM, δεξιά είναι τα στοιχεία της εοικεφαλίδας του.

Η παραπάνω λίστα παρέχει πλήθος πληροφοριών αποθηκευμένες σε αντίστοιχες κωδικοποιημένες μεταβλητές. Επίσης, το πρότυπο δίνει τη δυνατότητα στους χρήστες να δημιουργήσουν τις δικές τους ετικέτες (private tags) που δεν

αναγνωρίζονται, όμως, από κάθε συσκευή που διαβάζει αρχεία DICOM. Τα στοιχεία που αποθηκεύονται στην επικεφαλίδα είναι σε πολλές υλοποιήσεις ομαδοποιημένες ανάλογα με την οντότητα που καλούνται να περιγράψουν. Βέβαια, κάποια από τα χαρακτηριστικά της επικεφαλίδας θεωρούνται απαραίτητα και η παρουσία τους επιβεβλημένη, όπως X/Y Pixel Spacing, Slice Thickness, Slice Location, Patient name, Transfer Syntax UID. Από το σύνολο των χαρακτηριστικών θα επιλεγούν μόνο όσα θεωρούνται πιο σημαντικά και απαραίτητα σε αυτή την εργασία. Τα πεδία του πίνακα DICOM προέρχονται από την επικεφαλίδα του αρχείου, οπότε η εξαγωγή τους απαιτεί την χρήση εφαρμογών (APIs) υλοποιημένες με διάφορους τρόπους όπως είναι για παράδειγμα το DCMTK ώστε να παρέχεται ένα επίπεδο λειτουργικότητας ή μπορεί να πραγματοποιηθεί μέσω κώδικα PHP. Το DCMTK είναι ένα πακέτο διαχείρισης δεδομένων DICOM που αποτελείται από τον πηγαίο κώδικα, την τεκμηρίωσή του, και οδηγίες εγκατάστασης. Το εργαλείο αυτό περιλαμβάνει κάποια υπό-πακέτα καθένα από τα οποία ανταποκρίνεται σε διαφορετικές ανάγκες. Για παράδειγμα, το πακέτο dcmnet περιέχει μια συλλογή λειτουργιών που υλοποιούν τον τρόπο επικοινωνίας του πρωτοκόλλου DICOM όπως το echoscu, movescu, storescu ή το πακέτο dscmdata που έχει την λειτουργία dcm2xml που μετατρέπει ένα αρχείο DICOM σε XML μορφή. Πιο συγκεκριμένα, για την εξαγωγή των στοιχείων της επικεφαλίδας χρησιμοποιείται το πακέτο php-DICOM ή File_DICOM όπως αναφέρεται στον ιστότοπο της PEAR. Το File_DICOM επιτρέπει την ανάγνωση και μετατροπή των DICOM αρχείων αλλά δεν υποστηρίζει την ανταλλαγή ή μεταφορά δεδομένων. Για παράδειγμα, αφού δημιουργηθεί η μεταβλητή $dicom που περιέχει τα στοιχεία του αρχείου, μέσω php καλείται η συνάρτηση getValue για να διαβάσει την τιμή της μεταβλητής $modality . Κάθε μεταβλητή DICOM αποτελείται από τέσσερα στοιχεία:

- ✓ Μια ετικέτα της μορφής (ομάδα, στοιχείο)-> (XXXX,XXXX)
- ✓ Την αναπαράσταση τιμής (Value Representation (VR)) που περιγράφει τον τύπο των δεδομένων
- ✓ Το μήκος της τιμής
- ✓ Την τιμή του πεδίου, δηλαδή την ζητούμενη τιμή

Σε αυτή την εντολή παρατηρούμε ότι η ανάκτηση της τιμής γίνεται χρησιμοποιώντας την ετικέτα της που την καθορίζει μοναδικά. Επιλέγουμε, λοιπόν, τα στοιχεία που θέλουμε να υποθηκεύσουμε στην βάση δεδομένων εξάγοντας τα με παρόμοιο τρόπο. Οι τιμές των ετικετών προέρχονται από εδώ: http://star.pst.qub.ac.uk/idl/DICOM_Attributes.html.

```
$modality=$dicom->getValue(0x0008, 0x0060);
```

Ακόμη, επειδή ο τύπος DICOM είναι αναγνωρίσιμος μόνο από ειδικές συσκευές απαιτείται η μετατροπή τους σε JPEG ώστε να είναι εφικτή η παρουσίασή τους στο χρήστη. Για αυτό το λόγο χρησιμοποιείται το πακέτο Image_Transform. Το πακέτο αυτό περιλαμβάνει κάποιους οδηγούς για την μετατροπή του τύπου του αρχείου και περιορισμένη επεξεργασία του.

2.4 Επισκόπηση Βιβλιογραφίας

Η έρευνα στις επιστήμες ζωής στηρίζεται σε δύο βασικά εργαλεία, τις Βάσεις Δεδομένων και τις Βάσεις Γνώσεις. Από την εκμετάλλευσή τους για την εξαγωγή και ανάλυσή πληροφοριών εξαρτώνται άμεσα η ποιότητα και η απόδοση της έρευνας. Στις μέρες μας, οι εφαρμογές του πλέγματος δεν περιορίζονται στα όρια των εργαστηρίων, και πέρα από το "υπολογιστικό πλέγμα" (computational grid) διακρίνουμε το πλέγμα δεδομένων (data grid ή knowledge grid) το οποίο λειτουργεί ως ένα εργαλείο έξυπνης χρήσης για τη δημιουργία γνώσης. Σύμφωνα με τους [4], οι σύγχρονες τεχνικές πλέγματος βρίσκουν εφαρμογή στους ακόλουθους τομείς:

1.υπολογιστική γονιδιωματική και υπολογιστική πρωτεωμική [21]

2.συστημική βιολογία και ενσωμάτωση βιολογικών πληροφοριών [22]

3.αποθήκευση βιοϊατρικών δεδομένων

4.ανάκτηση κατανεμημένων βιοϊατρικών δεδομένων

5.βιοϊατρική μοντελοποίηση και προσομοίωση [23]

6.βιοϊατρική επεξεργασία εικόνας και προσομοίωση

7.διαχείριση/ενσωμάτωση κατανεμημένων ιατρικών βάσεων δεδομένων

8.εξόρυξη και οπτικοποίηση των βιοϊατρικών δεδομένων [24]

9.εφαρμογές τηλέ-συστημάτων διάγνωσης, πρόγνωσης και θεραπείας

10.μηχανογραφημένη επιδημιολογία

11.φαρμακευτικές και κλινικές δοκιμές (Cts) [25]

12.κοινωνική περίθαλψη και δίκτυα συνεργασίας

Παρόλαυτά, σύμφωνα με τους ίδιους ερευνητές ενώ υπάρχουν εφαρμογές και υλοποιήσεις στους παραπάνω τομείς, ακόμη παραμένουν ανοικτά κάποια ζητήματα όπως ηθικά θέματα, θέματα νομιμότητας και αξιοπιστίας, ασφάλειας και διαλειτουργικότητας τα οποία αφορούν τη περιοχή του BioGrid γενικότερα και παρουσιάζονται ακολούθως.

Είναι γεγονός ότι οι εφαρμογές, για παράδειγμα, στην γονιδιωματική οδηγούν σε ένα επίπεδο πέρα από το μοριακό. Εφόσον το γονιδίωμα ενός ατόμου είναι μοναδικό, τίθεται το ερώτημα πως και με ποια συγκεκριμένα και σαφή μέτρα θα

εξασφαλιστεί η ασφάλεια των πληροφοριών που παρέχει ώστε να μείνουν μακριά από οποιαδήποτε κατάχρηση. Αυτός ο περιορισμός δεν ισχύει, όμως, μόνο σε αυτό το πεδίο εφαρμογής αλλά και στην διαχείριση των βιοϊατρικών δεδομένων που θεωρούνται απόρρητα. Άρα, ποια είναι η αρχή εκείνη που θα διασφαλίζει την προστασία των δεδομένων αυτών και θα εξασφαλίζει την αξιοπιστία των ερευνών; Αυτό το κενό έρχονται να καλύψουν οι Εικονικοί Οργανισμοί (Virtual Organizations-VO) με την λειτουργία τους. Το θέμα της ασφάλειας έχει απασχολήσει ιδιαίτερα τους ερευνητές καθώς ένας μεγάλος αριθμός βιοϊατρικών δεδομένων κατανέμονται παγκοσμίως σε "αποθήκες" και πλήθος εφαρμογών βασίζονται στην κοινή τους χρήση. Στα βασικά έργα του πλέγματος χρησιμοποιείται η υποδομή ασφάλειας πλέγματος (Grid Security Infrastructure – GSI) που κάνει ασύμμετρη κρυπτογράφηση και η πρόσβαση ελέγχεται μέσω λιστών ελέγχου πρόσβασης (Access Control Lists - ACL). Τέλος, τα προβλήματα διαλειτουργικότητας περιορίζονται συνεχώς, καθώς όλο και περισσότερα πρωτόκολλα βρίσκουν έναν τρόπο επικοινωνίας με το λογισμικό (middleware) του πλέγματος για την υποστήριξη εφαρμογών όπως είναι η σύνδεση του πρωτοκόλλου DICOM με το GRID που εφαρμόστηκε στην υποδομή του MediGrid.

Σε αυτή την ενότητα, αφήνοντας πίσω τους όποιους περιορισμούς και ενδοιασμούς, κάποιοι από τους οποίους αποτελούν ήδη παρελθόν, προχωράμε στην επισκόπηση κάποιων ήδη υπαρχόντων εφαρμογών. Στο άρθρο [4] γίνεται αναφορά σε πλήθος εφαρμογών για την ανάπτυξη συστημάτων που μπορούν να λειτουργήσουν ως πλαίσια για την έρευνα και την πρακτική στις επιστήμες ζωής όπως για παράδειγμα η επιτάχυνση πολλαπλών εκτελέσεων μιας εφαρμογής που προσομοιώνει το πιθανό δυναμικό πολλαπλασιασμού δράσης των καρδιακών ιστών. Η συγκεκριμένη υλοποίηση, για παράδειγμα, χρησιμοποιεί μια συστάδα υπολογιστών (cluster) που εκτελεί παράλληλα τις προσομοιώσεις των περιπτώσεων μελέτης της καρδιακής λειτουργίας. Ακόμη, στα πλαίσια του EGEE πραγματοποιήθηκαν αρκετές εφαρμογές βασισμένες στις τεχνολογίες πλέγματος και ιστού. Υπάρχει, όμως, ένα μεγάλο πλήθος εφαρμογών. Καθεμιά είναι προσανατολισμένη σε διαφορετικούς στόχους αλλά όλες έχουν ένα κοινό χαρακτηριστικό: χρησιμοποιεί το πλέγμα ως βασικό συστατικό υλοποίησης και προσπαθεί να ενσωματώσει όσο το δυνατό περισσότερα στοιχεία στην υποδομή. Στη συνέχεια αυτού του κεφαλαίου, παρουσιάζονται πέντε εφαρμογές διαφορετικές μεταξύ τους γύρω από την περιοχή των επιστημών ζωής στοχεύοντας στην δημιουργία μιας συνολικής εικόνας για το εύρος αξιοποίησης της νέας αυτής τεχνολογίας καθώς επίσης, και να καταδείξει έναν αρκετά μεγάλο αριθμό προεκτάσεων.

2.4.1 Το σύστημα DataGrid – Εφαρμογή περίπτωσης [3]

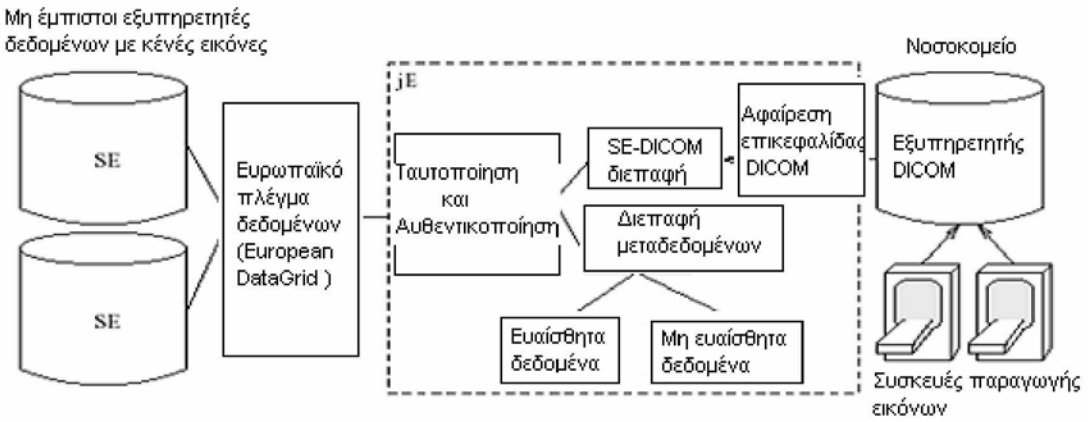
Ο όρος DataGrid δεν αφορά αποκλειστικά και μόνο μια συγκεκριμένη εφαρμογή αλλά υπό την ευρύτερη έννοια του όρου αναφέρεται στο πλήθος των εφαρμογών που ασχολούνται με την αποθήκευση και ανάκτηση δεδομένων σε περιβάλλον

πλέγματος. Αυτό σημαίνει ότι υπάρχει μια βάση δεδομένων στην οποία είναι αποθηκευμένα τα δεδομένα της εφαρμογής και στη περίπτωση μας τα ιατρικά δεδομένα. Τα στοιχεία του ασθενή, οι ιατρικές του εικόνες με τα μεταδεδομένα τους και οι εξετάσεις του θα μπορούσαν να είναι κάποια από αυτά. Η επεξεργασία των δεδομένων στο πλέγμα εγείρει προβληματισμούς αναφορικά με την εμπιστευτικότητα και την υποδομή ασφάλειας του όπως άλλωστε αναφέρθηκαν σε προηγούμενο κεφάλαιο, αλλά αυτά αντιμετωπίζονται με έλεγχο της πρόσβασης (και) σε αντίγραφα των δεδομένων και μέσω της κρυπτογράφησης αυτών. Οι συγγραφείς του άρθρου [3] διακρίνουν τρεις ομάδες χρηστών: ασθενείς, γιατροί και ερευνητές με την κάθε ομάδα να έχει διαφορετικά δικαιώματα πρόσβασης και να λειτουργεί με διαφορετικό τρόπο στις ποικίλες σχετικές με ιατρικά δεδομένα εφαρμογές.

Πριν προχωρήσουμε στην περιγραφή της αρχιτεκτονικής του συστήματος, κρίνεται σκόπιμο να αναφέρουμε κάποιες απαιτήσεις που οφείλει να πληρεί ένα Πληροφοριακό Σύστημα Υγείας (ΠΣΥ) εφόσον το DataGrid παρουσιάζει παρόμοιες λειτουργίες με ένα ΠΣΥ. Πρώτα από όλα, θα πρέπει να μπορεί να αποθηκεύει τις πρωτότυπες και επεξεργασμένες εικόνες και τα συσχετισμένα με αυτές δεδομένα, να παρέχει υπηρεσίες κρυπτογράφησης του περιεχομένου της εικόνας, να κρατά εγγραφές των επεξεργασμένων εικόνων από την αρχική και το αντίστροφο, να παρακολουθεί τους αλγορίθμους για την παραγωγή μιας δοσμένης εικόνας, να επιτρέπει την καταγραφή των δεδομένων πρόσβασης και, τέλος, να επικοινωνεί με το λογισμικό DataGrid. Σε αυτή την εφαρμογή, ερευνάται η δημιουργία μιας εξειδικευμένης αποθηκευτικής μονάδας που ονομάζεται jE και συνδέεται με το Grid. Υπό αυτό, λοιπόν, το πρίσμα η μονάδα jE λειτουργεί ως μια διεπαφή μεταξύ των εξυπηρετητών DICOM του νοσοκομείου με το λογισμικό DataGrid. Θεωρείται ότι ο DICOM εξυπηρετητής συμπεριφέρεται ως ένα τεράστιο αποθηκευτικό σύστημα (Mass Storage System - MSS) αλλά στην πραγματικότητα είναι ένα περιορισμένο MSS καθώς νέες εικόνες ή άλλου είδους δεδομένα που παράγονται στο πλέγμα δεν μπορούν να αποθηκευτούν σε αυτούς τους εξυπηρετητές. Οπότε, τα αποτελέσματα οποιασδήποτε επεξεργασίας θα πρέπει να αποθηκεύονται στα αποδεκτά αποθηκευτικά στοιχεία του πλέγματος (Storage Element - SE). Η μονάδα αυτή θα πρέπει να είναι σε θέση να αποθηκεύσει μεταδεδομένα και να ορίζει μια σύνδεση ανάμεσα σε εικόνες ή/και τα μεταδεδομένα ανεξαρτήτως σε ποια από τις δύο προηγούμενες θέσεις είναι αποθηκευμένες. Επίσης, είναι αναγκαίο να μπορεί πάντα ο χρήστης να δει την προέλευση των δεδομένων (ποιο νοσοκομείο παρήγαγε την εικόνα) αλλά και την σύνδεση με τους αλγορίθμους που χρησιμοποιούνται. (Η κατανεμημένη φύση του πλέγματος δεν επιτρέπει μια «κεντραρισμένη» αρχιτεκτονική jE που θα αποθηκεύει τα δεδομένα κάθε ιατρικής εικόνας.)

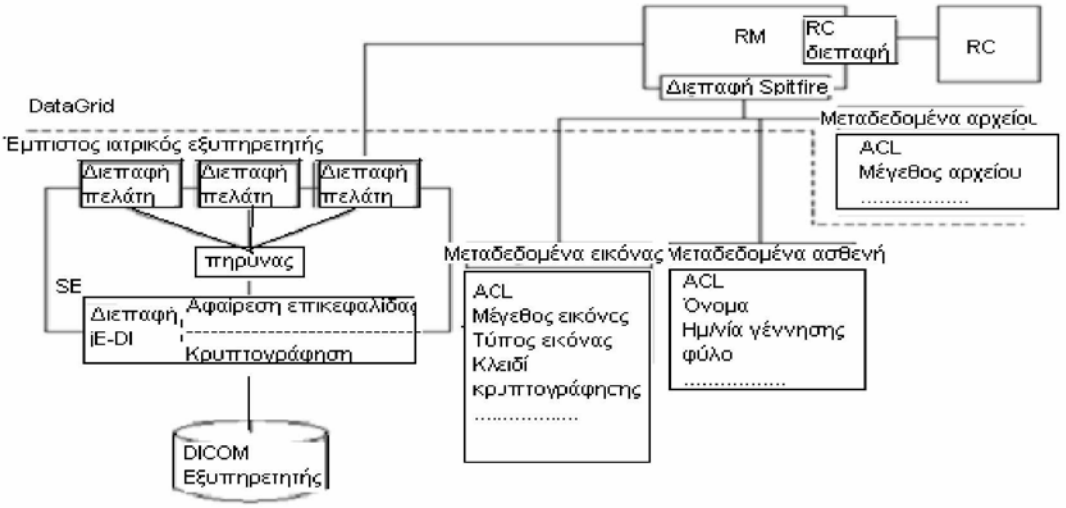
Νωρίτερα έγινε μια πολύ σύντομη αναφορά στην μονάδα jE. Σε αυτή την παράγραφο, λοιπόν, παρουσιάζεται η συνολική αρχιτεκτονική του συστήματος. Η

μονάδα jE βρίσκεται όπως φαίνεται από το ακόλουθο σχήμα (Εικόνα 4) μεταξύ της υποδομής του DataGrid που παρέχει πρόσβαση στα SE του πλέγματος και της βάσης δεδομένων του νοσοκομείου. Η αποστολή μιας εικόνας στο πλέγμα μαζί με την επικεφαλίδα της που αποτελείται από πλήθος ευαίσθητων δεδομένων είναι επισφαλής, για αυτό το λόγο η jE την αφαιρεί και την αποθηκεύει ως μεταδεδομένα στην βάση της κρατώντας πάντοτε μια σύνδεση (εγγραφή) ώστε να μπορεί να είναι εφικτή η αναζήτηση τους.



Εικόνα 4: Απλοποιημένη δομή του DataGrid.

Η συγκεκριμένη αρχιτεκτονική διαχείρισης των δεδομένων από το σύστημα MSS δίνει τη δυνατότητα πρόσβασης από κάποιους χρήστες μέσω μιας διεπαφής που συνδέεται με έναν διαχειριστή αντιγράφων (Replica Manager – RM). Ο RM με τη σειρά του επικοινωνεί με έναν κατάλογο αντιγράφων (Replica Catalog – RC) που κρατά την αντιστοιχία των φυσικών ονομάτων των αρχείων με τα λογικά, και, τέλος, μέσω μιας διεπαφής αποκτά πρόσβαση στα μεταδεδομένα της εικόνας. Αυτό σημαίνει ότι δεν υπάρχει η δυνατότητα ανεξάρτητης πρόσβασης τόσο στα μεταδεδομένα της εικόνας όσο και στο περιεχόμενο των αρχείων και τα μεταδεδομένα του ασθενή αφού ένας χρήστης μπορεί να έχει εξουσιοδότηση πρόσβασης μόνο στα μεταδεδομένα της εικόνας αλλά όχι του ασθενή. Έτσι, οι Montagnat et al [3], προχώρησαν σε τροποποίηση της αρχικής αρχιτεκτονικής τους, όπως παρουσιάζεται στην εικόνα 5. Ο εξυπηρετητής DICOM στέλνει τα δεδομένα στην μονάδα jE-DI (που μεταφράζει τις απαιτήσεις πηρύνα SE σε απαιτήσεις DICOM) όπου αφαιρείται η επικεφαλίδα και γίνεται κρυπτογράφηση της εικόνας έτσι ώστε να μην μπορεί ούτε ο διαχειριστής του συστήματος να δει το περιεχόμενό της. Στη συνέχεια, ο RM επιστρέφει τα στοιχεία που ζητήθηκαν κάνοντας διάκριση μεταξύ των μεταδεδομένων της εικόνας, του ασθενή και του αρχείου και άρα υλοποιεί την απαίτηση για εμπιστευτικότητα των δεδομένων.



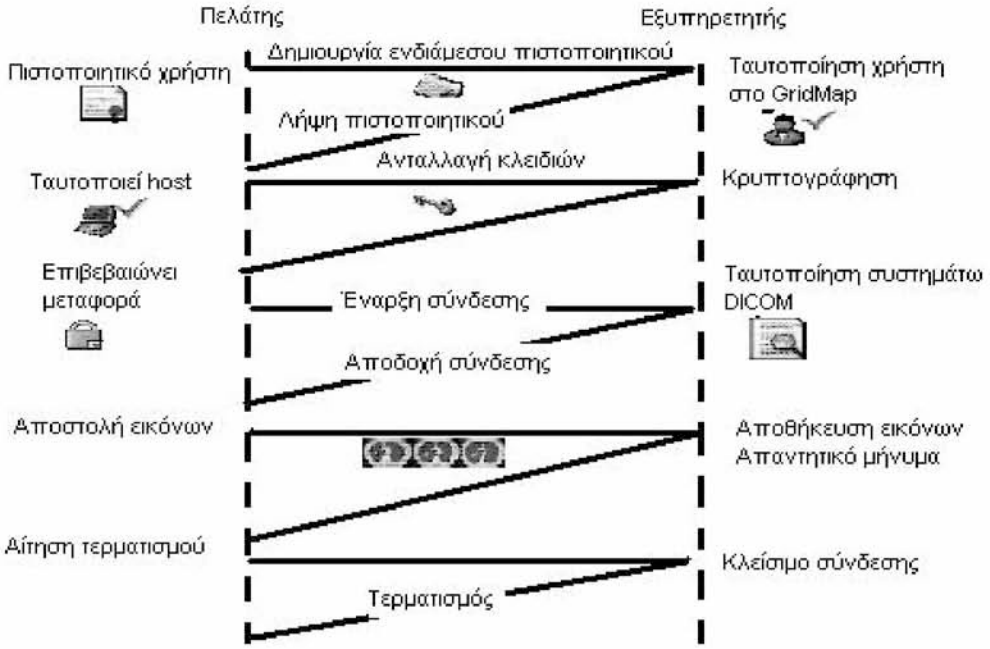
Εικόνα 5: Η συνολική αρχιτεκτονική του DataGrid.

Η εφαρμογή αυτή δεν είναι ακόμα ολοκληρωμένη αλλά στοχεύει να προσθέσει και να βελτιώσει στοιχεία στην αρχιτεκτονική λογισμικού. Για παράδειγμα, η μονάδα jE αποτελεί ένα κατανεμημένο στοιχείο οπότε κατανεμημένα ερωτήματα από διαφορετικούς κόμβους θα μπορούσαν να εξυπηρετούνται από μια σύνδεση jE – jE ή μια διεπαφή χρήστη (jE-user) που θα επέτρεπε την άμεση υποβολή ερωτημάτων.

2.4.2 Το σύστημα Grid-DICOM – Εφαρμογή περίπτωσης[1]

Η σύγχρονη κλινική πρακτική έχει συνδεθεί άμεσα με την χρήση ραδιολογικών εξετάσεων για ορθότερη και ασφαλέστερη διάγνωση. Εντούτοις, πολλές φορές εφαρμόζονται τεχνικές επεξεργασίας και ανάλυσης των αρχικών εικόνων ώστε να εξαχθούν χαρακτηριστικά και πληροφορίες που ενισχύουν ή/και στηρίζουν την διάγνωση. Γνωρίζουμε σήμερα ότι οι εικόνες διαχειρίζονται στις περισσότερες των περιπτώσεων από το σύστημα αρχειοθέτησης PACS το οποίο προσφέρει χρήσιμες αλλά περιορισμένες δυνατότητες, καθώς επιτρέπει την ανταλλαγή δεδομένων μόνο από και προς συστήματα PACS. Θεωρήθηκε, λοιπόν, ότι η υποδομή πλέγματος είναι η πλέον κατάλληλη για να υποστηρίξει μια εφαρμογή που επεξεργάζεται και αποθηκεύει το συγκεκριμένο τύπο δεδομένων. Έτσι, σχεδιάζοντας τον τρόπο με τον οποίο θα μπορούσαν να αξιοποιήσουν τις δυνατότητες του πλέγματος οι ερευνητές του γερμανικού έργου εφαρμογής του grid (MediGrid) για τις επιστήμες ζωής ήρθαν αντιμέτωποι με ένα πρόβλημα: την προσαρμογή του προτύπου DICOM με τα στοιχεία και τις ιδιότητες του πλέγματος. Μερικά από αυτά τα συστατικά αναφέρονται από τους [1] όπως είναι ή θα έπρεπε να είναι για να αποτελούν αποτέλεσμα αυτής της προσαρμογής, αποτελώντας ουσιαστικά και τους στόχους προς υλοποίηση. Αυτά τα στοιχεία είναι η διεπαφή ασφαλείας (Grid Security Interface – GSI) χρησιμοποιώντας προσωρινά έγκυρα (ενδιάμεσα-proxy) πιστοποιητικά ταυτοποιώντας την ταυτότητα του χρήστη, το πρωτόκολλο μεταφοράς αρχείων (Grid-FTP) με προφανή ρόλο, ένα σύστημα διαχείρισης

δεδομένων (Storage Resource Broker – SRB) για την αποθήκευση, ανάκτηση και δημιουργία αντιγράφων στο πλέγμα, η υποδομή υπηρεσιών του πλέγματος (DICOM Grid Interface Service – DGIS) που αποτελεί ουσιαστικά τον τρόπο επικοινωνίας του grid με τις συσκευές DICOM και, τέλος, τον Διαχειριστή των Ιατρικών Δεδομένων (Medical Data Manager – MDM) που λειτουργεί ως πύλη μεταξύ των συσκευών DICOM και του Διαχειριστή Αποθηκευτικών Πηγών του gLite middleware (Storage Resource Manager–SRM).



Εικόνα 6: Η επικοινωνία πελάτη-εξυπηρετητή αποτελείται από ανταλλαγή στοιχείων ταυτοποίησης όπως πιστοποιητικά και μηνυμάτων αποδοχής ή απόρριψης πριν τη μεταφορά των δεδομένων

Είναι γνωστό ότι το πρωτόκολλο DICOM καθορίζει τον τρόπο με τον οποίο γίνεται η ανταλλαγή δεδομένων και μηνυμάτων. Σε συνδυασμό με τη χρήση του πρωτοκόλλου TCP/IP παρέχει ένα επίπεδο ασφάλειας κάνοντας χρήση των TSL/SSL. Στην περίπτωση του πλέγματος, η επικοινωνία μεταξύ των κόμβων απαιτεί κάτι περισσότερο, γι' αυτό ξεκίνησε η προσαρμογή του DICOM για κρυπτογράφηση, αμοιβαία αυθεντικοποίηση, ταυτοποίηση και διανομή των δικαιωμάτων των χρηστών. Η παραπάνω μετατροπή πραγματοποιήθηκε χρησιμοποιώντας την υλοποίηση του DICOM σε Java, καθώς αυτό καθιστά την επικοινωνία με το GSI ακόμη πιο εύκολη. Μετατράπηκαν, λοιπόν, οι κλάσεις του επιπέδου δικτύου ώστε να εξασφαλιστεί ένα ασφαλές περιεχόμενο για το GSI πριν από οποιαδήποτε προσπάθεια συσχέτισης με το DICOM. Η τεχνική που χρησιμοποιήθηκε ήταν μέσω της χρήσης πιστοποιητικών που παρέχονται από μια Αρχή Πιστοποίησης (Certificate Authority - CA). Έχοντας ως βάση τις παραπάνω τροποποιήσεις του Grid-DICOM, αναπτύχθηκαν υλοποιήσεις όπως η αποστολή (GridDcmSend) και λήψη εικόνων (GridDcmRecieve). Ένα παράδειγμα του τρόπου λειτουργίας του φαίνεται στην

Εικόνα 2, όπου παρουσιάζεται η ακολουθία των βημάτων για την αποθήκευση μιας σειράς εικόνων από τον CT scanner σε μια βάση δεδομένων.
Η υλοποίηση, όμως, του πρωτοκόλλου βρίσκει μόνο περιορισμένη χρήση και δεν δίνει τη δυνατότητα επικοινωνίας με κάθε «νόμιμη» συσκευή DICOM, όπως στην περίπτωση χρήσης του συστήματος PACS. Αυτή η ασυμβατότητα θα καθιστούσε αδύνατη την ανάπτυξη ιατρικών και βιοϊατρικών εφαρμογών σε περιβάλλον πλέγματος, αλλά η χρήση ενός δρομολογητή (router) που θα λειτουργεί ως ενδιάμεσος σταθμός έδωσε τη λύση. Ο ρόλος του είναι να «ακούει» τα μηνύματα και από τις δύο πλευρές και να μεταφράζει το περιεχόμενό τους. Οι πιο σημαντικές, λοιπόν, κλάσεις που χρησιμοποιούνται από τις υπηρεσίες DICOM είναι οι εξής:

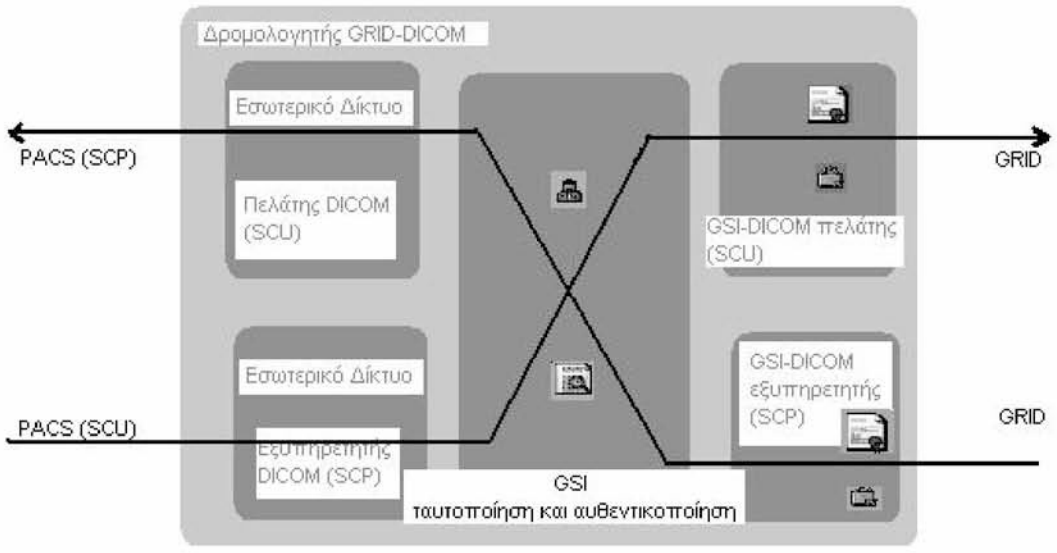
1.C-ECHO (Verification): κάνει έλεγχο συμβατότητας

2.C-STORE: αποθήκευση και αίτηση για λήψη εικόνων

3.C-FIND: ερωτήματα βάση κάποιων χαρακτηριστικών της εικόνας, όπως ένας αριθμός ID

4.C-GET και C-MOVE: Στην πρώτη περίπτωση δεν χρειάζεται να ανοίξει κάποια θύρα στο τείχος προστασίας του client σε αντίθεση με το δεύτερο όπου ο χρήστης ζητά από μια τρίτη οντότητα να μεταφέρει μια εικόνα σε άλλη συσκευή. Τότε ο εξυπηρετητής ανοίγει μια νέα σύνδεση με το σύστημα προορισμού και ζητά πρόσβαση.

Πέρα από την υλοποίηση του Grid-DICOM πραγματοποιήθηκε ένα ακόμα βήμα. Σχεδιάστηκε η



εφαρμογή JMX (Java Management eXtentions) για την ευέλικτη διαχείριση των

Εικόνα 7: Παρουσιάζεται ο τρόπος λειτουργίας του δρομολογητή ώστε να επιτυγχάνονται η αυθεντικοποίηση και η ταυτοποίηση

στοιχείων της εφαρμογής σε ένα κατανεμημένο δίκτυο, όπως είναι τα clusters και τα grids. Σε αυτή τη βάση σχεδιάστηκαν και υλοποιήθηκαν πλήθος βελτιώσεων μεταξύ των οποίων η χρήση «δεξαμενών μνήμης» (buffered memory blocks) για την εισερχόμενη και εξερχόμενη ροή, ελαχιστοποιώντας τη χρήση μνήμης για μεγάλα σετ εικόνων.

Στη συνέχεια όλων αυτών, εκτελέστηκαν τρία βασικά σενάρια λειτουργίας, τα οποία έχουν ήδη υλοποιηθεί στο MediGrid και αυτά είναι: Διανομή, Αποθήκευση και Μετακίνηση δεδομένων (πχ η μετακίνηση από ένα σταθμό εργασίας σε κάποιο κόμβο του «πλέγματος» για επεξεργασία). Στην Εικόνα 7 περιγράφεται ο τρόπος με τον οποίο κινείται ένα μήνυμα στην περίπτωση της χρήσης του Grid-DICOM δρομολογητή. Το κεντρικό τμήμα αναπαριστά την υποδομή GSI που πραγματοποιεί την αυθεντικοποίηση και ταυτοποίηση των χρηστών στην υποδομή. Στα αριστερά, υπάρχει η διεπαφή που υλοποιεί το πρωτόκολλο DICOM και στα δεξιά η αντίστοιχη διεπαφή που μεταφέρει τα μηνύματα DICOM στο πλέγμα και η αντίστοιχη πορεία για την ανάκτηση των δεδομένων από το πλέγμα. Οι δοκιμές λειτουργίας της εφαρμογής πραγματοποιήθηκαν κάτω από τις ίδιες συνθήκες συστήματος σε τρεις διαφορετικές ομάδες εικόνων (MR –250 MB, CT –250 MB, CR - 800MB). Από τις μετρήσεις των δοκιμών έγιναν οι παρατηρήσεις που παρουσιάζονται στον πίνακα 1. Πιο συγκεκριμένα, στην περίπτωση της απευθείας αποστολής σε κάποιον κόμβο οι ταχύτητες είναι σχετικά υψηλές αλλά ο ρυθμός μειώνεται στις επόμενες περιπτώσεις καθώς εμπλέκονται οι διαδικασίες της κρυπτογράφησης, αυθεντικοποίησης και ταυτοποίησης. Ακόμη, το πλήθος των εικόνων επηρεάζει τον ρυθμό μεταφοράς αντιστρόφως ανάλογα. Υπάρχει καλή απόδοση σε μικρό αριθμό σειρών με πολλές εικόνες αλλά χειρότεροι χρόνοι στη μεταφορά μεγάλων εικόνων διαφορετικών σειρών. Παρόμοια ποιοτικά αποτελέσματα προκύπτουν και στα δύο άλλα σενάρια.

Τα αποτελέσματα αυτά εξαρτώνται από το υλικό και το εύρος ζώνης που χρησιμοποιήθηκε. Βέβαια, η αρχιτεκτονική που υιοθετήθηκε θεωρείται ίδια με εκείνη του MediGrid. Το MediGrid είναι μια κατανεμημένη εφαρμογή που ενσωματώνει διάφορα στοιχεία λογισμικού ανάπτυξης σε περιβάλλον πλέγματος και προωθεί την δημιουργία κοινά μοιραζόμενων πόρων που βρίσκονται κάτω από τον ίδιο Εικονικό Οργανισμό (VO). Στην εφαρμογή αυτή, (όπως και σε κάθε εφαρμογή που χρησιμοποιεί το πλέγμα) γίνεται διαφανής χρήση απομακρυσμένων πηγών και στηρίζεται η συνεργασία ιατρών που βρίσκονται γεωγραφικά μακριά. Ο στόχος της εργασίας [7] είναι να μπορούν οι γιατροί να χρησιμοποιούν υπολογιστές υψηλής απόδοσης και αποθηκευτικά συστήματα για την πραγματοποίηση των λειτουργιών της επεξεργασίας, ανάλυσης, οπτικοποίησης και διαχείρισης των εικόνων PET/SPECT μέσα από ένα απλό πλαίσιο. Για την υλοποίηση αυτής της εφαρμογής χρησιμοποιήθηκε το προγραμματιστικό εργαλείο ImageJ, μια

ανεξάρτητη πλατφόρμας και ανοιχτού κώδικα υλοποίηση σε Java για την επεξεργασία και αποθήκευση μιας μελέτης DICOM.

Πίνακας 2: Τα ποσοτικά αποτελέσματα από την εκτέλεση των τριών διαφορετικών σεναρίων μέσα από εναλλακτικές προσεγγίσεις. Οι μετρήσεις αφορούν το ρυθμό μεταφοράς (MB/S)

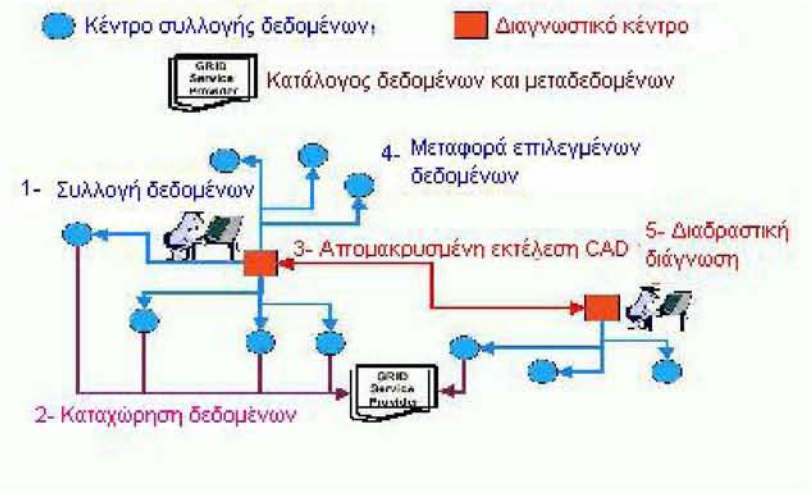
Διεργασία	MR	CT	CR
Σενάριο 1: Διανομή			
Α. Σταθμός εργασίας DICOM -> κόμβος πλέγματος	3,6	3,9	4,1
Β. Σταθμός εργασίας SSL-DICOM -> κόμβος πλέγματος	2,4	2,2	2,5
Γ. Σταθμός εργασίας Grid-DICOM -> κόμβος πλέγματος	2,0	2,1	2,3
Δ. Τοπικότητα + Δρομολογητής -> κόμβος πλέγματος	0,73	0,67	1,8
Ε. Τοπικότητα + DGIS -> κόμβος πλέγματος	1,0	0,56	0,25
Σενάριο 2: Αποθήκευση			
Τοπικότητα + Δρομολογητής -> Δρομολογητής + PACS	0,68	0,6	1,2
Σενάριο 3:Μετακίνηση			
Σταθμός εργασίας Grid-DICOM -> Τοπικότητα + Δρομολογητής -> Δρομολογητής + PACS	0,6	0,58	1,0

2.4.3 Το σύστημα MAGIC-5 – Εφαρμογή περίπτωσης [8]

Η δεύτερη εφαρμογή που εξετάζεται είναι η MAGIC-5 που αποτελεί μια ιατρική εφαρμογή στηριζόμενη στην υποδομή πλέγματος. Η ανάπτυξη Υπολογιστικών Συστημάτων Ανίχνευσης (CAD) μπορεί να βελτιώσει σημαντικά την ανίχνευση και τη διάγνωση του καρκίνου λειτουργώντας είτε υποστηρικτικά της διάγνωσης του ειδικού είτε ως ο μηχανισμός εκείνος που θα συλλέγει εικόνες που αντιστοιχούν στην μεγαλύτερη πιθανότητα καρκίνου. Η ποσότητα, όμως, των δεδομένων που παράγονται, για παράδειγμα, από τους περιοδικούς ελέγχους δεν μπορούν να διαχειριστούν από ένα απλό υπολογιστικό κέντρο και προς την κατεύθυνση αυτή κινήθηκαν οι συντελεστές του έργου MAGIC-5. Οι δύο βασικοί στόχοι αυτού του έργου είναι η υποστήριξη της ραδιολογικής διάγνωσης παρέχοντας αυτοματοποιημένη χρήση αλγορίθμων για την ανίχνευση παθολογικών δομών και τη βελτίωση της υπολογιστικής ταχύτητας και της προσβασιμότητας στα δεδομένα.

Σύμφωνα με την εργασία των I. Mitri et al. [8], η συλλογή εικόνων σε μεγάλη κλίμακα δημιουργεί εκ των πραγμάτων την ανάγκη για κατανεμημένες βάσεις δεδομένων συμπεριλαμβανομένων νοσοκομείων και διαγνωστικών κέντρων. Όπως φαίνεται από τη σχηματική αναπαράσταση (Εικόνα 8), τα κέντρα συλλογής δεδομένων και τα διαγνωστικά κέντρα επικοινωνούν μεταξύ τους μέσω καταλόγων δεδομένων και μεταδεδομένων. Επιπλέον, ένα σύστημα CAD (Computer Aided Detection) επωφελείται από το πλαίσιο των κατανεμημένων βάσεων, καθώς δίνει τη δυνατότητα εκπαίδευσης των αλγορίθμων ανίχνευσης σε μεγαλύτερο δείγμα δεδομένων εκπαίδευσης, βελτιώνοντας σημαντικά τους δείκτες ευαισθησίας και

ειδικότητας. Οπότε, με τη χρησιμοποίηση τους στην διάγνωση μπορεί να μειωθεί ο χρόνος μεταξύ της παραγωγής μιας εικόνας και της ολοκλήρωσης της διάγνωσης. Για την διαχείριση, λοιπόν, αυτών των δεδομένων το πλέγμα θεωρείται ως ο βέλτιστος τρόπος υλοποίησης.



Εικόνα 8: Ο τρόπος λειτουργίας του MAGIC-5

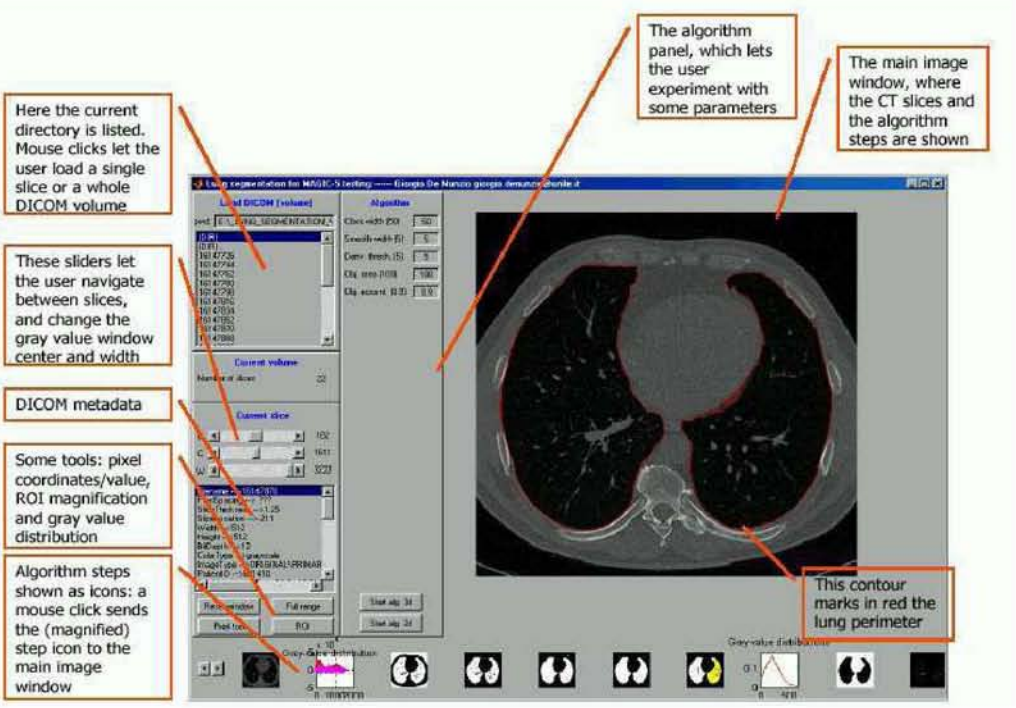
Τα δεδομένα αποθηκεύονται σε τοπικές πηγές και καταχωρούνται σε μια υπηρεσία που ονομάζεται Data Catalogue (Κατάλογος Δεδομένων) μαζί με τα μεταδεδομένα. Ο μηχανισμός αυτός επιλέγει τις εικόνες βάσει κάποιων χαρακτηριστικών σχετικών με τα μεταδεδομένα και γίνεται αντιστοίχηση μεταξύ του φυσικού (όπως είναι αποθηκευμένα) και λογικού ονόματος. Οι εικόνες αποθηκεύονται αφού αναλυθούν και ομαδοποιηθούν. Ο χρήστης όπως είναι φυσικό, αλληλεπιδρά με το σύστημα μέσω μιας γραφικής διεπαφής που διαθέτει τρία μενού επιλογών το καθένα από τα οποία παρουσιάζει στοιχεία για τον Ασθενή, την Εικόνα και τα επίπεδα διάγνωσης CAD (δηλαδή αν η εικόνα δείχνει έναν υγιή ιστό ή όχι). Ένα παράδειγμα παρουσίασης των παραπάνω θα μπορούσε να είναι η εικόνα μιας μαστογραφίας στην οποία είναι σημειωμένες περιοχές "ασβεστωμάτων" και μαζών με διαφορετικά χρώματα. Η παραπάνω διεπαφή οδηγεί στην εκτέλεση τριών βασικών λειτουργιών σχετικά με τον κατάλογο δεδομένων (Data Catalogue):

✓καταχώρηση ενός νέου ασθενή που ταυτοποιείται με την παραγωγή ενός μοναδικού κωδικού

✓καταχώρηση μιας νέας εξέτασης σχετική με κάποιον ήδη υπάρχων ασθενή

✓εκτέλεση ερωτημάτων στον κατάλογο δεδομένων (DC) για ανάκτηση των φυσικών ονομάτων των αρχείων αναφορικά με τις εξετάσεις ενός ασθενή.

Οι εικόνες παρουσιάζονται με βάση τον τύπο που έχει επιλεγεί από τον ραδιολόγο, ο οποίος μπορεί να επέμβει εισάγοντας ή μετατρέποντας μια διάγνωση ή τις



Εικόνα 9: Εικόνα από τη διεπαφή του ραδιολόγου με χρήσιμες πληροφορίες για την εικόνα, τα μεταδεδομένα της και με εργαλεία για απλές λειτουργίες.

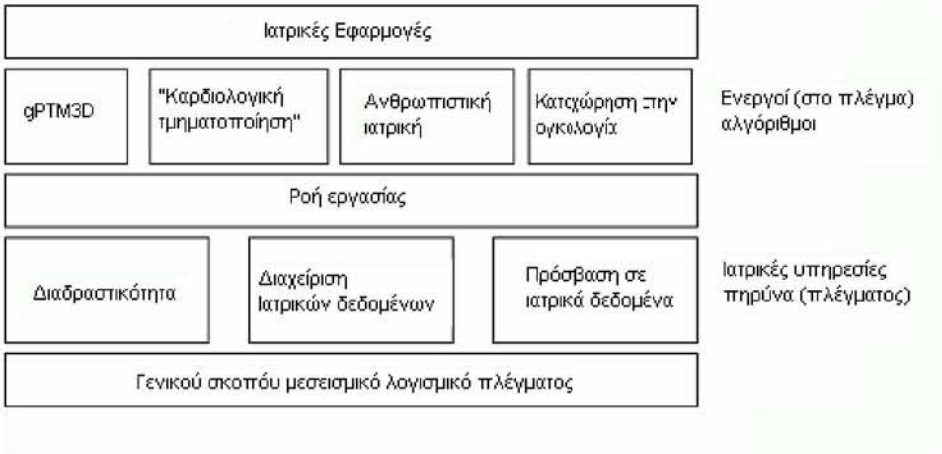
σημειώσεις/ παρατηρήσεις που τη συνοδεύουν. Λειτουργίες όπως η μεγέθυνση, η αλλαγή αντίθεσης και φωτεινότητας είναι μερικές από τις δυνατότητες της διεπαφής. Οι αλγόριθμοι ανίχνευσης υλοποιούνται και σε εικόνες τύπου DICOM εμφανίζοντας στο χρήστη την εικόνα και όλες τις χρήσιμες πληροφορίες που την αφορούν.

2.4.4 Το σύστημα AGIR – Εφαρμογή περίπτωσης [10]

Μια ακόμη προσέγγιση προσανατολισμένη στην επεξεργασία και ανάλυση ιατρικών εικόνων αποτελεί το έργο των German et al.[10] που καλείται AGIR (Grid Analysis of Radiological Data). Σε αυτή την εργασία περιγράφεται ο συνδυασμός του λογισμικού που είναι διαθέσιμο στο πλέγμα και στοχεύει σε απαιτήσεις για εφαρμογές επεξεργασίας ιατρικών δεδομένων, από τη μια, και την έρευνα σε επίπεδο αλγορίθμων για κλινικές εφαρμογές, από την άλλη. Τέθηκαν, λοιπόν, δύο βασικοί στόχοι, η αλγοριθμική έρευνα και υλοποίηση και οι υπηρεσίες του καλούμενου "New Grid" [10]. Η μεθοδολογία που ακολουθήθηκε αφορά τον συνδυασμό εξειδικευμένων εργαλείων όπως κλινικές εφαρμογές, αλγόριθμοι ανάλυσης ιατρικών εικόνων, IT συστήματα, κλπ και την ανασκόπηση του λογισμικού που ήδη έχει αναπτυχθεί στο συγκεκριμένο πεδίο εφαρμογής.

Η ανάπτυξη κάθε ιατρικής εφαρμογής στο πλέγμα μπορεί να δομηθεί σε τέσσερα επίπεδα όπως παρουσιάζεται στην Εικόνα 10. Στο πιο χαμηλό επίπεδο βρίσκονται οι βασικές υπηρεσίες του πλέγματος. Το αμέσως, επόμενο επίπεδο αναπαριστά τις υπηρεσίες που είναι σχεδιασμένες για ιατρικές εφαρμογές (ιατρικές υπηρεσίες πυρήνα). Σε αυτό το επίπεδο διακρίνουμε τρεις υποκατηγορίες: Πρόσβαση στα ιατρικά δεδομένα, διαχείριση και διαδραστικότητα. Στο τρίτο επίπεδο βρίσκονται οι

αλγόριθμοι ή/και τα εργαλεία για επεξεργασία. και στο τέταρτο επίπεδο βρίσκονται οι ιατρικές εφαρμογές



Εικόνα 10: Πλαίσιο ανάπτυξης μιας ιατρικής εφαρμογής.

Αναφορικά με τις ιατρικές υπηρεσίες πυρήνα, η πρόσβαση μέσω του πλέγματος σε ιατρικές εικόνες συνεπάγεται την δυνατότητα μεταφοράς πολύ μεγάλων σετ δεδομένων. Οι βασικές λειτουργίες διαχείρισης των δεδομένων στο πλέγμα συνδέονται με τους εξυπηρετητές DICOM κάνοντας χρήση της διεπαφής DCMTK (βλ κεφάλαιο 2.3 για περισσότερες πληροφορίες).

Οι German et al. εξέτασαν τέσσερις ιατρικές εφαρμογές από την οπτική γωνία εκτέλεσης αλγορίθμων για ανακατασκευή και τμηματοποίηση εικόνων, γενικότερα, την υλοποίηση ιατρικών εφαρμογών στο πλέγμα. Σε κάθε μια από τις εφαρμογές αυτές «ελέγχεται» αν και σε πιο βαθμό ακολουθείται το πλαίσιο – πρότυπο που παρουσιάζουν για να καταλήξουν στην τελική κλινική/ιατρική εφαρμογή.

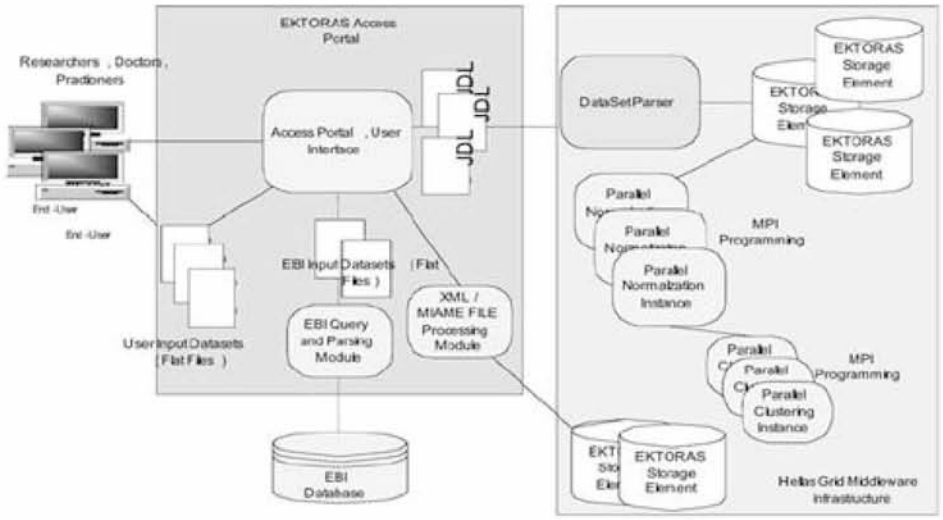
2.4.5 Το σύστημα ΕΚΤΟΡΑΣ– Εφαρμογή περίπτωσης [6]

Η ανάλυση του ανθρώπινου DNA ήταν μια από τις πρώτες εφαρμογές της βιοϊατρικής που χρησιμοποίησαν τα πλεονεκτήματα του πλέγματος. Η εφαρμογή ΕΚΤΟΡΑΣ αφορά την υλοποίηση ενός ευφυούς τρόπου επεξεργασίας βιολογικών δεδομένων από πειράματα με μικροσυστοιχίες σε περιβάλλον πλέγματος και συγκεκριμένα στην υποδομή του Hellinic Grid. Στηριζόμενοι στις δύο βασικές ιδιότητες του grid (υπολογιστική ισχύς και χωρητικότητα αποθήκευσης), υλοποιήθηκε μια εφαρμογή βασισμένη στον ιστό στοχεύοντας στην αποδοτικότερη ανάλυση και επεξεργασία των συστοιχιών cDNA. Η προκύπτουσα πλατφόρμα καλείται EKTORAS. Ο τελικός χρήστης μπορεί να δώσει ως είσοδο ένα αρχείο με δεδομένα από μικροσυστοιχίες, να επιλέξει έναν αλγόριθμο επεξεργασίας και να αποθηκεύσει τα αποτελέσματα αυτής της προ-επεξεργασίας σε κάποιο SE. "Παραλληλοποιώντας" την εφαρμογή και κατανέμοντας τις εργασίες της στους κόμβους δημιουργείται μια εικονική βάση δεδομένων για τα πειράματα

μικροσυστοιχιών. Η αναφορά της εδώ γίνεται διότι περιέχει μια συνολική εικόνα για την χρήση του grid με απλό τρόπο και μας δίνει στοιχεία για την "παραλληλοποίηση" εργασιών κάνοντας χρήση του προτύπου MPI που χρησιμοποιείται για την εκτέλεση παράλληλων εργασιών στο grid.

Η αρχιτεκτονική σχεδίαση της εφαρμογής αποτελείται από δύο βασικά τμήματα (Εικόνα 11). Πιο αναλυτικά, ο χρήστης αλληλεπιδρά με μια πλατφόρμα στην οποία μπορεί να εισάγει δικά του δεδομένα ή να ανακτήσει για τους πειραματικούς του σκοπούς κάποιο σετ δεδομένων από την Βάση δεδομένων της EBI. Παρέχεται, επίσης, η δυνατότητα επιλογής του επιθυμητού αλγορίθμου επεξεργασίας και καθορισμού των παραμέτρων του πειράματος – ανάλυσης. Στη συνέχεια, παράγεται ένα αρχείο JDL για το πείραμα με το οποίο γίνεται η υποβολή στην υποδομή. Ο ορισμός των αρχείων JDL και ο ρόλος τους περιγράφονται αναλυτικά στο κεφάλαιο 3. Το αρχείο JDL, λοιπόν, που παράγεται περιέχει όλες τις απαραίτητες παραμέτρους για την αποστολή του ερωτήματος- εργασίας στο πλέγμα και υποβάλλεται με τη βοήθεια του συστήματος διαχείρισης ροής των εργασιών WMS (Workload Manager System). Στη συνέχεια, τα δεδομένα εισόδου τροποποιούνται σε κατάλληλες δομές δεδομένων ώστε να είναι εφικτή η εκτέλεσή τους (πχ αρχεία εργασίας Matlab/Octave). Το αποτέλεσμα της παραπάνω διαδικασίας αποθηκεύεται σε κάποιο SE και η διαχείρισή του γίνεται με τη χρήση εντολών του LCG File Catalog (LFN). Το επόμενο βήμα αφορά το "σπάσιμο" του κώδικα, λαμβάνοντας υπόψη όλες τις απαιτούμενες ρυθμίσεις για την ορθή εκτέλεση των εργασιών αντιμετωπίζοντας τις εξαρτήσεις του κώδικα και προσαρμόζοντας τον στις απαιτήσεις για παράλληλη επεξεργασία. Η ροή της εφαρμογής, που αναφέρεται στην ροή των εργασιών που απαρτίζουν την υλοποίηση στο πλέγμα και την μεταξύ τους επικοινωνία, διακρίνεται σε τρεις διαφορετικούς τύπους

- ✓παράλληλη ροή
- ✓σειριακή ροή
- ✓"δικτυωμένη" ροή



Εικόνα 11: Η συνολικη αρχιτεκτονική του συστήματος HECTOR.

Η παραλληλοποίηση μιας εφαρμογής γίνεται με την εκτέλεση διακριτών εργασιών από διαφορετικούς κόμβους. Ο χρόνος εκτέλεσης μιας εργασίας είναι ένας παράγοντας που πρέπει να ληφθεί υπόψη αφού η προσπάθεια εκτέλεσης απλών υπολογισμών με παράλληλο τρόπο μπορεί να οδηγήσει σε καθυστέρηση και όχι επιτάχυνση ενός υπολογισμού. Η παράλληλη εκτέλεση εργασιών σημαίνει ότι πιθανόν να παρουσιάζονται εξαρτήσεις μεταξύ τους και άρα απαιτείται μια υπηρεσία διαχείρισης και συγχρονισμού. Οπότε χρησιμοποιείται το μοντέλο προγραμματισμού MPI που δίνει τη δυνατότητα επικοινωνίας μεταξύ των εργασιών παρέχοντας την μέγιστη απόδοση από την παράλληλη εκτέλεση των αλγορίθμων επεξεργασίας των μικροσυστοιχιών. Ένα παράδειγμα ψευδοκώδικα παράλληλης επεξεργασίας είναι αυτό που παρουσιάζεται στην ακολούθως, το οποίο περιέχει κάποιες από τις βασικές συναρτήσεις του πρότυπου MPI όπως είναι οι:

1. MPI_Init()
2. MPI_Comm_rank()
3. MPI_Recv()
4. MPI_Finalize()

Τέλος, η συγκέντρωση όλων των αποτελεσμάτων γίνεται από έναν μόνο κόμβο αλλά η ομαδοποίηση τους τρέχει παράλληλα σε διαφορετικούς κόμβους. Όπως είναι εμφανές και από την Εικόνα 11 ο τελικός χρήστης δεν έρχεται σε επαφή με το πλέγμα, καθώς όλη η διαδικασία υποβολής των εργασιών και η επεξεργασία τους είναι μια αδιαφανής διαδικασία, χωρίς αυτό να τους εμποδίζει να παρακολουθεί

```
tinclude <mpi.h>
int main(int argc, char *argv[])
{
. . . // initializing MPI
MPI_Init(&argc, &argv);
// learn node number
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
// load files (in parallel for each file)
// Start Find Bad Points (in parallel for each experiment)
[exptab,TotalBadpoints]=FindBadpoints(datstruct,t,exprp,imgsw);
// Normalize Data (in parallel for each experiment)
[DataCellNormLo]=NormalizationLO(exptab,exprp,t,gnID);
Gather processed data (On node 0)
for (i=1;i<NumOfExperiments;i++)
MPI_Recv(rcvbuf,count,datatype,i,tag,MPI_INT, MPI_COMM_WORLD,
&status);
/* MPI shutdown MPI */
MPI_Finalize ();
//Filter Replicates
[DataCellFiltered]=FilterReplicates2(DataCellNormLo);
//Statitical Test
[DataCellStat]=MA_StaTestExp_New_total(DataCellFiltered,DataCellNormLo)
//CLUSTERING
```

την κατάσταση στην οποία βρίσκεται η εκτέλεση της εργασίας που έχουν υποβάλει.

Το σύστημα ΕΚΤΟΡΑΣ προσπαθεί να δώσει τη δυνατότητα πρόσβασης σε δεδομένα και αποτελέσματα από πειράματα μικροσυστοιχιών με έναν οικονομικό και αποδοτικό τρόπο εκτέλεσης των πειραμάτων. Δηλαδή, τα αποτελέσματα των πειραμάτων αποθηκεύονται σε κάποιο SE από το οποίο μπορούν να ανακτηθούν μέσω μιας απλής αναζήτησης από τον φυλλομετρητή του χρήστη. Επομένως, μπορούν να χρησιμοποιηθούν τα αποτελέσματα και οι πληροφορίες που έχουν εξαχθεί από προηγούμενα πειράματα αν αυτό καλύπτει τις ανάγκες και απαιτήσεις του πειράματος ενός τελικού χρήστη. Είναι, λοιπόν, σαφές ότι υλοποίηση ενός τέτοιου συστήματος προϋποθέτει την συνεργασία διαφορετικών θεματικών περιοχών και παρουσιάζεται ως ένα σημείο εκκίνησης για την δημιουργία ενός περισσότερο ολοκληρωμένου περιβάλλοντος.

Το πλήθος των εφαρμογών που έχουν υλοποιηθεί, όπου ενδεικτικά κάποιες αναφέρθηκαν ανωτέρω, καθιστούν αδιαμφισβήτη την ανάγκη για διαχείριση των βιοϊατρικών δεδομένων και την αξιοποίηση τους για την παραγωγή γνώσης, πέρα από την έμφαση που έχει δοθεί μέχρι σήμερα στην ανάπτυξη τεχνολογιών συλλογής δεδομένων και παραγωγής εικόνων ή βίντεο. Το επόμενο, λοιπόν, βήμα είναι να καλυφθεί το κενό στην διαχείριση, ανάλυση και ερμηνεία των δεδομένων με έναν αποδοτικό και αποτελεσματικό τρόπο. Σε αυτό το πλαίσιο, η ανάπτυξη εφαρμογών προσανατολίζεται στην εκμετάλλευση της υποδομής του πλέγματος (Grid Infrastructure).

3. ΠΡΟΤΕΙΝΟΜΕΝΗ ΑΝΤΙΜΕΤΩΠΙΣΗ

Ο κατανεμημένος υπολογισμός δεν αποτελεί απλά ένα όραμα αλλά μια πραγματικότητα. Η υπολογιστική βιολογία, η βιοπληροφορική, η επεξεργασία και ανάλυση σημάτων και εικόνων καθώς και όλες οι εκφάνσεις της βιοϊατρικής προσανατολίζονται στην υλοποίηση εφαρμογών που "διασκορπίζονται" με στόχο την ελαχιστοποίηση του κόστους. Στο παρόν κεφάλαιο, παρουσιάζεται η αρχιτεκτονική της εφαρμογής IMAGEGRID κινούμενη σε αυτά τα πλαίσια. Στο προηγούμενο κεφάλαιο, έγινε αναφορά σε κάποιες εφαρμογές που αξιοποιούν η καθεμιά με το δικό της τρόπο και σε διαφορετικό βαθμό τις τεχνολογίες πλέγματος. Σε κάποιες από αυτές δόθηκε έμφαση στην χρήση του "υπολογιστικού πλέγματος" (computational grid) αξιοποιώντας τα CEs (Computing Elements) και σε άλλες το "πλέγμα δεδομένων" (DataGrid) αξιοποιώντας τα διαθέσιμα SEs (Storage Elements). Η εφαρμογή αυτή με τη σειρά της κινείται στα πλαίσια του Datagrid. Το κεφάλαιο 3 αποτελείται από 7 υπό-κεφάλαια στο καθένα από τα οποία γίνεται η παρουσίαση των επιμέρους τμημάτων της εφαρμογής καθώς και ο συνδυασμός τους στην τελική μορφή της εφαρμογής:

- ✓ Πρόσβαση στο Πλέγμα
- ✓ Υποβολή εργασιών
- ✓ Διαχείριση εργασιών
- ✓ Η Βάση Δεδομένων
- ✓ Οι Υπηρεσίες Ιστού (Web Services)
- ✓ Η συνολική αρχιτεκτονική του συστήματος

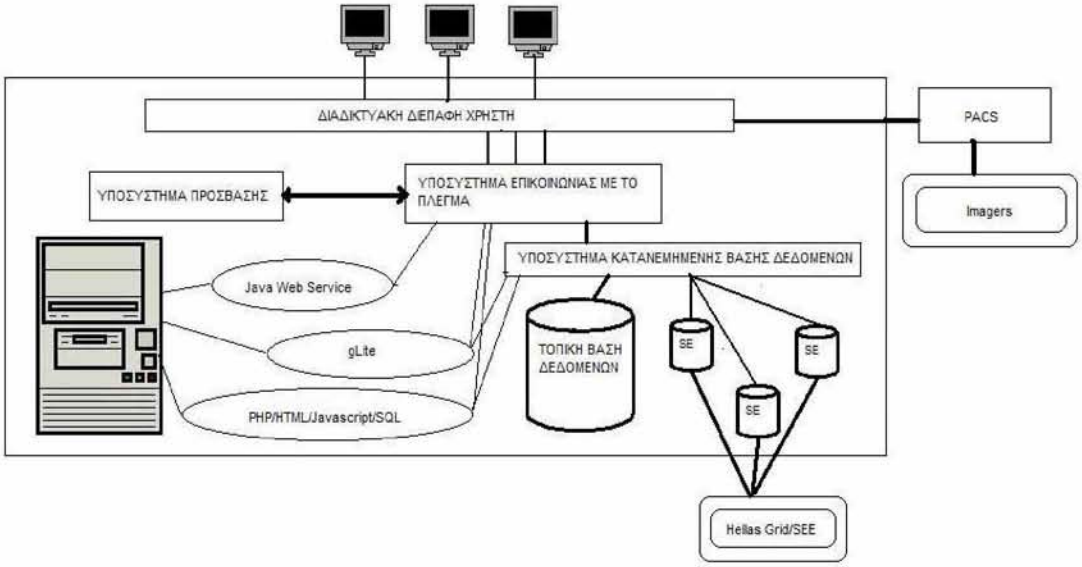
3.1 Η συνολική αρχιτεκτονική του συστήματος

Η έννοια του λογισμικού και ο ρόλος που παίζει έχουν διαποτίσει τον κόσμο μας σε τέτοιο βαθμό ώστε να θεωρείται κάτι απόλυτα φυσιολογικό. Στις μέρες μας το λογισμικό κατέχει μια εξέχουσα θέση στη ζωή μας, λειτουργώντας άλλοτε φανερά και άλλοτε παρασκηνιακά, και η ανάπτυξη συστημάτων επηρεάζει αναπόφευκτα την ασφάλεια ή/και την υγεία μας. Αυτό σημαίνει ότι ο σωστός σχεδιασμός ενός συστήματος αποτελεί αναγκαία προϋπόθεση ανάπτυξης λογισμικού ώστε να εξασφαλίζεται η ποιοτικότερη βελτίωση της ζωής. Για αυτό το λόγο χρησιμοποιούνται εργαλεία (βοηθήματα ή αυτοματοποιημένα συστήματα) για την επίτευξη του βέλτιστου δυνατού αποτελέσματος. Ένα από αυτά τα εργαλεία είναι τα διαγράμματα UML. Η UML παρέχει τέσσερις τύπους διαγραμμάτων για την μοντελοποίηση της δυναμικής συμπεριφοράς ενός συστήματος καθένα από τα οποία εξυπηρετεί διαφορετικό σκοπό[3]. Τα διαγράμματα αυτά είναι:

- Διαγράμματα Καταστάσεων
- Διαγράμματα Ακολουθίας
- Διαγράμματα Συνεργασίας
- Διαγράμματα Δραστηριοτήτων

Ένα διάγραμμα δραστηριοτήτων περιγράφει πως εκτελούνται οι διάφορες λειτουργίες ενός συστήματος και δείχνει τη ροή εργασίας. Στη συνέχεια παρουσιάζεται το διάγραμμα δραστηριοτήτων της εφαρμογής IMAGEGRID (Εικόνα 13), περιγράφοντας όλες τις δραστηριότητες που εκτελούνται στο σύστημα ως γεγονότα ξεκινώντας από ένα αρχικό σημείο ή σημείο εκκίνησης και το μονοπάτι που πρέπει να ακολουθήσει για την αποτελεσματική ολοκλήρωση τους σε ένα σημείο τερματισμού (για περισσότερες λεπτομέρειες βλ. ΠΑΠΑΡΤΗΜΑ Γ). Σχετικά με τη σημειογραφία που χρησιμοποιείται ο αρχικός και τελικός κόμβος αναπαρίστανται από μια μαύρη βούλα, τα "ορθογώνια" σχήματα αναπαριστούν τις καταστάσεις και τα βέλη τις μεταβάσεις από τη μια κατάσταση στην άλλη. Τέλος, ο ρόμβος αναπαριστά ένα σημείο απόφασης για το ποια δραστηριότητα θα εκτελεστεί στη συνέχεια [17]. Η αρχιτεκτονική του συστήματος περιγράφεται συνοπτικά από την ακόλουθη σχηματική αναπαράσταση (Εικόνα 12). Όπως φαίνεται ο χρήστης αλληλεπιδρά με ένα υποσύστημα επικοινωνίας που βρίσκεται σε αρμονία με το υποσύστημα πρόσβασης. Το υποσύστημα επικοινωνίας με το πλέγμα συνδιάζει την υπηρεσία ImagegridWS με το μεσεισμικό λογισμικό και είναι υπεύθυνο για την διαχείριση (αποθήκευη και ανάκτηση) των εικόνων. Το υποσύστημα κατανεμημένης βάσης δεδομένων ορίζει τον τρόπο με τον οποίο θα αποθηκευτούν τα δεδομένα τόσο στην τοπική βάση δεδομένων όσο και στα αποθηκευτικά στοιχεία της υποδομής πλέγματος. Τα παρόντα υποσυστήματα περιγράφονται με περισσότερες λεπτομέρειες στην συνέχεια του κεφαλαίου.

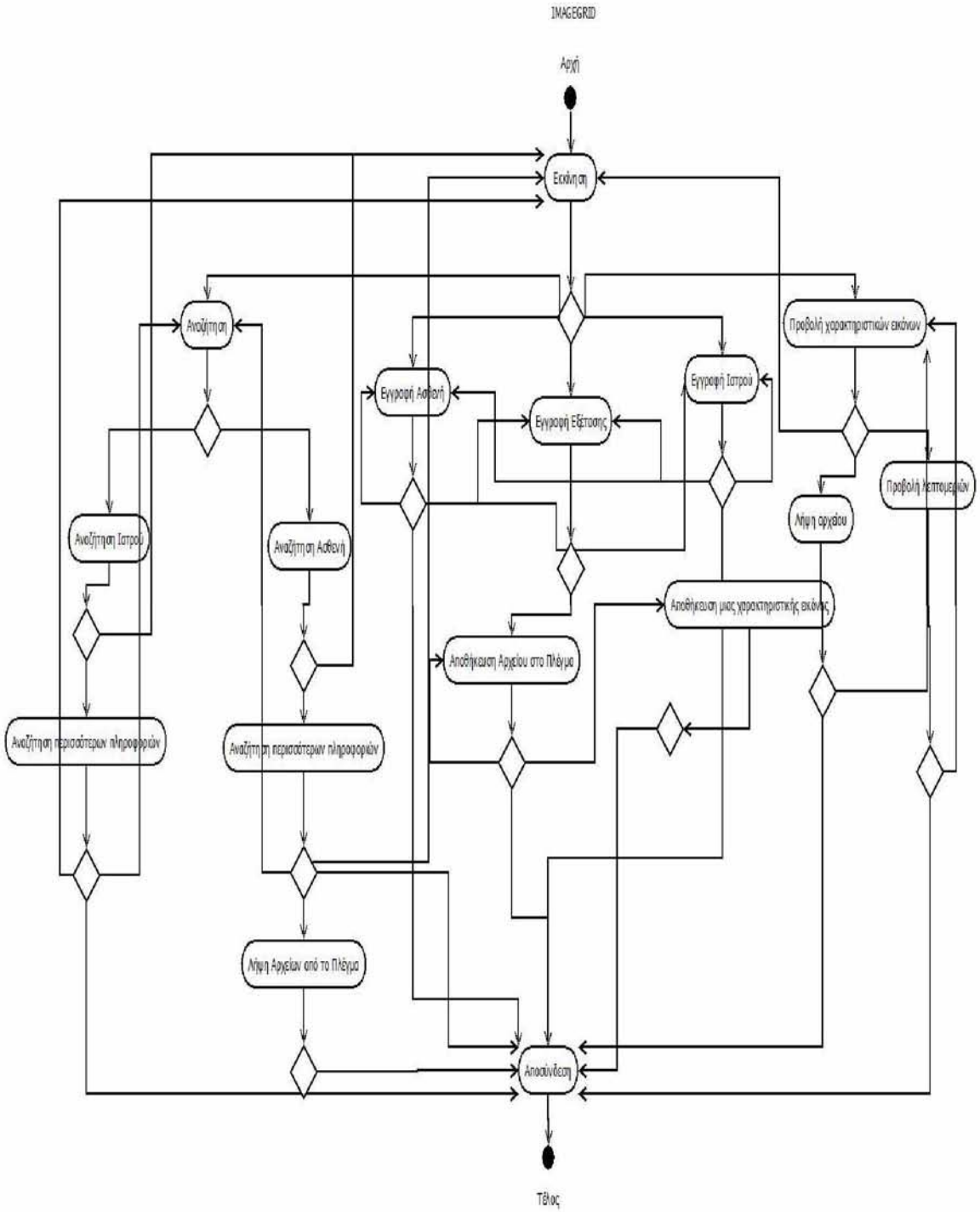
[3] Όλα τα συστήματα έχουν στατική και δυναμική συμπεριφορά. Η στατική δομή περιγράφεται με κλάσεις, συσχετίσεις, αντικείμενα και συστατικά.



Εικόνα 12: Η συνολική αρχιτεκτονική του συστήματος

Για την οικοδόμηση του παραπάνω συστήματος χρησιμοποιήθηκαν η γλώσσα προγραμματισμού PHP για την κατασκευή ιστοσελίδων με δυναμικό περιεχόμενο και η HTML για την εμφάνιση των αποτελεσμάτων που επιστρέφει ο εξυπηρετητής. Για απλές λειτουργίες και ελέγχους εγκυρότητας στα δεδομένα εισόδου και την υπόδειξη των απαραίτητων πεδίων κατά την συμπλήρωση των φορμών χρησιμοποιείται η Javascript. Για την καταχώρηση των στοιχείων χρησιμοποιούνται φόρμες υποβολής και τα ερωτήματα μέσω κώδικα PHP αποθηκεύονται στη βάση. Για παράδειγμα, για την καταχώρηση ενός νέου ασθενούς εκτελείται το αρχείο addPatient.php ελέγχοντας για την ύπαρξη μιας εγγραφής στη βάση δεδομένων με τα ίδια ακριβώς στοιχεία. Σε περίπτωση που τουλάχιστον ένα πεδίο είναι διαφορετικό θεωρείται ότι αφορά ξεχωριστή εγγραφή άρα μπορεί να αποθηκευτεί. Διαφορετικά τυπώνεται στην οθόνη ένα μήνυμα που μας πληροφορεί για ποιο λόγο δεν πραγματοποιήθηκε η καταχώρηση. Κατά αντιστοιχία πραγματοποιείται η εισαγωγή των στοιχείων του ιατρού στη βάση δεδομένων. Η συλλογή των στοιχείων γίνεται με την βοήθεια φορμών όπως αυτή που μαζεύει τα στοιχεία του ασθενή (Παράρτημα Β).

Οι συναρτήσεις ελέγχου εγκυρότητας των στοιχείων υποβολής είναι υλοποιημένες με Javascript στο HEAD τμήμα του εγγράφου. Με παρόμοιο τρόπο μπορεί να



Εικόνα 13: το διάγραμμα δραστηριοτήτων του συστήματος.

πραγματοποιηθεί αναζήτηση στη βάση για κάθε οντότητα υποβάλλοντας το κατάλληλο ερώτημα. Δηλαδή αν αναζητείται ένας ασθενής, η επιλογή των στοιχείων του από τη βάση δεδομένων γίνεται με βάση το όνομα και το επώνυμο του ή οποιοδήποτε χαρακτηριστικό επιθυμεί ο χρήστης. Σε πρώτη φάση

εμφανίζονται στοιχεία για τον ασθενή, για παράδειγμα, όπως το ονοματεπώνυμό του, η ηλικία του, το φύλο του, το ασφαλιστικό του ταμείο, η διεύθυνση, το τηλέφωνο του και το πλήθος των εξετάσεων που έχει πραγματοποιήσει ως υπερσύνδεσμο που του δώσει περισσότερες πληροφορίες σχετικά με την εξέταση και τον γιατρό που τον παρακολουθεί. Με ανάλογο τρόπο λειτουργεί η αναζήτηση για κάποιο γιατρό που παρουσιάζει το πλήθος των εξετάσεων που έχει παραγγείλει και ποιους ασθενείς αφορούν.

Εκτός όμως από αυτό, με το ερώτημα της SQL τίθενται κάποιες προϋποθέσεις που οδηγούν στην πληροφορία για το πλήθος των εξετάσεων που έχει πραγματοποιήσει ο συγκεκριμένος ασθενής. Αυτή η πληροφορία εμφανίζεται ως σύνδεσμος που καλεί το morePatientInfo.php για να δώσει περισσότερα στοιχεία για τις εξετάσεις του ασθενή, τα στοιχεία του γιατρού που τον παρακολουθεί και μια λίστα με τα αρχεία εξετάσεων.

Για κάθε εξέταση που πραγματοποιείται και καταχωρείται στη βάση δεδομένων (addExamination.php) δημιουργείται ένας νέος φάκελος μέσα στον οποίο αποθηκεύονται οι εικόνες (αρχεία) που συνδέονται με αυτή. Έτσι, στον φάκελο upload δημιουργούνται νέοι κατάλογοι που το όνομά τους προσδιορίζεται από τον κωδικό της εξέτασης μέσα στον οποίο αποθηκεύονται τα αρχεία. Αμέσως μετά ζητείται η εκτέλεση ενός αρχείου (RunMakeDIR.php) για την δημιουργία του αντίστοιχου καταλόγου στο πλέγμα.
Μέχρι αυτό το σημείο έχουμε δει την καταχώρηση των στοιχείων στη βάση δεδομένων και την αναζήτηση τους καθώς, επίσης, και τον τρόπο με τον οποίο γίνεται η δημιουργία των φακέλων που θα φιλοξενήσουν τα αρχεία. Αυτό σημαίνει ότι βρισκόμαστε σε μια φόρμα μέσω της οποίας θα διαλέξουμε το επιθυμητό αρχείο και θα πατήσουμε το κουμπί “Upload File” για αν μεταφερθεί το αρχείο σε έναν από τους παραπάνω φακέλους. Αυτή την εργασία αναλαμβάνει να διεκπεραιώσει το fileGrid_upload.php. Σε αυτή τη φάση ένα ZIP ή ένα RAR αρχείο αντιγράφεται στο πλέγμα μέσα στον φάκελο που δημιουργήθηκε στο προηγούμενο βήμα. Το αρχείο βρίσκεται προσωρινά αποθηκευμένο στον φάκελο της εξέτασης (στον εξυπηρετητή) που μέσω της RunLcgCr.php γίνεται η αποστολή του στο SE. Εφόσον αυτή η διαδικασία ολοκληρωθεί με επιτυχία εμφανίζεται μια δεύτερη φόρμα η οποία ζητά από τον χρήστη να του δώσει ένα απλό DICOM αρχείο και το αρχείο διαγράφεται από τον εξυπηρετητή. Στο αρχείο γίνεται ο έλεγχος του τύπου του αρχείου το οποίο πρέπει να είναι μόνο application/octet-stream, και καλούνται δύο άλλα αρχεία που το καθένα εκτελεί μια διαφορετική λειτουργία. Το πρώτο από αυτά είναι το exportdicom.php που λαμβάνει το αρχείο ως δεδομένα εισόδου και εξάγει τις πληροφορίες από την επικεφαλίδα αποθηκεύοντας τες στον πίνακα DICOM της βάσης δεδομένων. Το δεύτερο είναι το ImageConvert.php που μετατρέπει τον τύπο της εικόνας σε JPEG και κάνει σμίκρυνση σε αυτή. Για την εξαγωγή των στοιχείων χρησιμοποιείται το πακέτο DICOM με την υλοποίηση της συγκεκριμένης λειτουργίας που περιγράφηκε σε προηγούμενη ενότητα, δίνοντας

τη δυνατότητα στον σχεδιαστή του συστήματος να επιλέξει ο ίδιος ποια πεδία της επικεφαλίδας τον ενδιαφέρουν. Το πρωτότυπο αρχείο αντιγράφεται σε κάποιο SE στο πλέγμα ακολουθώντας την ίδια δομή καταλόγων που υπάρχει στον εξυπηρετητή χρησιμοποιώντας τις Υπηρεσίες Ιστού (Web Services) και συγκεκριμένα την ImageGridWS ενώ ο εξυπηρετητής κρατά μόνο τις μικρογραφίες των εικόνων και το αρχείο αντιγράφεται στο SE που έχει οριστεί και που αποθηκέυται στη βάση δεδομένων . Το αρχείο αυτό ζητά αν αποκτήσει πρόσβαση στην ImageGridWS, δημιουργεί έναν client στον οποίο λέει που να κοιτάξει για την περιγραφή της εφαρμογής (το αρχείο WSDL). Στη συνέχεια, ορίζει τις τιμές των παραμέτρων που αποτελούν τα ορίσματα της μεθόδου RunLcgCr που καλείται παρακάτω και αντιστοιχεί στη μέθοδο RunLcgCr στο πρόγραμμα java. Η δομή καταλόγων τόσο στον εξυπηρετητή όσο και στο πλέγμα περιγράφεται στο υποκεφάλαιο που περιγράφει το υποσύστημα κατανεμημένης βάσης δεδομένων.
Τέλος, γίνεται η μετατροπή της εικόνας από DICOM σε JPEG μορφή (προαιρετική λειτουργία) ώστε να είναι εφικτή η παρουσίαση της από τον φυλλομετρητή. Για να γίνει αυτό χρησιμοποιείται το πακέτο Image_Transform που περιλαμβάνει βιβλιοθήκες με μεθόδους για γραφικούς μετασχηματισμούς που υλοποιούνται μέσω κάποιων οδηγών όπως οι GD, Imagick2, Imagick3, IM. Αρχικά, καλούμε την βιβλιοθήκη που θέλουμε να χρησιμοποιήσουμε. Για την μετατροπή της εικόνας θα χρησιμοποιηθεί η βιβλιοθήκη "IM" που θα λειτουργήσει ως οδηγός. Οπότε, έχοντας το νέο όνομα του αρχείου το αποθηκεύουμε μέσα στον ίδιο φάκελο. Εκτός από την μετατροπή αυτή, γίνεται και σμίκρυνση της εικόνας στο 20% του αρχικού της μεγέθους.

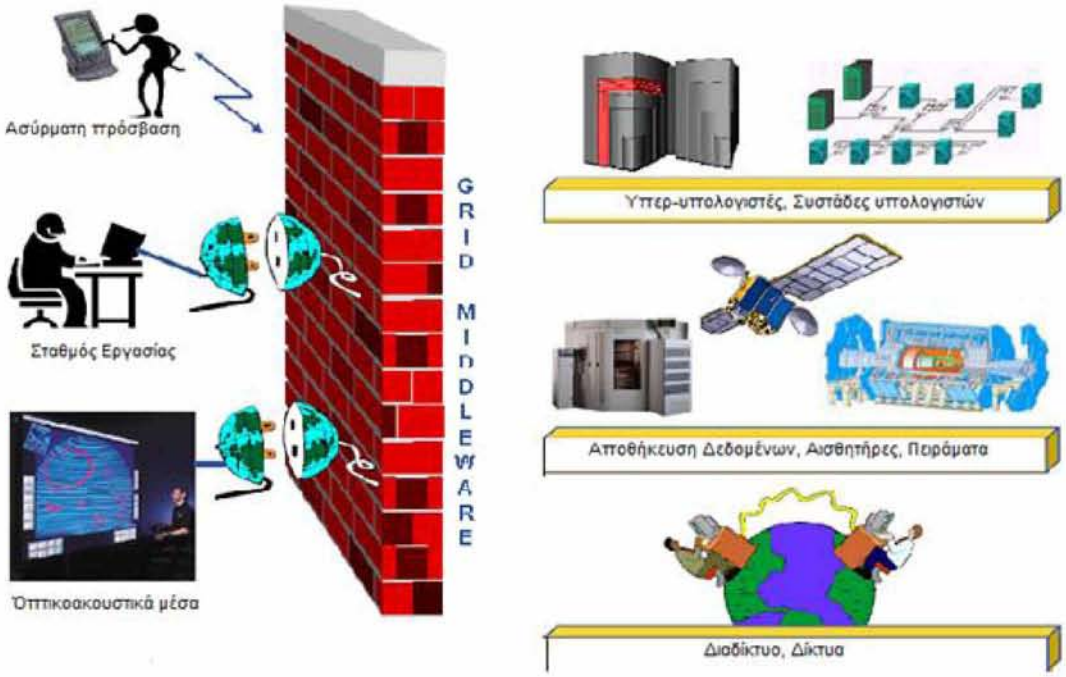
Η εφαρμογή δίνει τη δυνατότητα προβολής των εικόνων (μέσω του viewImages.php) και των μεταδεδομένων της (αυτά που έχουν επιλεγεί και όχι το σύνολο αυτών) απλά πατώντας με το ποντίκι σε έναν σύνδεσμο και μπορεί, επίσης, να κατεβάσει από το πλέγμα κάποιο αρχείο αν τον ενδιαφέρει. Η διαδικασία αυτή είναι η αντίστροφη της αντιγραφής στο πλέγμα. Χρησιμοποιώντας τον κωδικό και το όνομα του αρχείου, γίνεται αναζήτηση στη βάση δεδομένων και επιλέγεται ο κωδικός της εξέτασης. Στη συνέχεια, καλείται το αρχείο RunLcgCp.php (σε αντιστοιχία με το RunLcgCr.php) που αντιγράφει το αρχείο προσωρινά στον εξυπηρετητή από όπου και ο χρήστης «κατεβάζει» τοπικά στον υπολογιστή του καλώντας το download.php.

Συνοψίζοντας όλη την διαδικασία, ο τελικός χρήστης αλληλεπιδρά με μια διεπαφή μέσω HTML καταχωρώντας, αναζητώντας και ανακτώντας δεδομένα από τη βάση δεδομένων IMAGEGRID σχετικά με ασθενείς, γιατρούς και εξετάσεις εκτελώντας τα κατάλληλα SQL ερωτήματα. Η βάση δεδομένων κρατά στοιχεία για την αναζήτηση στο πλέγμα με στόχο την ανάκτηση τους όπως το se, τα ονόματα φακέλων και των αρχείων. Αξιοποιώντας τα χαρακτηριστικά γνωρίσματα των web services πραγματοποιούνται οι λειτουργίες της δημιουργίας νέου φακέλου, της αντιγραφής

ενός αρχείου από και προς το πλέγμα συνδυάζοντας την java με τις εντολές του gLite. Η εφαρμογή IMAGEGRID βρίσκεται στον διαδικτυακό τόπο 195.251.6.234/ypapanti που λειτουργεί ως διεπαφή μεταξύ του συστήματος PACS ενός οργανισμού υγείας και του πλέγματος. Ο πηγαίος κώδικας της εφαρμογής βρίσκεται στο Παράρτημα Β.

3.2 Υποσύστημα πρόσβασης στο Grid

Με τον όρο Grid Computing περιγράφεται όλη η περιοχή της υπολογιστικής και αποθηκευτικής διανομής πόρων μέσω του Διαδικτύου. Ουσιαστικά, αποτελεί μια συλλογή υλικού και λογισμικού που είναι διαθέσιμη σε μια ομάδα χρηστών, οι οποίοι πληρούν κάποιες συγκεκριμένες προϋποθέσεις. Το πλέγμα αποτελεί μια ολοκληρωμένη δομή υλικού και λογισμικού και διακρίνεται σε τρία διαφορετικά επίπεδα τεχνολογίας. Αναφορικά με το υλικό, η υποδομή αποτελείται από επεξεργαστές, σκληρούς δίσκους και τις δικτυακές συνδέσεις. Από πλευράς λογισμικού, αποτελείται από το “middleware” που τοποθετείται μεταξύ του πλέγματος και του χρήστη από τον οποίο, όμως, δεν είναι άμεσα προσβάσιμο, και σε επίπεδο εφαρμογών που περιλαμβάνει τις εφαρμογές των χρηστών μεταξύ άλλων. Στην πραγματικότητα, το ενδιάμεσο αυτό λογισμικό εξετάζει τους διαθέσιμους πόρους σε ένα δίκτυο και “αποφασίζει” που θα αποθηκεύσει ή από που θα ανακτήσει δεδομένα, που θα εκτελεστούν οι εργασίες που υποβλήθηκαν και, ακόμη, χειρίζεται θέματα ασφαλείας και αυθεντικοποίησης. Οι χρήστες αλληλεπιδρούν με αυτό (το ενδιάμεσο λογισμικό) που “μεταφράζει” τις εντολές υποβολής δεδομένων στην υποδομή ή ανάκτησης αυτών με έναν φιλικό τρόπο και τα δεδομένα μπορούν να παρουσιαστούν κάνοντας χρήση διάφορων οπτικοακουστικών μέσων. Η εικόνα 14 παρουσιάζει τα διάφορα στοιχεία που βρίσκονται στις δύο πλευρές του ενδιάμεσου λογισμικό τα οποία χρησιμοποιεί και από τα οποία χρησιμοποιείται.



Εικόνα 14: Η υποδομή και τα χαρακτηριστικά του πλέγματος.

Για την απόκτηση πρόσβασης στην υποδομή ακολουθείται η διαδικασία αυθεντικοποίησης του χρήστη ώστε να διασφαλιστεί η εφαρμογή της πολιτικής ασφαλείας. Η διαδικασία αυτή περιλαμβάνει τέσσερα βασικά βήματα[10]:

1. Την απόκτηση ενός Ψηφιακού Πιστοποιητικού από μια Αρχή Πιστοποίησης (Για τους Έλληνες χρήστες η πρόσβαση στην υποδομή γίνεται μέσω του HellasGrid https://access.hellasgrid.gr/)
2. Την δημιουργία ενός λογαριασμού σε κάποια διεπαφή (UI) του HellasGrid (HellasGrid User Interface π.χ. ui01.isabella.grnet.gr)
3. Την προσχώρηση σε έναν Εικονικό Οργανισμό (Virtual Organization – VO) (Οι χρήστες της ΝΑ Ευρώπης (συμπεριλαμβανομένου του HellasGrid) ανήκουν στον SEE-VO)
4. Και, τέλος, την εισαγωγή του ψηφιακού πιστοποιητικού στο UI όπου έχει δημιουργηθεί ο λογαριασμός. (Μεταφορά του πιστοποιητικού μέσω ενός FTP Client (WinSCP) στο UI και μετατροπή από .pcks12 σε .pem μορφή)

3.3 Υποσύστημα Επικοινωνίας με το στο Grid

3.3.1 Υποβολή Εργασιών

Όταν κάποιος, λοιπόν, αποκτήσει πρόσβαση στην υποδομή και συνδεθεί, είναι έτοιμος να υποβάλλει τις εργασίες του και να εκτελέσει τα πειράματα του με μια και μοναδική προϋπόθεση: να χρησιμοποιήσει τις εντολές όπως αυτές καθορίζονται από το λογισμικό gLite που ανήκει στην κατηγορία των DSL γλωσσών (Domain Scripting Language) και ονομάζεται jdl (Job Description Language). Γενικά, η JDL

είναι μια γλώσσα προγραμματισμού και έχει σχεδιαστεί για να καλύψει το συγκεκριμένο εύρος εφαρμογών σε αντίθεση με τις γλώσσες γενικού σκοπού όπως η C. Αφού, λοιπόν, ενεργοποιηθεί το πιστοποιητικό προσωρινά με διάρκεια 12 ώρες με την εντολή *voms-proxy-init --voms <VO>* η υποδομή είναι έτοιμη για χρήση. Κάθε εργασία που υποβάλλεται στο πλέγμα πρέπει να περιγράφεται από ένα .jdl αρχείο που καθορίζει ποιο αρχείο θα υποβληθεί, τι παραμέτρους έχει, ποια είναι τα αρχεία εισόδου και τι θα παραχθεί από την εκτέλεση της εργασίας. Ένα αρχείο .jdl στην απλοποιημένη του μορφή θα μπορούσε να είναι το run.jdl. Σε αυτό το παράδειγμα το αρχείο που θα εκτελεστεί είναι το run.exe και στην έξοδο παράγονται δύο αρχεία std.err και std.out. Το πρώτο καταγράφει αν υπήρξε κάποιο λάθος κατά την εκτέλεση της εργασίας και το δεύτερο περιέχει τα δεδομένα εξόδου.

Το αρχείο αυτό αποθηκεύεται στο UI και στην συνέχεια με την εντολή *glite-wms-job-list-match -a run.jdl* εμφανίζονται στην οθόνη όλα τα διαθέσιμα CEs που βρίσκονται στον εικονικό οργανισμό που προσδιορίζεται στην τελευταία γραμμή του .jdl αρχείου και από τις απαιτήσεις της εργασίας για την εκτέλεση της που περιγράφονται από το πεδίο Requirements (Δεν υπάρχει σε αυτό το παράδειγμα). Οι εντολές που χρησιμοποιούνται είναι οι ακόλουθες για υποβολή, παρακολούθηση της κατάστασης εκτέλεσης της εργασίας που υποβλήθηκε, ακύρωση και ανάκτηση της εξόδου. Η μεταβλητή jobID που εμφανίζεται δημιουργείται για την ταυτοποίηση της κάθε εργασίας ούτως ώστε κάθε ενέργεια που πραγματοποιείται να απευθύνεται στην κατάλληλη εργασία.

```
--------- run.jdl -----------
Executable = "run.exe";
Arguments = " ";
StdOutput = "std.out";
StdError = "std.err";
InputSandbox = {"./run.exe"};
OutputSandbox = {"std.err","std.out"};
VirtualOrganisation = "see";
```

- ✓ glite-wms-job-submit -o jobID -a run.jdl
- ✓ glite-wms-job-status -i jobID
- ✓ glite-wms-job-cancel -i jobID
- ✓ glite-wms-job-output -i jobID --dir ./

Εξ' ορισμού το πλέγμα προσφέρεται για την εκτέλεση πολλαπλών εργασιών (συλλογή εργασιών και παραμετρικές εργασίες) και παράλληλων εργασιών. Η πρώτη κατηγορία (job collection submission, parametric job submission) στην μεν περίπτωση της υποβολής μιας συλλογής εργασιών υποβάλλει ταυτόχρονα όλες τις εργασίες με την χρήση μιας και μόνο εντολής και στην περίπτωση των

παραμετρικών εργασιών υποβάλλεται ο ίδιος κώδικας με διαφορετικές παραμέτρους εισόδου όπως αυτές καθορίζονται στο αρχείο .jdl. Η δεύτερη κατηγορία περνά από το πεδίο της σειριακής εκτέλεσης στην παράλληλη επεξεργασία. Για την αποτελεσματική και ορθή υποβολή παράλληλων εργασιών χρησιμοποιείται το πρότυπο MPI (Message Passing Interface) που έχει σχεδιαστεί για την μεταφορά μηνυμάτων μεταξύ επεξεργαστών. Αυτό το είδος παράλληλης επεξεργασίας καλείται κατανεμημένη επεξεργασία όπως λεπτομερώς αναφέρθηκε στο Κεφάλαιο 2. Στην παράλληλη επεξεργασία, κάθε εργασία που πρόκειται να εκτελεστεί παράλληλα ταυτοποιείται με έναν αριθμό από 0 έως N-1, όπου N είναι ο αριθμός των επεξεργαστών ή κόμβων στον κόσμο του πλέγματος. Για να υποβληθεί μια παράλληλη εργασία είναι απαραίτητη η ύπαρξη τεσσάρων αρχείων, τα οποία είναι:

- ✓ Ο πηγαίος κώδικας ή το δυαδικό αρχείο που περιέχει τον κώδικα του MPI
- ✓ Ένα σενάριο MPI (mpi wrapper script) βασισμένο στην MPI υλοποίηση (παρέχεται)
- ✓ Το σενάριο που θα εκτελεστεί
- ✓ Το αρχείο.jdl

3.3.2 Διαχείριση Εργασιών

Στην προηγούμενη υποενότητα έγινε μια σύντομη περιγραφή της διαδικασίας υποβολής εργασιών στην υποδομή και άρα στην αξιοποίηση των CEs. Σε αυτή, παρουσιάζονται οι υπηρεσίες διαχείρισης των δεδομένων στο πλέγμα και ο ρόλος των SEs. Για να αποκτήσει κάποιος χρήστης πρόσβαση στις αποθηκευτικές υπηρεσίες, αφού συνδεθεί στην διεπαφή χρήστη – UI πρέπει να εκτελέσει 4 εντολές διαμόρφωσης:

- ✓ export LCG_CATALOG_TYPE=lfc
- ✓ export LFC_HOST=lfc.isabella.grnet.gr
- ✓ export LCG_GFAL_INFOSYS=bdii.isabella.grnet.gr:2170
- ✓ export LCG_GFAL_VO=see

Οι δύο πρώτες εντολές χρειάζονται για να είναι εφικτή η χρήση όλων των εντολών lfc-*, η τρίτη χρειάζεται από τον lcg διαχειριστή αντιγράφων και η τέταρτη για να μπορεί να παραληφθεί η παράμετρος –vo see κάθε φορά που εκτελείται μια εντολή lcg-*. Τα δεδομένα και, γενικά, τα αρχεία που αποθηκεύονται στο πλέγμα πρέπει με κάποιο τρόπο να ταυτοποιούνται ώστε να είναι εφικτή η αναζήτηση ή/και ανάκτηση τους. Υπάρχουν τρεις διαφορετικοί τρόποι ταυτοποίησης. Ο πρώτος γίνεται με τη χρήση ενός μοναδικού κωδικού Grid Unique Identifier (GUID) που είναι της μορφής:

guid:38ed3f60-c402-11d7-a6b0-f53ee5a37e1d

και παράγεται όταν ένα αρχείο αντιγράφεται απευθείας από έναν τοπικό δίσκο στο SE. Ο παραπάνω κωδικός που παράγεται δεν είναι εύκολος στην διαχείριση για αυτό το λόγο η χρήση λογικών ονομάτων Logical File Name (LFN) αποτελεί την καλύτερη αντιμετώπιση και έχει τη δομή ενός ιεραρχικά οργανωμένου καταλόγου. Ένα λογικό όνομα αναπαρίσταται ως εξής:

lfn:/grid/<MyVO>/<MyUsername>/<MyFile>

εδώ για παράδειγμα έστω ότι θέλω να αποθηκεύσω το αρχείο IM-0001-0004.dcm στον φάκελο 5 που δημιουργείται μέσα στον imagegrid στο πλέγμα,

lfn:/grid/see/imagegrid/5/IM-0001-0004.dcm

Τέλος, ο τρίτος τρόπος καλείται Physical File Name (SURL) και έχει τη μορφή ενός URL:

sfn://<hostname>/<Accesspoint>/<VO>/<filename>

Οι εντολές αλληλεπίδρασης διακρίνονται σε δύο κατηγορίες. Η LFC κατηγορία χρησιμοποιείται για την αρχικοποίηση και διαχείριση των μονοπατιών και καταλόγων ενώ η LCG έχει χρησιμότητα υψηλού επιπέδου κάνοντας εύκολη την αλληλεπίδραση με τα αποθηκευτικά στοιχεία του πλέγματος όπως αντιγραφή αρχείων από και προς το πλέγμα. Οι βασικές LFC και LCG εντολές παρουσιάζονται στους πίνακες 2 και 3.

Πίνακας 3: Οι εντολές LFC

lfc-mkdir	**Δημιουργία καταλόγου**
lfc-ls	**Εμφανίζει τα περιεχόμενα ενός καταλόγου**
lfc-rm	Αφαιρεί ένα αρχείο ή κατάλογο
lfc-chmod	**Αλλάζει τα δικαιώματα πρόσβασης**
lfc-rename	Μετονομάζει ένα αρχείο ή κατάλογο
lfc-setcomment	Προσθέτει ή αντικαθιστά ένα σχόλιο
lfc-delcomment	Διαγράφη ένα σχόλιο

Πίνακας 4: Οι εντολές LCG

lcg-cr	**Αντιγράφει ένα τοπίκο αρχείο σε ένα SE (upload)**
lcg-cp	**Αντιγράφει ένα αρχείο από κάποιο SE τοπικά (download)**
lcg-rep	**Δημιουργεί ένα αντίγραφο του αρχείου σε άλλο SE**
lcg-del	**Διαγράφει αντίγραφα ενός αρχείου**
lcg-aa	Δημιουργεί ένα νέο LFN για ένα αρχείο
lcg-la	Καταγράφει τα LFN ενός αρχείου
lcg-lr	Καταγράφει τα SURL(s) ενός αρχείου
lcg-lg	Καταγράφει τα GUID ενός αρχείου

Ο τελικός χρήστης μιας εφαρμογής δεν είναι υποχρεωμένος να γνωρίζει όλη την υποδομή και τον τρόπο που λειτουργεί αλλά τα οφέλη που προσφέρει. Για να επιτευχθεί αυτός ο στόχος, είναι απαραίτητη η δημιουργία μιας διεπαφής φιλικής και λειτουργικής προς το χρήστη. Η διαχείριση των αρχείων με τις lcg-* και lfc-* εντολές δεν θα είχε νόημα αν γινόταν ανεξάρτητα από μια εφαρμογή αδυνατώντας να αποκτήσει λειτουργική αξία. Για αυτό το λόγο, η αντιγραφή αρχείων DICOM από και προς το πλέγμα πραγματοποιείται ως τμήμα ενός συστήματος καταχώρησης ασθενών, ιατρών και εξετάσεων το οποίο περιγράφεται στη συνέχεια. Οι εντολές αυτές χρησιμοποιούνται σε πρώτη φάση γράφοντας τες απλά στη γραμμή εντολών του PUTTY. Το Putty είναι μια εφαρμογή ανοιχτού κώδικα που λειτουργεί ως client για SSH και Telnet πρωτόκολλα.

3.3.3 Οι Υπηρεσίες Ιστού (Web Services)

Ένα άλλο συστατικό στοιχείο της εφαρμογής είναι οι υπηρεσίες ιστού. Μια Υπηρεσία Ιστού ή όπως, κοινώς, αναφέρονται με τον όρο web services αποτελεί μια εφαρμογή προσβάσιμη μέσω δικτύου και μπορεί να εκτελεστεί σε ένα απομακρυσμένο σύστημα που φιλοξενεί τις υπηρεσίες που έχουν ζητηθεί. Οι web services τείνουν να χωριστούν σε δύο μεγάλες κατηγορίες: τις Μεγάλες Υπηρεσίες Ιστού (Big Web Services) και τις RESTful. Οι Web Services χρησιμοποιούν μηνύματα XML (eXtensive Markup Language) που ακολουθούν το πρότυπο SOAP (Simple Object Access Protocol) και χρειάζονται την περιγραφή της εφαρμογής γραμμένη σε μορφή WSDL αρχείου (Web Service Description Language). Το βασικό συστατικό μιας Web Service είναι η «μηχανή» Apache Axis2 η οποία αναλαμβάνει να μετατρέψει ένα πρόγραμμα που είναι γραμμένο σε μια γλώσσα προγραμματισμού σε Υπηρεσία Ιστού παράγοντας όλα τα απαραίτητα συστατικά. Λαμβάνοντας υπόψη ότι στην παρούσα υλοποίηση η εφαρμογή είναι γραμμένη σε Java για τη μετατροπή της σε Web Service χρησιμοποιούνται τα εργαλεία Axis2 Eclipse Plugins, που έχουν σχεδιαστεί ώστε να μπορούν να ενσωματωθούν στο Eclipse. Αυτά τα

εργαλεία αξιοποιούνται από τους προγραμματιστές για την μετατροπή ενός προγράμματος Java σε Web Service κάνοντας χρήση των αντίστοιχων οδηγών.

Τα εργαλεία που απαιτούνται για την υλοποίηση μιας Web Service είναι τα ακόλουθα (http://wso2.org/library/1719):

- ✓ Java Development Kit 1.4.2.x
- ✓ Eclipse SDK 3.2.x
- ✓ Apache Tomcat 4.1.x
- ✓ Axis2 Web Application
- ✓ Axis2 Eclipse Plugins

Οι δύο βασικοί τρόποι δημιουργίας μιας Web Service είναι η "Από κάτω προς τα πάνω"- Button Up προσέγγιση και η "Από πάνω προς τα κάτω" - Top Down προσέγγιση. Στην πρώτη περίπτωση, παίρνουμε ως είσοδο ένα αρχείο Java και εξάγεται ένα αρχείο .aar .ή jar χρησιμοποιώντας τον οδηγό: Apache Axis2 Service Archiver Generator Wizard – Eclipse Plugin. Στην δεύτερη περίπτωση χρησιμοποιείται ο οδηγός Apache Axis2 Code Generator Wizard – Eclipse Plugin που παράγει κώδικα WSDL2Java και Java2WSDL. Πιο αναλυτικά, για να μπορέσουμε να δημιουργήσουμε μια web service χρειάζεται να εγκατασταθούν τα δύο αυτά plugins μέσα στο φάκελο plugins του Eclipse. Πέρα από τους δύο αυτούς οδηγούς χρειάζεται το Axis2 για να αναπτύξει την υπηρεσία που έχει δημιουργηθεί. Στην περίπτωσή μας χρησιμοποιείται ένας Apache Tomcat, οπότε η εγκατάσταση του Axis2 γίνεται αντιγράφοντας το .aar αρχείο μέσα στον φάκελο webapp και κάνοντας επανεκκίνηση του εξυπηρετητή tomcat.

1. Προσέγγιση "Από κάτω προς τα πάνω" (Buttom – Up Approach)

Εφόσον όλα τα απαραίτητα εργαλεία έχουν εγκατασταθεί σωστά, ξεκινάει η διαδικασία δημιουργίας της υπηρεσίας. Πρώτα από όλα, δημιουργούμε ένα πρόγραμμα java στο Eclipse και το μεταγλωττίζουμε. Αν δεν υπάρχουν λάθη κατά τη μεταγλώττιση ξεκινάμε τον οδηγό Axis2 Service Archiver Plugin πατώντας Cltr+N. Στα παράθυρα που εμφανίζονται μπορούμε να αφήσουμε τις εξ' ορισμού (default) επιλογές ή να κάνουμε τις δικές μας. Όταν ολοκληρωθεί η διαδικασία, τυπώνεται στην οθόνη το μήνυμα: " Service Archive Generated Successfully", εισάγουμε το αρχείο .aar που θα δημιουργηθεί μέσα στο φάκελο `<TOMCAT_HOME>/webapp/axis2/WEB-INF/services` και κάνουμε επανεκκίνηση στο server. Καλώντας από το φυλλομετρητή τη σελίδα 195.251.6.234:8080/axis2 μπορούμε να δούμε τις εγκατεστημένες υπηρεσίες καθώς και ποιες συγκεκριμένες λειτουργίες είναι διαθέσιμες. Με την επανεκκίνηση δημιουργείται ένα αρχείο WSDL που περιέχει την περιγραφή της υπηρεσίας και βάσει του οποίου δημιουργείται ο κώδικας για τον client. Στην περίπτωση μας η

υπηρεσία καλείται μέσω ενός απλού προγράμματος γραμμένο σε php (οι υλοποιημένοι php-clients: RunLS.php, RunMakeDIR.php, RunLcgCr.php και RunLcgCp.php).

2. Προσέγγιση "Από πάνω προς τα κάτω" (Top – Down Approach)

Η δεύτερη προσέγγιση παίρνει ως είσοδο ένα αρχείο WSDL από το οποίο παράγεται ο σκελετός ενός προγράμματος Java το οποίο καλείται να ολοκληρώσει ο προγραμματιστής, δηλαδή υπάρχει η περιγραφή της εφαρμογής και από αυτήν δημιουργείται το πρόγραμμα που θα την καλέσει.

Για να επιστρέψουμε στην δημιουργία της υπηρεσίας ImageGridWS που πραγματοποιεί τη σύνδεση με το Grid, η λειτουργία του προγράμματος σε Java αποτελεί, ουσιαστικά, την εκτέλεση των εντολών commandline για την διαχείριση των εργασιών στο πλέγμα και ένα παράδειγμα είναι η εκτέλεση της εντολής που εμφανίζει τη λίστα με τα στοιχεία που υπάρχουν στον κατάλογο που ορίζεται από το ζόρισμα path και παρουσιάζεται στη συνέχεια.

```
public class ImageGridWS {

public String RunLS (String path) {
          String result = "";
          String temp = "";

           try {

                    Process p =
Runtime.getRuntime().exec("/home/glite/lcg/bin/lfc-ls
"+path);

                    BufferedReader stdInput = new
BufferedReader(new
InputStreamReader(p.getInputStream()));

                    BufferedReader stdError = new
BufferedReader(new
InputStreamReader(p.getErrorStream()));

                    // read the output from the command

                    System.out.println("Here is the
standard output of the command:\n");
```

```
                while ((temp = stdInput.readLine())
!= null) {
                    System.out.println(temp);
                  result+=temp +"\t\n";
                }

                // read any errors from the
attempted command

                System.out.println("Here is the
standard error of the command (if any):\n");
                while ((temp = stdError.readLine())
!= null) {
                    System.out.println(temp);
                  result+=temp;
                }
               }
               catch (Exception e) {
                System.out.println("exception
happened:" + e.toString());
                result = e.toString();
              e.printStackTrace();
                }

             return result;
         } //End method
…..................................................
..
         } //End class
```

Οπότε, η Web Service αποτελείται από τις λειτουργίες (195.251.6.234:8080/axis2/services/listServices):

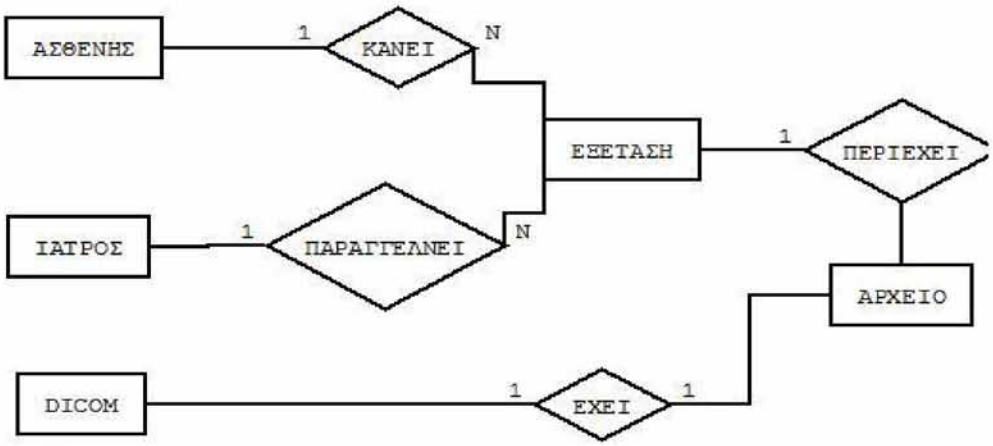
- ✓ RunLS
- ✓ RunMakeDIR
- ✓ RunLcgCr
- ✓ RunLcgCp
- ✓ RunLcgDel
- ✓ RunLcgRep
- ✓ RunLfcRm

Για να είναι εφικτή η πρόσβαση σε αυτές τις υπηρεσίες χρειάζεται από την πλευρά του πελάτη (client) ένα πρόγραμμα (Java, PHP, C++) που θα ζητά την υπηρεσία και θα κάνει χρήση αυτής. Εδώ, το ρόλο του client αναλαμβάνει η php που τελικά καλεί την επιθυμητή μέθοδο και καθορίζει τα ορίσματα του προγράμματος, όπως αναφέρθηκε νωρίτερα. Ένα παράδειγμα αποτελεί το αρχείο RunLS.php που ζητά την εκτέλεση της εντολής lfc-ls (Πίνακας 3) καλώντας μέσω της ImageGridWS την μέθοδο RunLS και δίνοντας την παράμετρο path.

```
<?php

require_once('/var/www/html/eLICO/lib/nusoap.php');

$wsdl="http://195.251.6.234:8080/axis2/services/ImageGridWS?wsdl";
$client=new nusoap_client($wsdl, true);

$param = array('path'=>"/grid/see/imagegrid");

$result = $client->call("RunLS", $param);

print_r($result);
```

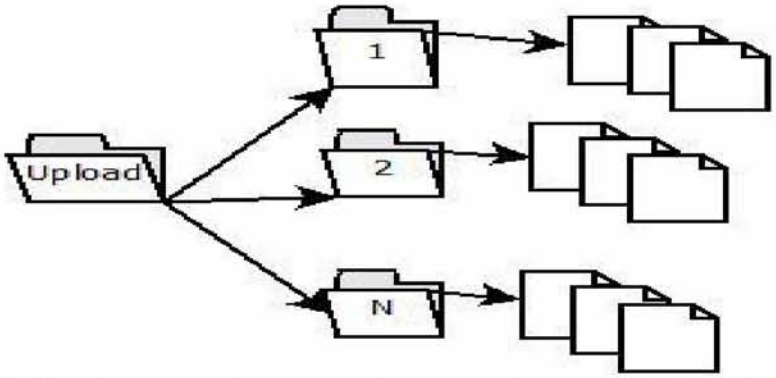
3.4 Υποσύστημα κατανεμημενης βάσης δεδομένων

Το επόμενο συστατικό στοιχείο της εφαρμογής είναι η βάση δεδομένων. Μια βάση δεδομένων αποτελεί μια ολοκληρωμένη συλλογή από συσχετισμένα δεδομένα. Για την υλοποίηση της εφαρμογής είναι απαραίτητη η δημιουργία της για να επικοινωνεί τόσο με την διεπαφή όσο και με το grid. Η βάση δεδομένων καλείται imagegrid και αποτελείται από έξι πίνακες. Πρώτα από όλα, οι πίνακες ΑΣΘΕΝΗΣ και ΙΑΤΡΟΣ αποθηκεύουν τα μεταδεδομένα των ασθενών και των ιατρών αντίστοιχα όπως παραδείγματος χάριν δημογραφικά στοιχεία, ο πίνακας ΕΞΕΤΑΣΗ περιέχει πληροφορίες για τα χαρακτηριστικά του περιστατικού ή της επίσκεψης, ποιος είναι ο χειριστής του μηχανήματος, την ημερομηνία της εξέτασης και τον αριθμό των αρχείων που περιλαμβάνει, ο πίνακας ΑΡΧΕΙΟ αποτελείται από τα μεταδεδομένα των αρχείων όπως ο τύπος ή το μέγεθος τους, η θέση στην οποία είναι αποθηκευμένα στο πλέγμα, και, τέλος, ο πίνακας DICOM κρατά στοιχεία από τα μεταδεδομένα της εικόνας που δεν έχουν αποθηκευτεί σε κάποιον από τους προηγούμενους πίνακες. Η βάση υλοποιήθηκε κάνοντας χρήση του phpMyAdmin 2.11.7 που βρίσκεται εγκατεστημένη στον διαδυκτιακό τόπο 195.251.6.234/phpmyadmin. Η σχηματική αναπαράσταση της βάσης παρουσιάζεται στην Εικόνα 15 και αποτελεί, ουσιαστικα, το μοντέλο οντοτήτων – συσχετίσεων της εφαρμογής. Η Εικόνα 16 δείχνει τον τρόπο με τον οποίο αποθηκεύονται τα δεδομένα τόσο στην τοπική βάση δεδομένων όσο και στο πλέγμα.

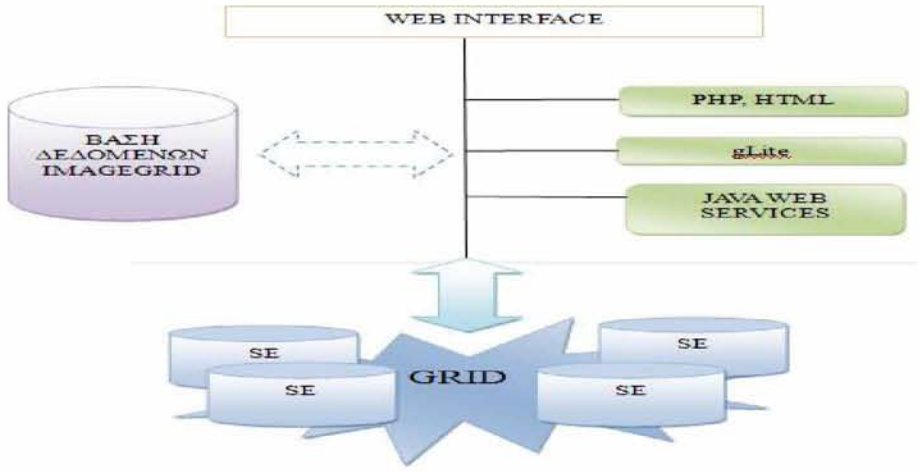


Εικόνα 15: Οι πίνακες της βάσης δεδομένων και οι μεταξύ τους συσχετίσεις.

Η σχεσιακή βάση δεδομένων ακολουθεί το παραπάνω διάγραμμα, δηλαδή η οντότητα εξέταση λειτουργεί ως πυλώνας με το οποίο συσχετίζονται οι υπόλοιπες οντότητες. Ένα ασθενής μπορεί να έχει κάνει 1:N εξετάσεις ή ένας ιατρός να έχει παραγγείλει 1:N εξετάσεις. Κάθε εξέταση μπορεί να αποτελείται από περισσότερα του ενός αρχεία ενώ καθένα αρχείο περιλαμβάνει στοιχεία DICOM. Στην περίπτωσή μας θεωρούμε ότι ο πίνακας DICOM συμμετέχει υποχρεωτικά στη συσχέτιση. Στην επόμενη ενότητα παρουσιάζονται αρκετά συνοπτικά κάποιες πληροφορίες για το πρότυπο DICOM καθώς, επίσης, και με ποιο τρόπο γίνεται η αξιοποίηση του από την εφαρμογή. Η συλλογή, ενημέρωση και επιλογή των στοιχείων από τη βάση γίνεται κάνοντας χρήση ερωτημάτων της SQL όπως αυτά ορίζονται από το συντακτικό της. Η PHP λειτουργεί ως ο συνδετικός κρίκος μεταξύ της διεπαφής και της βάσης για την συλλογή των στοιχείων από τις φόρμες και την πραγματοποίηση των αντίστοιχων λειτουργιών στη βάση. Επίσης, το σχεσιακό μοντέλο παρουσιάζει τα πεδία των πινάκων και τα κύρια και ξένα κλειδιά. Οι εντολές SQL που, κυρίως, χρησιμοποιούνται είναι οι: INSERT, SELECT και UPDATE. Επίσης, η βάση περιλαμβάνει και έναν πίνακα USERS που κρατά τα στοιχεία των χρηστών της εφαρμογής.



Εικόνα 16: Η δομή των φακέλων στον εξυπηρετητή και στο πλέγμα (στο πλέγμα ο φάκελος upload ονομάζεται imagegrid).



Εικόνα 17: Τα επιμέρους στοιχεία που συγκροτούν την εφαρμογή IMAGEGRID.

Σε αυτό το κεφάλαιο περιγράφηκε η αρχιτεκτονική του συστήματος και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της. Η Εικόνα 17 παρουσιάζει συνοπτικά όλα τα επιμέρους στοιχεία που χρησιμοποιήθηκαν στην ανάπτυξη αυτής της εφαρμογής. Στο παράρτημα Β υπάρχει ο πηγαίος κώδικας και τα αρχεία που χρησιμοποιούνται. Το κεφάλαιο 4 παρουσιάζεται η συνολική λειτουργία της εφαρμογής καθώς και οδηγίες χρήσης για αυτήν περιγράφοντας τις δυνατότητες που παρέχει.

4. ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΥΛΟΠΟΙΗΜΕΝΟΥ ΣΥΣΤΗΜΑΤΟΣ

Στα προηγούμενα κεφάλαια έγινε εισαγωγή στις τεχνολογίες που θα χρησιμοποιηθούν και σε εφαρμογές που έχουν ήδη υλοποιηθεί. Στο κεφάλαιο 3 παρουσιάστηκε η αρχιτεκτονική του συστήματος και ο τρόπος με τον οποίο συνδυαστήκαν όλα τα επιμέρους στοιχεία για να δώσουν το τελικό αποτέλεσμα. Η εφαρμογή είναι πλέον έτοιμη. Αυτό σημαίνει ότι οι χρήστες μπορούν να τη χρησιμοποιήσουν. Οποιαδήποτε εφαρμογή, όσο απλή κι αν είναι, συνοδεύεται από τις οδηγίες χρήσης που θα κάνουν πιο εύκολη τη χρήση της και θα μπορούν να βοηθήσουν στην προβολή και αξιοποίηση των δυνατοτήτων της βήμα προς βήμα. Σε αυτό το κεφάλαιο παρουσιάζεται η εφαρμογή μέσα από τα μάτια του τελικού χρήστη.

4.1 Είσοδος

Η πρώτη σελίδα που θα συναντήσει όταν κάποιος επισκεφθεί την εφαρμογή (http://195.251.6.234/ypapanti) είναι μια φόρμα όπου συμπληρώνει δύο πεδία username και password για να συνδεθεί κάνοντας στη συνέχεια login (Εικόνα 18). (Αν κάποιος δεν έχει λογαριασμό για να μπορεί να δημιουργήσει μια απλή καταχώρηση πατώντας στο σύνδεσμο REGISTER για να δημιουργήσει τα στοιχεία ταυτοποίησης του στη βάση δεδομένων. Αυτή η δυνατότητα είναι προαιρετική και υπάρχει για την πραγματοποίηση δοκιμών, καθώς οι χρήστες ενός τέτοιου συστήματος πρέπει να είναι πλήρως ταυτοποιημένοι.)

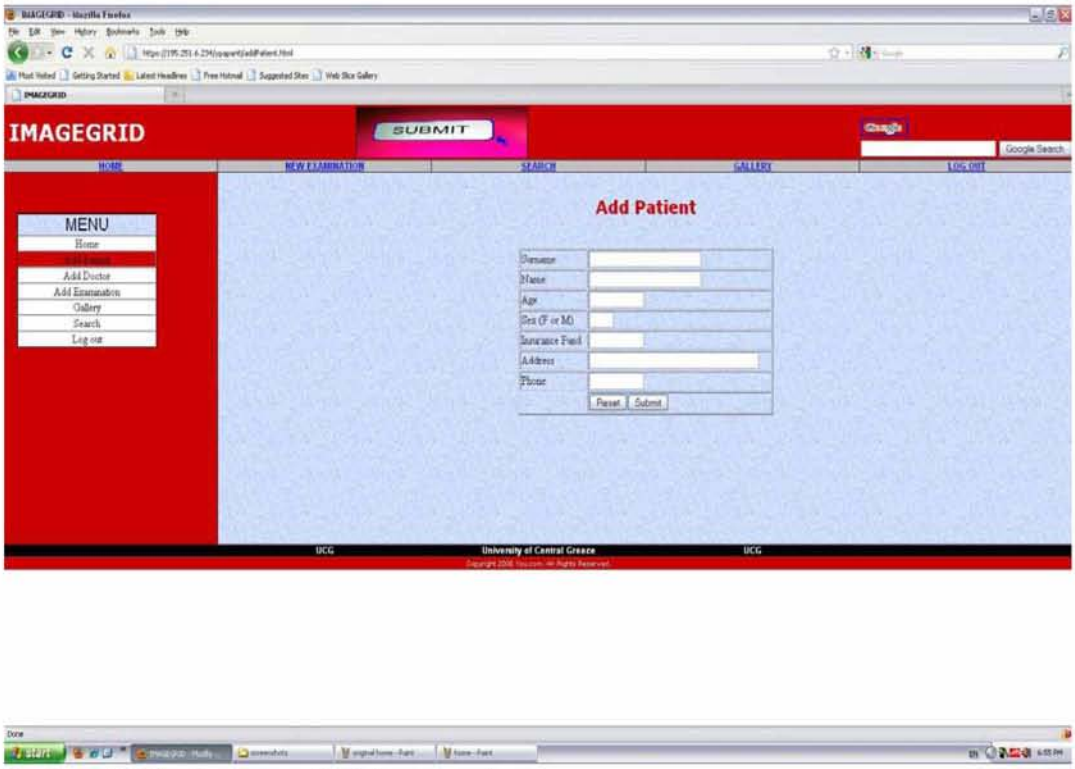
Εικόνα 18: Η πρώτες σελίδες που συναντά ο χρήστης.

Εφόσον τα στοιχεία είναι σωστά ο χρήστης βρίσκεται στην αρχική σελίδα που υπάρχουν μερικές πληροφορίες για την εφαρμογή και στα αριστερά της οθόνης βρίσκεται το μενού επιλογών. Υπάρχουν 7 επιλογές από τις οποίες μπορεί να επιλέξει. Οι επιλογές αυτές είναι:

- Home (Αρχική Σελίδα)
- Add patient (Προσθήκη Ασθενή)
- Add doctor (Προσθήκη Ιατρού)
- Add examination (Προσθήκη Εξέτασης)
- Gallery (Συλλογή εικόνων)
- Search (Αναζήτηση)
- Logout (Αποσύνδεση)

4.2 Καταχώρηση στοιχείων

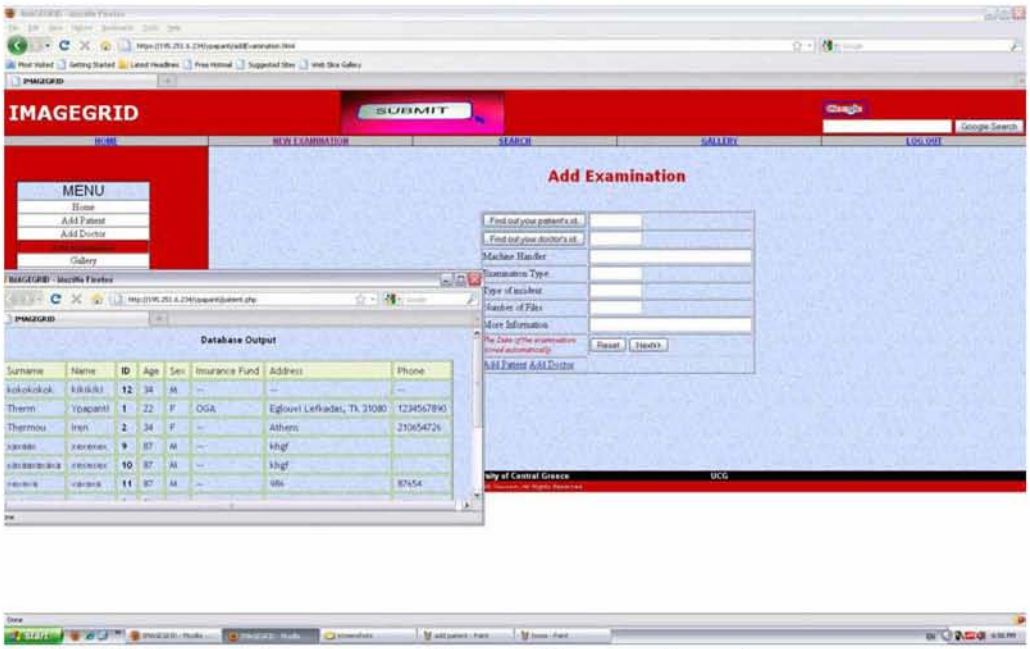
Αν ο χρήστης επιλέξει από το μενού επιλογών την προσθήκη ασθενή ή την προσθήκη ιατρού θα εμφανιστεί μια φόρμα που θα πρέπει να συμπληρωθεί. Και στις δύο περιπτώσεις υπάρχουν κάποια πεδία που η συμπλήρωσή τους είναι υποχρεωτική και αυτό επισημαίνεται σε περίπτωση που μείνουν κενά. Όταν η φόρμα συμπληρωθεί, τα στοιχεία της υποβάλλονται στη βάση πατώντας το πλήκτρο Submit αφού προηγηθεί έλεγχος για διπλοεγγραφή. Αν όλα τα στοιχεία είναι ίδια σημαίνει ότι το συγκεκριμένο άτομο είναι ήδη καταχωρημένο οπότε δεν χρειάζεται να εγγραφεί ξανά. Για κάθε νέα καταχώρηση που πραγματοποιείται δίνεται ένας μοναδικός κωδικός ο οποίος δημιουργείται αυτόματα από το σύστημα. Τα πεδία που συμπληρώνονται είναι το ονοματεπώνυμο του ασθενή, το φύλο, η ηλικία του, το ασφαλιστικό του ταμείο και στοιχεία επικοινωνίας όπως διεύθυνση και τηλέφωνο. Με ανάλογο τρόπο λειτουργεί και η καταχώρηση ενός ιατρού.



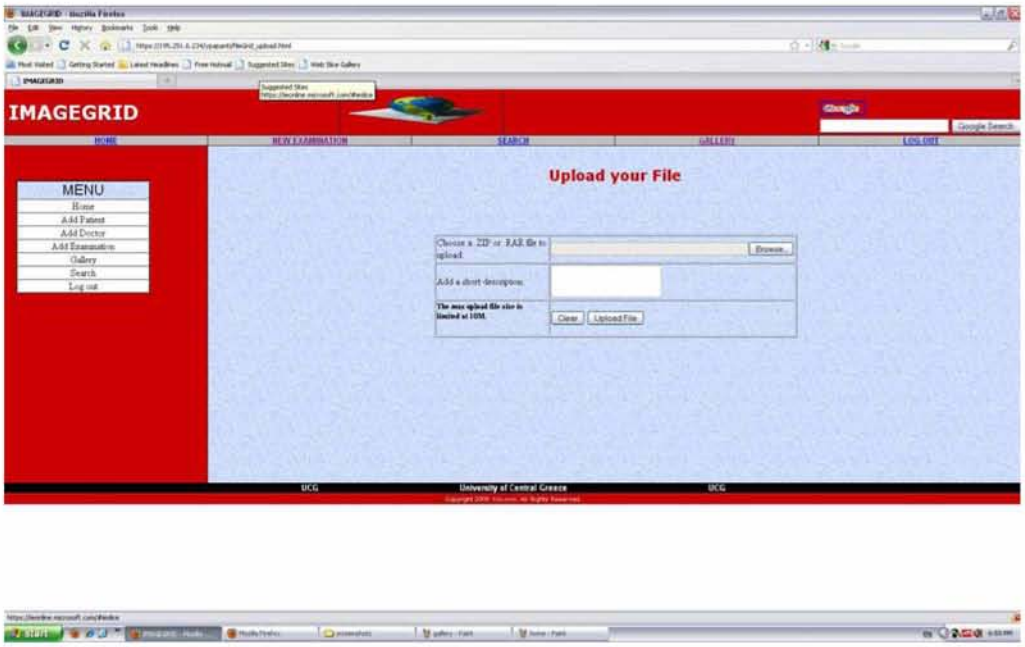
Εικόνα 19 Η φόρμα υποβολής στοιχείων.

Πέρα από την καταχώρηση στοιχείων των παραπάνω στοιχείων, ο κεντρικός άξονας λειτουργίας είναι η εκτέλεση εξετάσεων. Κάθε εξέταση που πραγματοποιείται συνδέει έναν ασθενή με τον γιατρό που θα παραγγείλει την εξέταση οπότε χρειάζεται να καταχωρηθεί ο κωδικός τους. Για το λόγο αυτό υπάρχουν δύο κουμπιά ("Find Patient id", "Find Doctor id") που εμφανίζουν στην οθόνη ένα νέο παράθυρο με όλες τις εγγραφές σε αλφαβητική σειρά και τονισμένο τον κωδικό ID. Όταν, λοιπόν, βρεθεί ο ασθενής που επιθυμούμε πληκτρολογούμε τον κωδικό του στο αντίστοιχο πεδίο και συνεχίζουμε με την υπόλοιπη φόρμα. Το ίδιο μπορεί να συμβεί και στην περίπτωση του ιατρού. Ζητούνται, επίσης, τα πεδία με το όνομα του χειριστή του μηχανήματος (που θα μπορούσε να είναι ο χρήστης της

εφαρμογής), ο τύπος της εξέτασης (CT, MRI, κλπ), το είδος του περιστατικού (πχ τροχαίο ατύχημα), ο αριθμός των αρχείων και οποιαδήποτε σχόλια ή πληροφορίες θεωρούνται σημαντικές για την ολοκληρωμένη εικόνα της εξέτασης. Η ημερομηνία της εξέτασης καταχωρείται αυτόματα μαζί με τα υπόλοιπα στη βάση δεδομένων. Όταν συμπληρωθεί η φόρμα και καταχωρηθούν τα στοιχεία εμφανίζεται μια ακόμη τελευταία φόρμα που αποτελείται από δύο πεδία μόνο. Το πρώτο χρησιμοποιείται για να βρούμε το αρχείο που θέλουμε να αποθηκεύσουμε και το δεύτερο υπάρχει για μια σύντομη προαιρετική περιγραφή του αρχείο. Όταν η αντιγραφή του αρχείου ολοκληρωθεί εμφανίζεται για μικρό χρονικό διάστημα μια σελίδα με τα στοιχεία του αρχείου που μόλις αποθηκεύτηκε και την πληροφορία αν τα στοιχεία από την επικεφαλίδα DICOM έχουν εξαχθεί και αποθηκευτεί σωστά ("DICOM Done!"). Στη συνέχεια, η εφαρμογή επιστρέφει στην φόρμα για το "ανέβασμα" αρχείων από όπου μπορούν να καταχωρηθούν όσα αρχεία περιλαμβάνει η εξέταση. Σε αυτό το σημείο ολοκληρώνεται η καταχώρηση των στοιχείων.



Εικόνα 20: Η καταχώρηση μιας εξέτασης δίνει την δυνατότητα ανάκτησης των κωδικών ασθενών και ιατρών.



Εικόνα 21: Η φόρμα μέσω της οποίας αντιγράφονται τα αρχεία (συμπιεσμένο την πρώτη φορά με όλο το περιεχόμενο της εξέτασης και ένα απλό DICOM αρχείο τη δεύτερη).

4.3 Παρουσίαση δεδομένων

Πατώντας την επιλογή "Gallery" από το μενού επιλογών εμφανίζεται η ακόλουθη οθόνη δείχνοντας την μικρογραφία κάθε αρχείου που είναι αποθηκευμενό. Κάτω από κάθε εικόνα εμφανίζεται το όνομα του αρχείου και δύο σύνδεσμοι: View Details και Download Here. Πατώντας στην πρώτη επιλογή εμφανίζονται τα στοιχεία της εικόνας δηλαδή πέρα από τα τεχνικά χαρακτηριστικά της όπως ο κωδικός της, ο τύπος του αρχείου, το μέγεθος της και σε ποιο SE είναι αποθηκευμένη, τυπώνονται τα στοιχεία της επικεφαλίδας DICOM. Κάποια από τα χαρακτηριστικά αυτά είναι τα ακόλουθα:

- ✓ Bits stored: δηλώνει τον αριθμό των bits που αναπαριστούν κάθε pixel
- ✓ Patient Position: απαιτείται για εικόνες CT και MR και δηλώνει την θέση του ασθενή
  - HFP= head first-prone
  - HFS=head first-supine
  - HFDR= head first-decibitus right
  - HFDL = head first-decubiturs left
  - FFP = feet first-prone
  - FFS, FFDR, FFDL

- ✓ Photometric Interpretation: δηλώνει τον τρόπο που απεικονίζονται τα δεδομένα

- MONOCHROME1= low values=bright, high values=dim
- MONOCHROME2= low values=dark, high values=bright
- RGB, CMYK
- Οι δύο πρώτες αναπαραστάσεις αναφέρονται κυρίως σε εικόνες CT ή MR ενώ οι επόμενοι, κυρίως, σε υπέρηχους ή ιατρικές φωτογραφίες.

✓ Slice Thickness: δηλώνει το πάχος της τομής

✓ Transfer Syntax UID: αναφέρεται στην δομή των δεδομένων και προσδιορίζει αν έχει γίνει συμπίεση στην εικόνα ή όχι.

Εικόνα 22: Η γενική απεικόνιση των στοιχείων και οι πρόσθετες πληροφορίες που παρέχονται.

Σε περίπτωση που κάποιο αρχείο δεν έχει απεικονιστεί σωστά (πιο συγκεκριμένα αν η εικόνα αποτελείται από δύο περιοχές, μια λευκή και μια μαύρη, σημαίνει ότι υπήρξε σφάλμα κατά τη μετατροπή της σε JPEG από τον οδηγό πιθανόν λόγω έλλειψης στοιχείων αλλά αν ανακτηθεί εμφανίζεται κανονικά σε DICOM viewers).

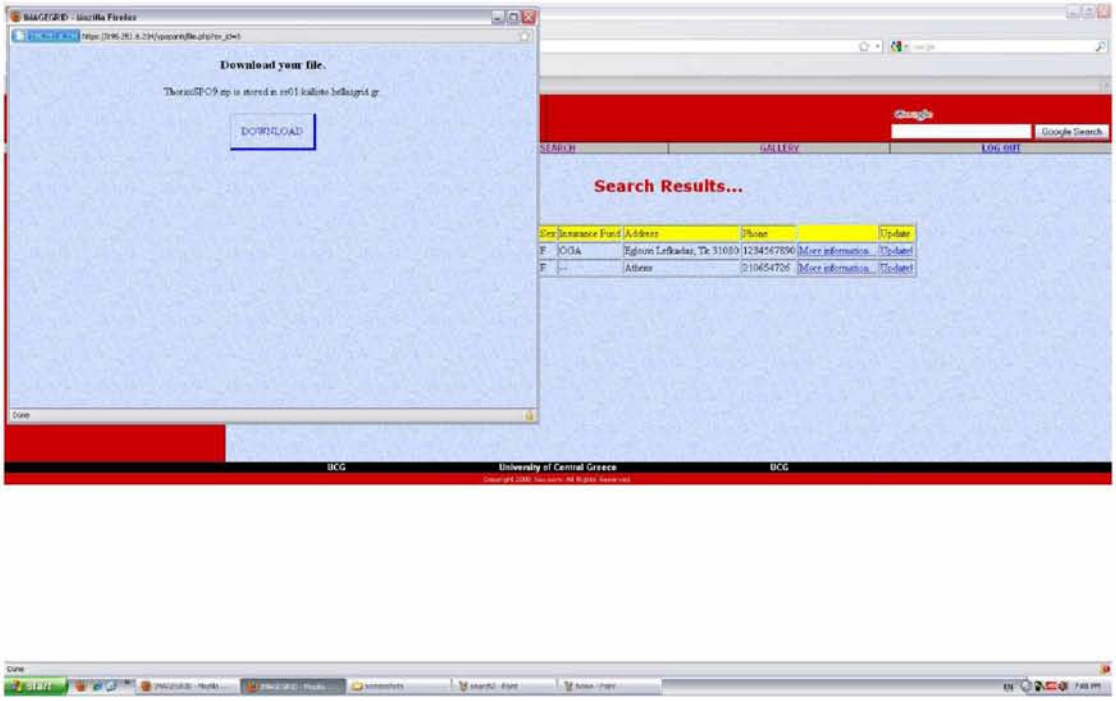
4.4 Αναζήτηση

Μια άλλη ενότητα της εφαρμογής είναι η αναζήτηση. Ο χρήστης συμπληρώνει την φόρμα που φαίνεται στην εικόνα επιλέγοντας αν θέλει να κάνει αναζήτηση για ασθενή ή γιατρό και αυτό που εμφανίζεται είναι το αποτέλεσμα της αναζήτησης και είναι αυτό που παρουσιάζεται στο πλαίσιο. Ο χρήστης μπορεί να επιλέξει την παρουσίαση περισσότερων πληροφοριών σχετικά με τις εξετάσεις του ασθενή (ή γιατρού) που επιθυμεί ή την ενημέρωση των στοιχείων μιας ήδη καταχωρημένης εγγραφής.

Εικόνα 23: Η αναζήτηση πραγματοποιείται μέσω της παραπάνω φόρμας και εμφανίζονται τα πρώτα αποτελέσματα.

Πατώντας αυτό το σύνδεσμο εμφανίζεται στην οθόνη η παρακάτω διάταξη. Οι επιπλέον πληροφορίες αφορούν το γιατρό που παρήγγειλε την εξέταση, καθώς και στοιχεία της εξέτασης, όπως τον χειριστή του μηχανήματος, την ημερομηνία της εξέτασης ή το είδος του περιστατικού. Μετά από αυτές τις πληροφορίες εμφανίζεται μια λίστα με τα αρχεία των εξετάσεων όπως παρουσιάζεται στην Εικόνα 24. Τέλος, στην εικόνα 25, αίνεται ο σύνδεσμος που δημιουργείται για τη λήψη του αρχείου.

Εικόνα 24: Οι εξετάσεις ενός συγκεκριμένου ασθενή και η λίστα με τα αρχεία του τα οποία μπορεί να λάβει τοπικά.



Εικόνα 25: Η λήψη ενός συγκεκριμένου αρχείου.

5. ΣΥΖΗΤΗΣΗ - ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι σύγχρονες ανάγκες επιτάσσουν την αναζήτηση σύγχρονων μορφών αντιμετώπισης με κεντρικό άξονα την βελτιστοποίηση της απόδοσης, ελαχιστοποιώντας το κόστος. Το πλέγμα παρουσιάζεται ως ένα πολλά υποσχόμενο εργαλείο για την διαχείριση και επεξεργασία ιατρικών δεδομένων. Οι έννοιες του κατανεμημένου και παράλληλου υπολογισμού είναι άμεσα συνδεδεμένες με το πλέγμα. Σε αυτή την εργασία μελετήθηκαν και αξιοποιήθηκαν κάποιες από τις βασικές λειτουργίες του πλέγματος αναφορικά με τη διαχείριση ιατρικών εικόνων άρα μιας «κατανεμημένης βάσης δεδομένων». Οι αποθηκευτικές μονάδες του πλέγματος (SE) μπορούν να λειτουργήσουν ως μια «τεράστια» βάση δεδομένων παρέχοντας ένα ιδιαίτερα ασφαλές περιβάλλον στους χρήστες του. Το πρώτο, λοιπόν, βήμα αποτελεί η αξιοποίηση των αποθηκευτικών μονάδων και των κανόνων διαχείρισης αυτών των δεδομένων, με ασφαλή τρόπο που εξασφαλίζεται με ταυτοποίηση των χρηστών μέσα από ψηφιακά πιστοποιητικά. Η αξιολόγηση ενός συστήματος πρέπει να γίνεται τόσο από τους σχεδιαστές και κατασκευαστές ενός συστήματος όσο και από τους χρήστες του. Κάποιες από τις βασικές ιδιότητες που οφείλει να έχει ένα σύστημα είναι αξιοπιστία, αποτελεσματικότητα, ακεραιότητα, ευχρηστία, ευκολία συντήρησης, ευελιξία και ευκολία δοκιμής, μεταξύ πολλών άλλων, που κάθε μια μεταφράζεται σε επιμέρους στόχους [17]. Στόχοι όπως αποτελεσματικότητα εκτέλεσης, αποδοτική διαχείριση, έλεγχο πρόσβασης, απλότητα, επεκτασιμότητα, είναι μερικοί που έχουν επιτευχθεί από την εφαρμογή IMAGEGRID.

Σε οποιοδήποτε υπό μελέτη πρόβλημα ακόμα κι αν θεωρητικά η λύση του είναι γνωστή η πορεία για την επίλυση του μπορεί να περιέχει εμπόδια και δυσκολίες. Μια από τις βασικές δυσκολίες κατά την υλοποίηση αυτής της εργασίας ήταν η εύρεση του κατάλληλου τρόπου σύνδεσης και επικοινωνίας με το πλέγμα. Δοκιμάστηκαν διάφοροι τρόποι όπως η απευθείας εκτέλεση τους μέσα από κώδικα PHP ή την κλήση ενός προγράμματος Java. Τελικά ο τρόπος με το οποίο έγινε η σύνδεση βασίζεται στις υπηρεσίες ιστού (Web Services) περνώντας όλες τις χρήσιμες παραμέτρους και αποτελεί τον βέλτιστο τρόπο υλοποίησης. Επίσης, η αναζήτηση και τελική επιλογή του καταλλήλου API για την εξαγωγή και μετατροπή του DICOM αρχείου καθώς δοκιμάστηκαν διάφορα εργαλεία χειρισμού των DICOM αρχείων αποτέλεσαν μια χρονοβόρα διαδικασία.

Οι μελλοντικές προεκτάσεις μιας τέτοιας υλοποίησης είναι αρκετές. Μια από αυτές είναι η άμεση σύνδεση των μηχανημάτων σάρωσης και παραγωγής εικόνων με το πλέγμα αντιμετωπίζοντας πρωτίστως τις πιθανές απειλές μέσω δικτύου χρησιμοποιώντας κτυπτογραφικές μεθόδους. Οπότε, το μόνο που θα βρίσκεται μεταξύ τους θα είναι το Ραδιολογικό Πληροφοριακό Σύστημα. Ένα δεύτερο βήμα είναι η χρησιμοποίηση των υπολογιστικών στοιχείων (CE) για επεξεργασία και ανάλυση εικόνων που μπορούν να λειτουργήσουν υποστηρικτικά της διάγνωσης υλοποιώντας με κατανεμημένο τρόπο αλγορίθμους βελτίωσης των εικόνων όπως απομάκρυνση θορύβου και εξαγωγής χαρακτηριστικών μέσω φιλτραρίσματος. Από τη βιβλιογραφία παρατηρούμε ότι υπάρχουν αρκετές εφαρμογές παγκοσμίως που προσανατολίζονται προς αυτή την κατεύθυνση υλοποιώντας πλήθος αλγορίθμων σχεδιάζοντας συστήματα ανίχνευσης παθολογικών δομών ή μη φυσιολογικών

προτύπων και ενσωματώνουν εργαλεία οπτικοποίησης των αποτελεσμάτων ανάλυσης και ανακατασκευής τρισδιάστατων δομών.

ΑΝΑΦΟΡΕΣ

[1]. Michal Vossberg, Thomas Tolxdorff, and Dagmar Krefting, DICOM Image Communication in Globus-Based Medical Grids, Institute of Medical Informatics, Charite, Universitatsmedizin Berlin, 2007, IEEE

[2]. Ian Foster, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, 1st International Symposium on Cluster Computing and the Grid (CCGRID '01), 2001, IEEE

[3]. J. Montagnat (johan@creatis.insa-lyon.fr) , H.Duque(duque@creatis.insa-lyon.fr) DataGrid: Medical Data Storage and Retrieval on the DataGrid, IST, August 22, 2002

[4]. Chun-Hsi Huang, Akihiko Konagaya, Vincenzo Lanza, Peter M. A. Sloot, Guest Editorial Introduction to the Special Section on BioGrid: Biomedical Computations on the Grid, IEEE Trancactions on Information Tachnology in Biomedicine, vol.12, no. 2, March 2008

[5]. Michael Creel (michael.cree@uab.es) and William L. Goffe (goff@oswego.edu), Multi- core, Clusters, and Grid Computing: a Tutorial, November, 2007 Blaise Barney, Introduction to Parallel Computing, Lawrence Livermore National Laboratory

[6]. Maglogiannis, I., Soldatos, J., Chatzioannou, A., Milonakis, V., Kanaris, Y., 2007, in IFIP International Federation for Information Processing, Volume 247, Artificial Intelligence and Innovations 2007: From Theory to Applications, eds. Boukis, C., Pnevmatikakis, L., (Boston: Springer), pp. 117-126

[7]. I. Foster, C. Kesselman, J. M. Nick, S. Tuecke, The Physiology of the Grid, An Open Grid Services Architecture Systems Integration, www.globus.org/research/papers/ogsa.pdf

[8]. Ivan De Mitri, The MAGIC-5 Project: Medical Applications on a Grid Infrastructure Connection, On Behalf of the MAGIC-5 Collaboration

[9]. M. Bertero, P. Bonetto, L. Carracciuolo, L. D' Amore, A. Formiconi, M. R. Guarraccino, G. Laccetti, A. Murli, MedIGrid: a Medical Imaging application for computational Grids, IEEE, 2003

[10]. C. Germain, V. Breton, P. Clarysse, Y. Gaudeau, T. Glatard, E. Jeannot, Y. Legre, C. Loomis, I.Magnin, J. Montagnat, J. –M. Moureaux, A. Osorio, X. Pennec, and R. Texier, GRID-ENABLING MEDICAL IMAGE ANALYSIS, Journal of clinical Monitoring and Computing (2005) 19: 339-349

[11]. http://wiki.hellasgrid.gr/wiki/bin/view/HellasGrid/GOC/AccessHellasGrid

[12]. http://www.gridcafe.org

[13]. http://gridbus.cs.mu.oz.au/~raj/

[14]. medical.nema.net

[15]. Ιωάννης Μανωλόπουλος, Απόστολος Ν. Παπαδόπουλος, ΣΥΣΤΗΜΑΤΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ – Θεωρία & Πρακτική Εφαρμογή, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, Έκδοση 1η, Copyright © 2006

[16]. Δουληγέρης Χρήστος, Μαυρτοπόδη Ρόζα, Κοπανάκη Ευη, Τεχνολογίες Διαδικτύου – αρχές λειτουργίας & προγραμματισμός εφαρμογών στο Διαδίκτυο, Εκδόσεις Νηρηίδες, Β΄ Έκδοση, 2004

[17]. Shari Lawrence Pfleeger, (Επιμέλεια Ελληνικής έκδοσης Γιάννης Σταμέλος), ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ – ΘΕΩΡΙΑ ΚΑΙ ΠΡΑΞΗ, Εκδόσεις Κλειδάριθμος, Β' Έκδοση, 2007

[18]. -John Mantas, Arie Hasman, Πληροφορική της υγείας – Νοσηλευτική προσέγγιση, Ιατρικές εκδόσιες Π.Χ. Πασχαλίδης, 2007

[19]. -Ταξιάρχης Μπότσης, Στέλιος Χαλκιώτης, Πληροφορική Υγείας, Η εφαρμογή της πληροφορικής στο χώρο της υγείας, Εκδόσεις Δίαυλος, Αθήνα 2005

[20]. -Ιωάννης Αποστολάκης, Πληροφοριακά Συστήματα Υγείας, 2η βελτιωμένη έκδοση, Εκδόσεις Παπαζήση, Αθήνα 2007

[21]. Ingrid M. Keseler, Julio Collado-Vides, Socorro Gama-Castro, John Ingraham, Suzanne Paley, Ian T. Paulsen, Martín Peralta-Gil and Peter D. Karp, EcoCyc: a comprehensive database resource for Escherichia coli, Nucleic Acids Research, 2005, Vol. 33, Database issue D334-D337

[22]. Irina Strizh, Alexei Joutchkov, Nikolay Tverdokhlebov and Sergey Golitsyn, Systems Biology and grid technologies: Challenges for understanding complex cell signaling networks, Future Generation Computer Systems Volume 23, Issue 3, March 2007, Pages 428-434

[23]. P Kohl, D Noble, RL Winslow, PJ Hunter, computational modeling of Biological systems: tools and visions, The royal society, 2000

[24]. Jiawei Han, Micheline Kamber, Data mining – Concepts and Techniques, 2nd edition – 2006

[25]. Tsiknakis M., Brochhausen M., Nabrzyski J., Pucacki J., Sfakianakis SG., Potamias G., Desmedt C., Kafetzopoulos D., A semantic grid infrastructure enabling integrated access and analysis of multilevel biomedical data in support of postgenomic clinical trials on cancer, IEEE Trans Inf Technol Biomed. 2008 Mar;12(2):205-17.

ΠΑΡΑΡΤΗΜΑ Α: Ευρετήριο όρων

API	Application Programming Interface	Μια διεπαφή υλοποιημένη σε κάποια γλώσσα προγραμματισμού, ικανή να αλληλεπιδρά με άλλο λογισμικό.
CE	Computing Element	Μονάδα επεξεργασίας που μπορεί να χρησιμοποιηθεί από τους χρήστες του πλέγματος
Cluster		Ομάδα υπολογιστών – επεξεργαστών
DICOM	Digital Imaging and Communication in Medicine	Πρότυπο για την διαχείριση και ανταλλαγή ιατρικών εικόνων.
DSL	Domain Specific Language	Γλώσσα προγραμματισμού σχεδιασμένη για μια εξειδικευμένη ομάδα εντολών.
EGEE	Enabling Grids for E-sciencE	Είναι το ευρωπαϊκό πρόγραμμα για την προώθηση των έργων που βασίζονται στο πλέγμα για την Ευρώπη.
gLite		Το λογισμικό που χρησιμοποιείται για την υποβολή και διαχείριση εργασιών και δεδομένων στο πλέγμα.
Grid		Ένα σύνολο επεξεργαστών και αποθηκευτικών μονάδων γεωγραφικά κατανεμημένων με κοινές αρχές και κανόνες.
Grid Computing		Ο συνδυασμός υπολογιστικών πηγών για την επίλυση προβλημάτων.
Grid Infrastructure		Περιλαμβάνει τον συνδυασμό ενός αριθμού δυνατοτήτων και πηγών σε ένα συγκεκριμένο περιβάλλον.
GUID	Grid Unique IDentifier	Μια συμβολοσειρά από 16 χαρακτήρες που προσδιορίζει μοναδικά κάθε αρχείο που αντιγράφεται στο πλέγμα.
JDL	Job Description Language	Η γλώσσα που περιγράφει τις εργασίες που στέλνονται στο πλέγμα.
LCG		Εντολές υψηλού επιπέδου για την αλληλεπίδραση των χρηστών με τα SEs.
LFC	Logic File Catalogue	Εντολές χαμηλού επιπέδου για την αλληλεπίδραση των χρηστών με τα SEs.
Middleware	Μεσεισμικό	Το ενδιάμεσο λογισμικό

MPI	Message Passing Interface	Βιβλιοθήκη –πρότυπο για την δημιουργία παράλληλων προγραμμάτων
PACS	Picture Archiving and Communication System	Αποτελεί συνδυασμό υλικού και λογισμικού για την αποθήκευση, ανάκτηση, διαχείριση, κατανομή και παρουσίαση ιατρικών εικόνων
SE	Storage Element	Αποθηκευτική μονάδα που μπορεί να χρησιμοποιηθεί από τους χρήστες του πλέγματος
SEE	South Eastern Europe	Είναι ο εικονικός οργανισμός για τους χρήστες της ΝΑ Ευρώπης.
SOAP	Simple Object Application Protocol	Πρωτόκολλο για την ανταλλαγή δομημένων πληροφοριών στην υλοποίηση web services.
SURL	Storage URL	Αναπαριστά το φυσικό όνομα ενός αρχείου που αποθηκεύεται στο πλέγμα και έχει τη μορφή ενός URL
UI	User Interface	Είναι ένα σύστημα μέσω του οποίου ένας χρήσης αλληλεπιδρά με την μηχανή.
VO	Virtual Organization	Ένας εικονικός οργανισμός καθορίζει τους κανόνες βάσει των οποίων λειτουργούν οι χρήστες του.
Web Service		API με πρόσβαση μέσω διαδικτύου που εκτελούν απομακρυσμένα μια λειτουργία που ζητείται.
WMS	Workload Manager System	Το σύστημα διαχείρισης των εργασιών που υποβάλλονται στο πλέγμα.
WSDL	Web Service Description Language	Γλώσσα προγραμματισμού για την περιγραφή υπηρεσιών ιστού.
XML	EXtensible Markup Language	Μια γλώσσα σήμανσης που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων.

ΠΑΡΑΡΤΗΜΑ Β: Κώδικας

Στις σελίδες που ακολουθούν παρουσιάζονται ενδεικτικά οι λειτουργίες της εφαρμογής όπως έχουν υλοποιηθεί από την php.

config.php

```
<?php

$server = 'localhost';
$username = 'ypapanti';
$password = 'ypapanti2009';
$database = "imagegrid";

?>
```

Login.php

Δίνει πρόσβαση μόνο σε ταυτοποιημένους χρήστες.

```
<?php

session_start();

?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<?php

include 'config.php';

$username_field=$_POST['username'];
$password_field=$_POST['password'];

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$query = "SELECT username, password FROM `USERS` WHERE
username='".$username_field."' AND password='".$password_field."'";
$result = mysql_query($query, $link);
$num = mysql_numrows($result);

if($num==0){
  echo "Incorrect username or password...";
  echo "<br>";
  echo "Please try again.";
  include ("login.html");
}
```

```
else{
  include ("Menu.html");
}
//echo $num;
if(!$result){
       die('Can\'t view elements: ' . mysql_error());
}


if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }

mysql_close();
?>

</body>
</html>
```

register.php

Για να αποκτήσει κάποιος πρόσβαση στην εφαρμογή αρκεί να δημιουργήσει έναν λογαριασμό μέσω του παρακάτω κώδικα. Η δυνατότητα αυτή παρέχεται κυρίως για δοκιμαστικούς λόγους.

```
<?php

include 'config.php';

function is_alphachar($text) {

    for ($i = 0; $i < strlen($text); $i++) {

       if (!ereg("[A-Za-z0-9]", $text[$i])) {
              return 1;
       }
    }
    }

echo "<table border=\"0\" width=\"300\" align=\"center\">
<tr><td width=\"300\">
<h3>Register for a new account.</h3>
<form action=\"register.php\" method=\"POST\">
</td></tr><tr><td>
Username: <br><input type=\"text\" name=\"username\" ><br>
Password: <br><input type=\"password\" name=\"password\"><br>
E-mail: <br><input type=\"text\" name=\"email\" ><br><br>
<input type=\"submit\" value=\"Create!\">
</form>
<br><br><br><br><br>
</td></tr>
</td></tr>

";

if($_POST[username] == ""){
echo $form;
} elseif(strlen($_POST[password]) < 6){
echo $form;
echo "<br> Error password must be 6 characters or more";
} else {

       $link = mysql_connect($server, $username, $password);
              if (!$link) {
                  die('Not connected : ' . mysql_error());
       }
```

```
        // make imagegrid the current db
        $db_selected = mysql_select_db($database, $link);
        if (!$db_selected) {
            die ('Can\'t use imagegrid : ' . mysql_error());
        }


$sql = "SELECT username FROM `USERS`
        WHERE username = '$_POST[username]'";

$sql2 = "SELECT email FROM `USERS`
        WHERE email = '$_POST[email]'";

$result = mysql_query($sql)
        or die ("Couldn't execute query.");

$result2 = mysql_query($sql2)
        or die ("Couldn't execute query.");

$num = mysql_num_rows($result);
$num2 = mysql_num_rows($result2);

if (is_alphachar($_POST[username]) == 1) {
echo $form;
echo "Invalid Username. Only numbers/letters and underscores are
allowed.<br>";
die;
}

if ($num == 1) {

echo "<b>Error, username already exists!</b>";
echo "<br><br>";
echo "<a href=\"register.php\">Back to registration form.</a>";

} elseif ($num2 == 1) {
echo "Error, that email address has already been registered. Please select a
different one.";
} else {


$query = "INSERT INTO `USERS` (user_id, username, password, email) VALUES
('','".$_POST[username]."','".$_POST[password]."','".$_POST[email]."')";
$result = mysql_query($query,$link);
if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }
mysql_close();
echo "<tr><td align=\"center\">Congratulations <b><i>" . $_POST[username] .
"</i></b>! Your account has been created and added to database.";
echo "<br><a href=\"login.html\" onClick=\"close();\">Back to login
area</a></td></tr>";

}
}

?>
```

home.html

Περιέχει πληροφορίες για την εφαρμογή και παρουσιάζει το πρότυπο (template) των σελίδων.

```
<html>
<head>
```

```
<title>IMAGEGRID</title>
<style type="text/css">
<!--
.style1 {color: #CA0000}
-->
</style>

<STYLE TYPE="text/css">

.menuheader {
       BORDER-COLOR : #000000 ;
       cursor : hand ;
       Border-Left : #000000 ;
       Border-Top : #000000 ;
       Padding-Left : 1px ;
       Padding-Top : 1px ;
       Background-Color : #CA0000 ;
}

.menu {
       Background-Color : white ;
}

.home {
       cursor : pointer ;
}

.menulinks {
       text-decoration:none;
}

</STYLE>

</head>

<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">

<table border="1" cellpadding="0" cellspacing="0" style="border-collapse:
collapse; border-width: 0" bordercolor="#111111" width="100%"
id="AutoNumber1" height="244">
  <tr>
    <td width="100%" style="border-style: none; border-width: medium"
bgcolor="#E2E2E2" height="68" valign="top">
    <table border="1" cellpadding="0" cellspacing="0" style="border-
collapse: collapse; border-width: 0" bordercolor="#111111" width="100%"
id="AutoNumber5" height="72">

      <tr>
        <td width="33%" style="border-style: none; border-width: medium"
height="70" bgcolor="#CA0000">
        <p style="margin-left: 7">
        <b><font face="Verdana, Arial, Helvetica, sans-serif" size="6"
color="#FFFFFF">IMAGEGRID</font></b></td>
        <td width="16%" style="border-style: none; border-width: medium"
height="70" bgcolor="#CA0000"><img src="images/home.png" width="100%"
height="70"></td>
        <td width="51%" style="border-style: none; border-width: medium"
height="70" bgcolor="#CA0000">
        <p align="center" style="margin-top: 0; margin-bottom: 0"><b><i>
        <font face="Verdana" color="#FFFFFF">

<!-- Search Google -->
<right>
<FORM method=GET action="http://www.google.com/search">
<table align="right"><tr><td>
<a href="http://www.google.com/">
<IMG  width="70" height="20"
SRC="http://www.google.com/logos/Logo_40wht.gif"> <br></a>
```

```
<INPUT TYPE=text name=q size=31 maxlength=255 value="">
<INPUT TYPE=hidden name=hl value="en">
<INPUT type=submit name=btnG VALUE="Google Search">

</td></tr></TABLE>
</FORM>
</right>
<!-- Search Google -->


 <br></font></i></b></p>
        </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr>
    <td width="100%" style="border-style: none; border-width: medium"
height="14">

    <table border="1" cellpadding="0" cellspacing="0" style="border-
collapse: collapse; border-width: 0" bordercolor="#111111" width="100%"
id="AutoNumber2">
      <tr>
        <td width="20%" bgcolor="#C0C0C0" bordercolor="#E2E2E2"
style="border-left-style: none; border-left-width: medium; border-top-style:
none; border-top-width: medium; border-bottom-style: none; border-bottom-
width: medium" align="center">
        <b><font face="Arial" size="2" color="#FFFFFF"><a href="home.html"
color="#FFFFFF">HOME</a></font></b></td>
        <td width="20%" bgcolor="#C0C0C0" bordercolor="#E2E2E2"
style="border-top-style: none; border-top-width: medium; border-bottom-
style: none; border-bottom-width: medium" align="center">
        <b><font face="Arial" size="2" color="#FFFFFF"><a
href="https://195.251.6.234/ypapanti/addExamination.html">NEW
EXAMINATION</a></font></b></td>
        <td width="20%" bgcolor="#C0C0C0" bordercolor="#E2E2E2"
style="border-top-style: none; border-top-width: medium; border-bottom-
style: none; border-bottom-width: medium" align="center">
        <b><font face="Arial" size="2" color="#FFFFFF"><a
href="https://195.251.6.234/ypapanti/allSearch.html">SEARCH</a></font></b></
td>

        <td width="20%" bgcolor="#C0C0C0" bordercolor="#E2E2E2"
style="border-top-style: none; border-top-width: medium; border-bottom-
style: none; border-bottom-width: medium" align="center">
        <b><font face="Arial" size="2" color="#FFFFFF"><a
href="https://195.251.6.234/ypapanti/viewImages.php">GALLERY</a></font></b><
/td>
        <td width="20%" bgcolor="#C0C0C0" bordercolor="#E2E2E2"
style="border-right-style: none; border-right-width: medium; border-top-
style: none; border-top-width: medium; border-bottom-style: none; border-
bottom-width: medium" align="center">
        <b><font face="Arial" size="2" color="#FFFFFF"><a
href="https://195.251.6.234/ypapanti/logout.php">LOG OUT</a></font></b></td>
      </tr>
    </table>
    </td>
  </tr>

  <tr>
    <td width="100%" style="border-style: none; border-width: medium"
height="123" valign="top">
    <table border="1" cellpadding="0" cellspacing="0" style="border-
collapse: collapse; border-width: 0" bordercolor="#111111" width="100%"
id="AutoNumber6">
      <tr>
```

```
        <td width="20%" style="border-style: none; border-width: medium"
bgcolor="#CA0000">
          <div align="center">
          <left>
          <p> </p>

<br><br><br><br><br>
<!-- this script got from www.javascriptfreecode.com-Coded by: Krishna Eydat
-->
<!-- START OF Watermark Navigation Menu DHTML -->
<SCRIPT Language="Javascript1.2">

var menuwidth=220
var offsetleft=20
//var offsettop=150

var ns4=document.layers?1:0
var ie4=document.all?1:0
var ns6=document.getElementById&&!document.all?1:0

function makeStatic() {
if (ie4) {object1.style.pixelTop=document.body.scrollTop+offsettop}
else if (ns6)
{document.getElementById("object1").style.top=window.pageYOffset+offsettop}
else if (ns4)
{eval(document.object1.top=eval(window.pageYOffset+offsettop));}
setTimeout("makeStatic()",0);
}

if (ie4||ns6) {document.write('<span ALIGN="LEFT" ID="object1"
STYLE="Position:absolute; Top:150; Left:'+offsetleft+'; Z-
Index:5;cursor:hand;background-color:black;"><TABLE BORDER="1"
width="'+menuwidth+'" CELLPADDING="0" CELLSPACING="0" BORDERCOLOR="black"
bgcolor="white">')}
else if (ns4){ document.write('<LAYER top="150" name="object1"
left="'+offsetleft+'" BGCOLOR=black><TABLE BORDER="0" CELLPADDING="0"
CELLSPACING="1"><TR><TD><TABLE BORDER="0" CELLPADDING="0" CELLSPACING="0"
width="'+menuwidth+'">')}

if (ie4||ns6||ns4)
document.write('<TR><TD background="images/ob06.jpg"
BORDERCOLORDARK="#99CCFF" BORDERCOLORLIGHT="#003399"><P ALIGN=CENTER><FONT
SIZE="5" FACE=ARIAL>MENU</FONT></TD></TR>')

var menui = new Array();
var menul = new Array();

menui[0]="Home";
menui[1]="Add Patient";
menui[2]="Add Doctor";
menui[3]="Add Examination";
menui[4]="Gallery";
menui[5]="Search";
menui[6]="Log out";

menul[0]="https://195.251.6.234/ypapanti/home.html";
menul[1]="https://195.251.6.234/ypapanti/addPatient.html";
menul[2]="https://195.251.6.234/ypapanti/addDoctor.html";
menul[3]="https://195.251.6.234/ypapanti/addExamination.html";
menul[4]="https://195.251.6.234/ypapanti/viewImages.php";
menul[5]="https://195.251.6.234/ypapanti/allSearch.html";
menul[6]="https://195.251.6.234/ypapanti/logout.php";

for (i=0;i<=menui.length-1;i++)

if (ie4||ns6) {document.write('<TR><TD BORDERCOLOR="white"
ONCLICK="location=\''+menul[i]+'\'" onmouseover="className=\'menuheader\'"
```

```
onMouseout="className=\'menu\'"><CENTER><FONT>'+menui[i]+'</FONT></TD></TR>'
)}
else if (ns4){document.write('<TR><TD BGCOLOR="white"><ILAYER><LAYER
width="'+menuwidth+'" onmouseover="bgColor=\'yellow\'"
onmouseout="bgColor=\'white\'"><CENTER><A HREF="'+menul[i]+'"
class=menulinks>'+menui[i]+'</A></CENTER></LAYER></ILAYER></TD></TR>')}

if (ie4||ns6) {document.write('</TABLE></span>')}
else if (ns4){document.write('</TABLE></TD></TR></TABLE></LAYER>')}

function menu3(){
if (ns6||ie4||ns4)
makeStatic()
}

window.onload=menu3

</SCRIPT>

        </left>
        </div>
        <p> </p>
        <p> </p>
        </td>

        <td width="80%" rowspan="2" valign="top"
background="images/ob06.jpg" style="border-style: none; border-width:
medium">
       <h1 align="center" style="margin-left: 4; margin-top: 5; margin-
bottom: 2; margin-right:5"><font face="Verdana, Arial, Helvetica, sans-
serif"></font><font size="5" face="Verdana, Arial, Helvetica, sans-serif"
color="#000000">            <br>
              <font color="#CC0000">IMAGEGRID APPLICATION</font></font></h1>
        <p align="left" style="margin-left: 4; margin-top: 5; margin-bottom:
2; margin-right:5"> 
          </p>
        <table width="100%" border="0" cellpadding="20" cellspacing="0"
style="margin-bottom: 2">
          <tr>
        <td><p align="left" style="margin-left: 4; margin-top: 5; margin-
bottom: 2; margin-right:5"><font face="Verdana, Arial, Helvetica, sans-
serif"><font size="2"></font><font face="Verdana, Arial, Helvetica, sans-
serif"><font face="Verdana, Arial, Helvetica, sans-serif"><font
face="Verdana, Arial, Helvetica, sans-serif"></font></font></font>
       <font size="2">File sciences are in the center of scientific research
with an ongoing growing intensity especially in the field of medical and/or
biomedical data analysis.  The produced data worldwide need a great amount
of storage resources.</font><font face="Verdana, Arial, Helvetica, sans-
serif"><font face="Verdana, Arial, Helvetica, sans-serif"><font
face="Verdana, Arial, Helvetica, sans-serif"><font face="Verdana, Arial,
Helvetica, sans-serif"><font face="Verdana, Arial, Helvetica, sans-serif">
       <img src="images/pictureperfect.jpg" width="225" height="169"
hspace="5" vspace="15" align="left">
             </font></font></font></font></font><font size="2"></font><font
face="Verdana, Arial, Helvetica, sans-seri">
       <font face="Verdana, Arial, Helvetica, sans-serif"></font></font>
<font size="2">In this way, clever solutions are searched in order to
efficiently manage all these data. New technologies are in favor of this
approach. The grid offers an alternative approach at the process (parallel
programming) and data management because it is able to "scatter" all these
data in a great number of computing and storage elements in a safety way.
Medical data and, especially, medical images which play a key role in this
project, can use the grid technologies, as SEs are able to behavior like a
huge database. In this project, IMAGEGRID interface is presented and it
links the final user with the grid. Web services are a magic key that
undertake to do this connection absolutely implemented.</font><font
face="Verdana, Arial, Helvetica, sans-serif"></font>
             <font size="2"><p align="center"></p>
```

```
              </font></font></p>

              <p align="left" style="margin-left: 4; margin-top: 5; margin-
bottom: 2; margin-right:5"> <br><br></p>
              <p align="left" style="margin-left: 4; margin-top: 5; margin-
bottom: 2; margin-right:5"><font size="2" face="Verdana, Arial, Helvetica,
sans-serif">
                The Grid offers a number of Storage and Computing Elements
that can be used from the grid users in order to execute their experiments
and manage their input and output data. This application benefits from the
grid's SEs. Here, a user is able to upload DICOM files to the grid and
download them in a simple way. Totally, the implementation simulates a
limited information system that stores records for all involved patients and
doctors for each examination and each file as follows.</font><font
face="Verdana, Arial, Helvetica, sans-serif"><font size="2" face="Verdana,
Arial, Helvetica, sans-serif"><img src="images/Medical_Imaging.jpg"
width="300" height="207" align="right"></font></font><font size="2"
face="Verdana, Arial, Helvetica, sans-serif">
                A central point of this implementation is that final user
can use the grid with a friendly way by just creating an account at the
database. Therefore, users use the grid as using every other system, but
benefits from the advantage of having "infinite" storage space viewing his
files with a consolidated way. To implement such a system the combination of
various significant factors including gLite software provided by the Grid,
Internet technologies and database data is of major importance. The way that
engages all the above components is not only their technical characteristics
but also the philosophy that supports them. </font><font size="2"
face="Verdana, Arial, Helvetica, sans-serif"></font></p></td>
          </tr>
        </table>
        </td>
      </tr>
      <tr>
        <td style="border-style: none; border-width: medium"
bgcolor="#CA0000" valign="bottom"><p><a
href="http://www.theoscommercestore.com/custom-oscommerce-templates.htm"
target="_blank" class="style1">Custom osCommerce templates</a> </p>

          </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr>
    <td width="100%" style="border-style: none; border-width: medium"
height="17">
    <table border="1" cellpadding="0" cellspacing="0" style="border-
collapse: collapse; border-width: 0" bordercolor="#111111" width="100%"
id="AutoNumber3" height="1">
      <tr>

        <td width="100%" height="1" style="border-style: none; border-width:
medium" valign="top" bgcolor="#111111">
        <table border="1" cellpadding="0" cellspacing="0" style="border-
collapse: collapse; border-width: 0" bordercolor="#111111" width="100%"
id="AutoNumber8">
          <tr>
        <td width="20%" bgcolor="#000000" bordercolor="#E2E2E2"
style="border-left-style: none; border-left-width: medium; border-top-style:
none; border-top-width: medium; border-bottom-style: none; border-bottom-
width: medium" align="center">
        <b></b></td>
        <td width="20%" bgcolor="#000000" bordercolor="#E2E2E2"
style="border-top-style: none; border-top-width: medium; border-bottom-
style: none; border-bottom-width: medium" align="center">
        <b><font face="Arial" size="2" color="#FFFFFF">UCG</font></b></td>
```

```
        <td width="20%" bgcolor="#000000" bordercolor="#E2E2E2"
style="border-top-style: none; border-top-width: medium; border-bottom-
style: none; border-bottom-width: medium" align="center">

        <b><font face="Arial" size="2" color="#FFFFFF">University of Central
Greece</font></b></td>
        <td width="20%" bgcolor="#000000" bordercolor="#E2E2E2"
style="border-top-style: none; border-top-width: medium; border-bottom-
style: none; border-bottom-width: medium" align="center">
        <b><font face="Arial" size="2" color="#FFFFFF">UCG</font></b></td>
        <td width="20%" bgcolor="#000000" bordercolor="#E2E2E2"
style="border-right-style: none; border-right-width: medium; border-top-
style: none; border-top-width: medium; border-bottom-style: none; border-
bottom-width: medium" align="center">
        <b></b></td>
          </tr>
        </table>
        </td>

      </tr>
      <tr>
        <td width="100%" height="18" style="border-style: none; border-
width: medium" bgcolor="#CA0000">
        <p align="center"><b><font face="Arial" size="1" color="#FFFFFF">
        Copyright 2008 You.com. All Rights Reserved.</font></b></td>
      </tr>
    </table>
    </td>

  </tr>
</table>
</body>

</html>
```

addPatient.html

Η φόρμα υποβολής των στοιχείων του ασθενή και οι έλεγχοι των υποχρεωτικών πεδίων. Παρόμοια δομή παρουσιάζουν οι αντίστοιχες φόρμες για την καταχώρηση ασθενών και εξετάσεων.

```
<html>
<head>
<title>IMAGEGRID</title>


<script type="text/javascript" language="JavaScript">
function notEmpty(){
       var myTextField = document.getElementById('surname');
       if(myTextField.value == "")
              alert("Would you please enter doctor's surname?")

}
</script>

<script type="text/javascript" language="JavaScript">
 function validateInt()
   {
      var o = document.getElementById('age');
      switch (isInteger(o.value))
      {
         case true:
            //alert(o.value + " is an integer")
            break;
         case false:
            alert(o.value + " is not an integer")
```

```
            }
   }
   function isInteger (s)
   {
      var i;

      if (isEmpty(s))
      if (isInteger.arguments.length == 1) return 0;
      else return (isInteger.arguments[1] == true);

      for (i = 0; i < s.length; i++)
      {
         var c = s.charAt(i);

         if (!isDigit(c)) return false;
      }

      return true;
   }

   function isEmpty(s)
   {
      return ((s == null) || (s.length == 0))
   }

   function isDigit (c)
   {
      return ((c >= "0") && (c <= "9"))
   }
</script>

</head>

<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">

        <p> </p>
        <p> </p>
        </td>

        <td width="80%" rowspan="2" valign="top"
background="images/ob06.jpg" style="border-style: none; border-width:
medium">
       <h3 align="center" style="margin-left: 4; margin-top: 5; margin-
bottom: 2; margin-right:5"><font face="Verdana, Arial, Helvetica, sans-
serif"></font><font size="5" face="Verdana, Arial, Helvetica, sans-serif"
color="#000000">            <br>
              <font color="#CC0000">Add Doctor</font></font></h3>

       <br><br>
<table width="400" border="1" align="center">
<form action="addPatient.php" method="post">
  <tr>
    <td>Surname:</td><td><input type="text" name="surname" id="surname"
size="25" value="" onBlur="notEmpty();"/></td>

  </tr>
  <tr>
    <td>Name:</td><td><input type="text" name="name" id="name" size="25"
value=""/></td>
  </tr>
  <tr>
    <td>Age:</td><td><input type="text" name="age" id="age" size="10"
value=""/></td>
  </tr>
  <tr>

    <td>Sex (F or M)</td><td><input type="text" name="sex" id="sex" size="2"
value=""/></td>
```

```
  </tr>
  <tr>
    <td>Insurance Fund </td><td><input type="text" name="insurance"
id="insurance" size="10" value=""/></td>
  </tr>
  <tr>
    <td>Address</td><td><input type="text" name="address"
id="address"size="40" value=""/></td>

  </tr>
  <tr>
    <td>Phone</td><td><input type="text" name="phone" id="phone" size="10"
value=""/></td>
  </tr>
  <tr>
  <td> </td>
  <td><input type="reset" value="Reset" align="right" /><input type="submit"
value="Submit" onclick="validateInt(age)"/></td>
  </tr>
  </form>
</td></tr>
</table>
<br><br><br><br><br><br><br><br><br><br>

       </td>
          </tr>
        </table>
</body>

</html>
```

addDoctor.html

```
<html>
<head>
<title>IMAGEGRID</title>


<script type="text/javascript" language="JavaScript">
function notEmpty(){
       var myTextField = document.getElementById('surname');
       if(myTextField.value == "")
              alert("Would you please enter patient's surname?")

}
</script>

<script type="text/javascript" language="JavaScript">
 function validateInt()
   {
      var o = document.getElementById('age');
      switch (isInteger(o.value))
      {
         case true:
            //alert(o.value + " is an integer")
            break;
         case false:
            alert(o.value + " is not an integer")
              }
   }
   function isInteger (s)
   {
      var i;

      if (isEmpty(s))
      if (isInteger.arguments.length == 1) return 0;
      else return (isInteger.arguments[1] == true);
```

```
      for (i = 0; i < s.length; i++)
      {
         var c = s.charAt(i);

         if (!isDigit(c)) return false;
      }

      return true;
   }

   function isEmpty(s)
   {
      return ((s == null) || (s.length == 0))
   }

   function isDigit (c)
   {
      return ((c >= "0") && (c <= "9"))
   }
</script>

</head>

<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">

        <p> </p>
        <p> </p>
        </td>

        <td width="80%" rowspan="2" valign="top"
background="images/ob06.jpg" style="border-style: none; border-width:
medium">
       <h3 align="center" style="margin-left: 4; margin-top: 5; margin-
bottom: 2; margin-right:5"><font face="Verdana, Arial, Helvetica, sans-
serif"></font><font size="5" face="Verdana, Arial, Helvetica, sans-serif"
color="#000000">            <br>
              <font color="#CC0000">Add Doctor</font></font></h3>

       <br><br>
<table width="400" border="1" align="center">
<form action="addDoctor.php" method="post">
  <tr>
    <td>Surname:</td><td><input type="text" name="surname"
id="surname"size="25" value="" onBlur="notEmpty();"/></td>

  </tr>
  <tr>
    <td>Name:</td><td><input type="text" name="name" id="name" size="40"
value=""/></td>
  </tr>
  <tr>
    <td>Medical Specialty:</td><td><input type="text" name="specialty"
id="age" size="10" value=""/></td>
  </tr>
  <tr>
    <td>Address:</td><td><input type="text" name="address"
id="address"size="40" value=""/></td>
  </tr>
  <tr>
    <td>Phone:</td><td><input type="text" name="phone" id="phone" size="10"
value=""/></td>
  </tr>
  <tr>
  <td> </td>
  <td><input type="reset" value="Reset" align="right" /><input type="submit"
value="Submit" /></td>
  </tr>
```

```
  </form>
</td></tr>
</table>
<br><br><br><br><br><br><br><br><br><br><br>

       </td>
          </tr>
        </table>
</body>

</html>
```

addExamination.html

```
<html>
<head>
<title>IMAGEGRID</title>


<SCRIPT language=JavaScript>
function view1(){
       window.open('http://195.251.6.234/ypapanti/patient.php', "patient",
"status=1, toolbar=1, scrollbars=1,  width=800, height=300");
}
</script>

<SCRIPT language=JavaScript>
function view2(){
       window.open('http://195.251.6.234/ypapanti/doctor.php', "doctor",
"status=1, toolbar=1, scrollbars=1, width=800, height=300");
}
</script>


</head>

<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
       <h3 align="center" style="margin-left: 4; margin-top: 5; margin-
bottom: 2; margin-right:5"><font face="Verdana, Arial, Helvetica, sans-
serif"></font><font size="5" face="Verdana, Arial, Helvetica, sans-serif"
color="#000000">            <br>
              <font color="#CC0000">Add Examination</font></font></h3>

       <br><br>
<table width="400" border="1" align="center">

<form name="main" action="addExamination.php" method="post">
       <tr>
              <td>
              <input type="button" value="Find out your patient's id."
onClick="view1();">
              </td><td><input type="text" name="patient_id" id="patient_id"
size="10" value=""/></td>
       </tr>
       <tr>
              <td>
              <input type="button" value="Find out your doctor's id."
onClick="view2();">
              </td><td><input type="text" name="doctor_id" id="doctor_id"
size="10" value=""/></td>
       </tr>
  <tr>
    <td>Machine Handler:</td><td><input type="text" id="name" name="name"
size="40" value="" onblur="notEmpty(id)"/></td>
  </tr>
  <tr>
```

```
    <td>Examination Type:</td><td><input type="text" id="type" name="type"
size="10" value=""/></td>
  </tr>
  <tr>
    <td>Type of incident</td><td><input type="text" id="incident"
name="incident" size="10" value=""/></td>
  </tr>
  <tr>
    <td>Number of Files</td><td><input type="text" id="filesNum"
name="filesNum" size="10" value=""/></td>
  </tr>
  <tr>
    <td>More Information</td>
    <td><input type="textarea" name="comment" id="comment" size="40"
value=""/></td>
  </tr>
  <tr>
  <td><i><font size= "2" color= "red">The Date of the examination stored
automatically.</font></i></td><td><input type="reset" value="Reset"
align="right" />
             <input type="submit" value="Next>>"
onclick="validateInt(filesNum)"/></td>
</tr>
</form>

</td></tr>
<tr><td><a href="addPatient.html">Add Patient</a> <a href="addDoctor.html">
Add Doctor</a></td></tr>
</table>
<br><br><br><br><br><br><br><br>
</body>

</html>
```

addPatient.php

Ο παρακάτω κώδικας ελέγχει την ύπαρξη μιας ίδιας εγγραφής στην βάση δεδομένων και αν δεν υπάρχει καταχωρούνται τα νέα στοιχεία. Στη συγκεκριμένη περίπτωση, πραγματοποιείται έλεγχος του πεδίου Surname, το οποίο δεν επιτρέπεται να είναι κενό, μέσω της συνάρτησης notEmpty(), και του πεδίου Age μέσω της validateInt(age) για να διαπιστωθεί αν η τιμή που εισήχθη είναι πράγματι αριθμός. Οι συναρτήσεις αυτές είναι υλοποιημένες με Javascript στο HEAD τμήμα του εγγράφου. Με παρόμοιο τρόπο μπορεί να πραγματοποιηθεί αναζήτηση στη βάση για κάθε οντότητα υποβάλλοντας το κατάλληλο ερώτημα. Δηλαδή αν αναζητείται ένας ασθενής, η επιλογή των στοιχείων του από τη βάση δεδομένων γίνεται με βάση το όνομα και το επώνυμο του όπως ορίζονται από τα πεδία $ssearch και $nsearch. Σε πρώτη φάση εμφανίζονται στοιχεία για τον ασθενή, για παράδειγμα, όπως το ονοματεπώνυμό του, η ηλικία του, το φύλο του, το ασφαλιστικό του ταμείο, η διεύθυνση, το τηλέφωνο του και το πλήθος των εξετάσεων που έχει. Κατ' αντιστοιχία, γίνεται η εισαγωγή νέων ασθενών.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="REFRESH"
content="1;url=http://195.251.6.234/ypapanti/addPatient.html"></HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
```

```
<?php

include 'config.php';

$id=$_POST['id'];
$surname=$_POST['surname'];
$name=$_POST['name'];
$age=$_POST['age'];
$sex=$_POST['sex'];
$insurance_fund=$_POST['insurance'];
$address=$_POST['address'];
$phone=$_POST['phone'];

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

//Elegxos gia hdh kataxwrhmenh eggrafh
$f_query = "SELECT * FROM `PATIENT` WHERE Surname='".$surname."' AND
Name='".$name."' AND Age='".$age."' AND Sex='".$sex."' AND `Insurance
Fund`='".$insurance_fund."' AND Address='".$address."' AND
Phone='".$phone."'";
$f_result = mysql_query($f_query, $link);
while($thisrow=mysql_fetch_row($f_result))
{
$i=0;
while ($i < mysql_num_fields($f_result)){
$field_name=mysql_fetch_field($f_result, $i);
$i++;
//echo $thisrow[$i] . " ";
}
}
if($field_name==NULL){
$query = "INSERT INTO `PATIENT` (Patient_id, Surname, Name, Age, Sex,
`Insurance Fund`, Address, Phone) VALUES
('','".$surname."','".$name."',".$age.",'".$sex."','".$insurance_fund."','".
$address."','".$phone."')";
}
else{
echo "There is another record with the same elements at the Database.";

}


if (!mysql_query($query))
  {
 // include ("insert2.html");
  die('Error: ' . mysql_error());
  }
mysql_close();

//echo "1 patient record added";

?>


</html>
```

addDoctor.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="REFRESH"
content="1;url=http://195.251.6.234/ypapanti/addDoctor.html"></HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<?php

include 'config.php';

$id=$_POST['id'];
$surname=$_POST['surname'];
$name=$_POST['name'];
$MedicalSpecialty=$_POST['specialty'];
$address=$_POST['address'];
$phone=$_POST['phone'];

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

//Elegxos gia hdh kataxwrhmenh eggrafh
$d_query = "SELECT * FROM `DOCTOR` WHERE Surname='".$surname."' AND
Name='".$name."' AND MedicalSpecialty='".$MedicalSpecialty."' AND
Address='".$address."' AND Phone='".$phone."'";
$d_result = mysql_query($d_query, $link);

while($thisrow=mysql_fetch_row($d_result))
{
       $i=0;
       while ($i < mysql_num_fields($d_result)){
              $field_name=mysql_fetch_field($d_result, $i);
              $i++;
              //echo $thisrow[$i] . " ";
       }
}

if($field_name==NULL){

$query = "INSERT INTO `DOCTOR` (Doctor_id, Surname, Name, MedicalSpecialty,
Address, Phone) VALUES
('','".$surname."','".$name."','".$MedicalSpecialty."','".$address."','".$ph
one."')";
//echo "size of result = ".sizeof($query)."<br>";
}
else{
echo "There is another record with the same elements at the Database.";
}

if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }
mysql_close();

//echo "1 doctor record added";
```

```
?>

</html>
```

addExamination.php

Η δημιουργία μιας νέας εξέτασης σημαίνει πέρα από την αποθήκευση των δεδομένων που λαμβάνονται από τη φόρμα στη βάση δεδομένων, εκτέλεση των εντολών για την δημιουργία των φακέλων που θα φιλοξενήσουν τα δεδομένα τόσο σε τοπικό επίπεδο όσο και στο πλέγμα (RunMakeDIR.php).

```
<?php
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="REFRESH"
content="5;url=http://195.251.6.234/ypapanti/fileGrid_upload.html"></HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>

<?php

include 'config.php';

$date=date("Y-m-d");
$patient=$_POST['patient_id'];
$doctor=$_POST['doctor_id'];
$handler=$_POST['name'];
$exam=$_POST['type'];
$incident=$_POST['incident'];
$filesNum=$_POST['filesNum'];
$comment=$_POST['comment'];

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$query = "INSERT INTO `EXAMINATION` (Examination_id, Patient_id, Doctor_id,
Date, Machine_Handler, Examination_Type, Type_of_incident, Number_of_files,
More_information) VALUES ('', '".$patient."', '".$doctor."' , '".$date."',
'".$handler."','".$exam."','".$incident."','".$filesNum."','".$comment."')";

if (!mysql_query($query))
  {
  include ("insert3.html");
  die('Error: ' . mysql_error());
  }

mysql_close();

$num = mysql_insert_id($link);
$_SESSION['exam_id'] = $num;
```

```
mkdir("upload/" . $num);
require("RunMakeDIR.php");
?>
</body>
</html>
```

Patient.php

Εμφανίζει μια λίστα με ασθενείς σε αλφαβητική σειρά και δίνει τον αντίστοιχο κωδικό σε περίπτωση που ο ίδιος ο ασθενής δεν τον έχει απομνημονεύσει.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<!--
<font color ="blue"><MARQUEE BEHAVIOR="scroll" DIRECTION ="right"
background="ob06.jpg">University of Central Greece</MARQUEE></font>
-->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body id="customers" background="images/ob06.jpg">
<?php

include 'config.php';

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$p_query= "SELECT * FROM `PATIENT` ORDER BY `Surname`";
$p_result=mysql_query($p_query, $link);

$num = mysql_numrows($p_result);


if(!$p_result){
       die('Can\'t view elements: ' . mysql_error());
}

echo "<b><center>Database Output</center></b><br>";

$i=0;
echo "<table border=\"1\">";
echo "<tr bgcolor=\"yellow\" class=\"alt\">";
echo
"<td>Surname</td><td>Name</td><td><b>ID</b></td><td>Age</td><td>Sex</td><td>
Insurance Fund</td><td>Address</td><td>Phone</td>";
echo "</tr>";
while ($i < $num) {

$id=mysql_result($p_result,$i,"Patient_id");
$surname=mysql_result($p_result,$i,"Surname");
$name=mysql_result($p_result,$i,"Name");
$age=mysql_result($p_result,$i,"Age");
$sex=mysql_result($p_result,$i,"Sex");
$insurance=mysql_result($p_result,$i,"Insurance Fund");
$Address=mysql_result($p_result,$i,"Address");
```

```
$Phone=mysql_result($p_result,$i,"Phone");


echo "<tr>";
echo
"<td>$surname</td><td>$name</td><td><b>$id</b></td><td>$age</td><td>$sex</td
><td>$insurance</td><td>$Address</td><td>$Phone</td>";
echo "</tr>";

$i++;
}
echo "</table>";
if (!mysql_query($p_query))
  {
  die('Error: ' . mysql_error());
  }

mysql_close();
?>
</body>
</html>
```

Doctor.php

Είναι αντίστοιχη λειτουργία του patient.php.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<!--
<font color ="blue"><MARQUEE BEHAVIOR="scroll" DIRECTION ="right"
background="ob06.jpg">University of Central Greece</MARQUEE></font>
-->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
#customers
{
font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
width:100%;
border-collapse:collapse;
}
#customers td, #customers th
{
font-size:1em;
border:1px solid #98bf21;
padding:3px 7px 2px 7px;
}
#customers th
{
font-size:1.1em;
text-align:left;
padding-top:5px;
padding-bottom:4px;
background-color:#A7C942;
color:#ffffff;
}
#customers tr.alt td
{
color:#000000;
background-color:#EAF2D3;
}
</style>
</head>
```

```
<body id="customers" background="images/ob06.jpg">
<?php

include 'config.php';

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$d_query= "SELECT * FROM `DOCTOR` ORDER BY `Surname`";
$d_result=mysql_query($d_query, $link);

$num = mysql_numrows($d_result);


if(!$d_result){
       die('Can\'t view elements: ' . mysql_error());
}

echo "<b><center>Database Output</center></b><br><br>";

$i=0;
echo "<table border=\"1\">";
echo "<tr bgcolor=\"yellow\" class=\"alt\">";
echo "<td>Surname</td><td>Name</td><td><b>ID</b></td><td>Medical
Specialty</td><td>Address</td><td>Phone</td>";
echo "</tr>";

while ($i < $num) {

$id=mysql_result($d_result,$i,"Doctor_id");
$surname=mysql_result($d_result,$i,"Surname");
$name=mysql_result($d_result,$i,"Name");
$med=mysql_result($d_result,$i,"MedicalSpecialty");
$Address=mysql_result($d_result,$i,"Address");
$Phone=mysql_result($d_result,$i,"Phone");

echo "<tr>";
echo
"<td>$surname</td><td>$name</td><td><b>$id</b></td><td>$med</td><td>$Address
</td><td>$Phone</td>";
echo "</tr>";

$i++;
}

if (!mysql_query($d_query))
  {
  //include ("insert3.html");
  die('Error: ' . mysql_error());
  }

mysql_close();
?>
</body>
</html>
```

fileGrid upload.html

```
<?php
```

```
session_start();

$file_id = $_SESSION['file_id'];
?>

<html>
<head>
<title>IMAGEGRID</title>
</head>

<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
       <h3  align="center"  style="margin-left:  4;  margin-top:  5;  margin-
bottom:  2;  margin-right:5"><font  face="Verdana,  Arial,  Helvetica,  sans-
serif"></font><font  size="5"  face="Verdana,  Arial,  Helvetica,  sans-serif"
color="#000000">            <br>
              <font color="#CC0000">Upload your File</font></font></h3>

       <br><br><br><br>

<table width="600" border="1" align="center" >
<form action="fileGrid_upload.php" method="post" enctype="multipart/form-
data">
<tr><td>
<label for="file">Choose a .ZIP or .RAR file to upload:</label></td><td>
<input type="file" name="file" id="file" size="50" /></td>
<tr>
<td>
Add a short description:</td><td>
<textarea type="text" name="description" id="description" value="" rows="2"
cols="20"/></textarea>
</td>
</tr>
<tr><td><h5>The max upload file size is limited at 10M.</h5></td><td>
<input type="reset" name="reset" value="Clear" align="right">
<input type="submit" name="submit" value="Upload File" aling="right"/>
</td>
</tr>
</form>
</body>

</html>
```

fileGrid upload.php

Το αρχείο αποθηκεύεται στον εξυπηρετητή και εξάγονται κάποια από τα μεταδεδομένα του αρχείου πριν γίνει η αποστολή του σε κάποιο SE.

```
<?php
session_start();
//echo $_SESSION['exam_id'];
$exam_id = $_SESSION['exam_id'];
//$filename = $_SESSION['filename'];
?>


<html>
<head>
<title>IMAGEGRID</title>

<?php

include 'config.php';

$description=$_POST['description'];
$se = "se01.kallisto.hellasgrid.gr";
```

```
$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}


if (($_FILES["file"]["type"] == "application/zip") &&
($_FILES["file"]["type"] == "application/rar"))
      {
  if ($_FILES["file"]["error"] > 0)
    {
    echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
    }
  else
    {
    echo "<h3 align=\"left\" >Information about your file...</h3>";
    echo "<b>";
    echo "You just upload: " . $_FILES["file"]["name"] . "<br />";
    echo "Type: " . $_FILES["file"]["type"] . "<br />";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

    $tmp = $_FILES["file"]["tmp_name"];

    $filename = $_FILES["file"]["name"];

    if (file_exists("upload/$exam_id/" . $_FILES["file"]["name"]))
      {
      echo $_FILES["file"]["name"] . " already exists. ";
      }
    else
      {

       $path = "upload/" . $exam_id;
       move_uploaded_file($_FILES["file"]["tmp_name"], $path . "/" .
$_FILES["file"]["name"]);

       $query = "INSERT INTO `FILE` (File_id, Examination_id, Filename,
SE_Location, FileType, FileSize, Description) VALUES
('','".$exam_id."','".$filename."','".$se."', '" . $_FILES["file"]["type"]
."', '" . ($_FILES["file"]["size"] / 1024) . "', '".$description."')";
       echo $query;
      }
    }
  echo "<br>";
       if (!mysql_query($query))
       {
              die('Error: ' . mysql_error());
       }
}
else
  {
  echo "Invalid file";
  echo "<br>";
  }



mysql_close();

$num = mysql_insert_id($link);
$_SESSION['file_id'] = $num;
```

```
$_SESSION['filename'] = $filename;
$_SESSION['tmp'] = $tmp;

require("RunLcgCr.php");
echo "<br>";
echo "<a href=\"imageUpload.html\">Add a single DICOM (.dcm) file if you
wish. </a>";
echo "<br><br><br><br><br><br><br><br>";
?>
```

RunMakeDIR.php

Αποτελέι έναν php-client και ζητά την εκτέλεση της αντίστοιχης μεθόδου από τη ImageGridWS. Αφορά την δημιουργία ενός νέου καταλόγου στο πλέγμα.

```
<?php
session_start();

$exam_id = $_SESSION['exam_id'];

require_once('/var/www/html/eLICO/lib/nusoap.php');

$wsdl="http://195.251.6.234:8080/axis2/services/ImageGridWS?wsdl";
$client=new nusoap_client($wsdl, true);

$param = array('path'=>"/grid/see/imagegrid",
               'examId'=>"$exam_id");

$result = $client->call("MakeDIR", $param);

//print_r($result);

?>
```

RunLcgCr.php

Αποτελέι έναν php-client και ζητά την εκτέλεση της αντίστοιχης μεθόδου από τη ImageGridWS. Αφορά την αντιγραφή ενός αρχείου σε κάποιο SE που καθορίζεται μέσω τον παραμέτρων, όπως και το μονοπάτι για το σωστό φάκελο.

```
<?php
session_start();

$exam_id = $_SESSION['exam_id'];
$filename = $_SESSION['filename'];
//$se = "se01.kallisto.hellasgrid.gr";

require_once('/var/www/html/eLICO/lib/nusoap.php');

$wsdl="http://195.251.6.234:8080/axis2/services/ImageGridWS?wsdl";
$client=new nusoap_client($wsdl, true);

$param = array('path'=>"/grid/see/imagegrid",
              'examId'=>"$exam_id",
              'filename'=>"$filename",
              'se'=>"se01.kallisto.hellasgrid.gr");
//'se'=>"se"

$result = $client->call("RunLcgCr", $param);

echo "The GUID from your uploaded file: ";
echo "<br>";
print_r($result);

?>
```

RunLcgCp.php

Αποτελέι έναν php-client και ζητά την εκτέλεση της αντίστοιχης μεθόδου από τη ImageGridWS. Αφορά την αντιγραφή ενός αρχείου από κάποιο SE στον εξυπηρετητή και στη συνέχεια μέσω του dowload.php τοπικά στον υπολογιστή του χρήστη.

```
<?php
session_start();

$dexam_id = $_SESSION['dexam_id'];
$filename = $_SESSION['filename'];

require_once('/var/www/html/eLICO/lib/nusoap.php');

$wsdl="http://195.251.6.234:8080/axis2/services/ImageGridWS?wsdl";
$client=new nusoap_client($wsdl, true);


$param = array('path'=>"/grid/see/imagegrid",
              'dexamId'=>"$dexam_id",
              'filename'=>"$filename");

$result = $client->call("RunLcgCp", $param);

?>
```

download.php

```
<?php
session_start();
$filename = $_GET['download_file'];
$file_id = $_GET['file_id'];

include 'config.php';

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}


$query = "SELECT `FILE`.`Examination_id` FROM `FILE` WHERE `Filename` =
'".$filename."' AND `File_id` = '".$file_id."'";
$result = mysql_query($query, $link);
$num = mysql_numrows($result);

$i=0;
while ($i < $num) {

$dexam_id = mysql_result($result,$i,"Examination_id");
$i++;

}
$_SESSION['dexam_id'] = $dexam_id;
$_SESSION['filename'] = $filename;

if(!$result){
       die('Can\'t view elements: ' . mysql_error());
}
```

```
if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }

mysql_close();
require("RunLcgCp.php");


$path = $_SERVER['DOCUMENT_ROOT']."/ypapanti/download/";
$fullPath = $path.$_GET['download_file'];

if ($fd = fopen ($fullPath, "r")) {
    $fsize = filesize($fullPath);
    $path_parts = pathinfo($fullPath);
    $ext = strtolower($path_parts["extension"]);
    switch ($ext) {
        case "pdf":
        header("Content-type: image/jpeg"); // add here more headers for
diff. extensions
        header("Content-Disposition: attachment;
filename=\"".$path_parts["basename"]."\""); // use 'attachment' to force a
download
        break;
        default;
        header("Content-type: application/octet-stream");
        header("Content-Disposition:
filename=\"".$path_parts["basename"]."\"");
    }
    header("Content-length: $fsize");
    header("Cache-control: private"); //use this to open files directly
    while(!feof($fd)) {
        $buffer = fread($fd, 2048);
        echo $buffer;
    }
}
fclose ($fd);
exit;

unlink('/var/www/html/ypapanti/download/' . $filename);

?>
```

ViewImages.php

Παρουσιάζει μια σειρά από μικρογραφίες εικόνων που κάθε μια αντιστοιχεί στο αρχείο (ένα σετ εικόνων) που έχει αποθηκευτεί στο πλέγμα.

```
<?php
session_start();
?>

<html>
<head>
<title>IMAGEGRID</title>
</head><body>

<h3 align="center" style="margin-left: 4; margin-top: 5; margin-bottom: 2;
margin-right:5"><font face="Verdana, Arial, Helvetica, sans-
serif"></font><font size="5" face="Verdana, Arial, Helvetica, sans-serif"
color="#000000">            <br>
              <font color="#CC0000">View Images</font></font></h3>

<?php
//phpinfo();
```

```
include 'config.php';

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$query = "SELECT * FROM `FILE`";


$result = mysql_query($query, $link);


if(!$result){
       die('Can\'t view elements: ' . mysql_error());
}

$num = mysql_numrows($result);


$i=0;


while ($i < $num) {

$id = mysql_result($result,$i,"File_id");
$exam_id = mysql_result($result,$i,"Examination_id");
$filename = mysql_result($result,$i,"Filename");
$se_loc = mysql_result($result,$i,"SE_Location");
$filetype = mysql_result($result,$i,"FileType");
$filesize = mysql_result($result,$i,"FileSize");
$description = mysql_result($result,$i,"Description");
$newfilename = mysql_result($result,$i,"newFilename");
$name = mysql_result($result,$i,"name");


$file="upload/" . $exam_id . "/" . $filename;

echo "<div class=\"img\">";

echo "<img src=\"upload/$exam_id/$newfilename\" alt=\"\" width=\"100\"
height=\"100\" id=\"image\" >";
echo "<div class=\"desc\">";
echo "<br>Filename: $filename<br>";
echo "<a href=\"javascript:void(0);\" NAME=\"DICOM\" title=\" IMAGEGRID\"
onClick=window.open(\"dicom.php?id=$id\",\"DICOM\",\"width=600,height=800,0,
status=0\");>View Details... </a>";
echo "<a href=\"download.php?download_file=$filename&file_id=$id\">Download
here</a>";
echo "</div>";
echo "</div>";
$_SESSION['id'] = $id;

$i++;

}

if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }
```

```
mysql_close();

?>
</body></html>
```

dicom.php

Το αρχείο αυτό τραβά από τη βάση δεδομένων τις επιθυμητές πληροφορίες (από την επικεφαλίδα DICOM) και τις τυπώνει στην οθόνη.

```
<?php

session_start();
//$newfilename = $_SESSION['newfilename'];
//$exam_id = $_SESSION['$exam_id'];
include 'config.php';

$File_id = $_GET['id'];


$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$query =  "SELECT DISTINCT * FROM `FILE`, `DICOM` WHERE `DICOM`.`Did` =
'".$File_id."' AND `FILE`.`File_id` = '".$File_id."'";

$result = mysql_query($query, $link);

$num = mysql_numrows($result);


if(!$result){
       die('Can\'t view elements: ' . mysql_error());
}

echo "<html>";
echo "<body background=\"images/ob06.jpg\">";
echo "<b>Some information about the file.</b><br><br><br>";

$i=0;


echo "<table>";

while ($i < $num) {

$did = mysql_result($result,$i,"Did");
$PatientPosition= mysql_result($result,$i,"PatientPosition");
$BitsStored= mysql_result($result,$i,"BitsStored");
$ImageComments = mysql_result($result,$i,"ImageComments");
$Modality = mysql_result($result,$i,"Modality");
$PhotometricInterpretation =
mysql_result($result,$i,"PhotometricInterpretation");
$SliceThickness = mysql_result($result,$i,"SliceThickness");
$TransferSyntaxUID = mysql_result($result,$i,"TransferSyntaxUID");
$NumberOfFrames = mysql_result($result,$i,"NumberOfFrames");

$File_id = mysql_result($result,$i,"File_id");
```

```
$exam_id = mysql_result($result,$i,"Examination_id");

$filename = mysql_result($result,$i,"Filename");
$se_loc = mysql_result($result,$i,"SE_Location");
$filetype = mysql_result($result,$i,"FileType");
$filesize = mysql_result($result,$i,"FileSize");
$newfilename = mysql_result($result,$i,"newFilename");
$description = mysql_result($result,$i,"Description");
echo "<tr><td>";

echo "ID: $File_id";
echo "<br>";
echo "Filename: $filename";
echo "<br>";
echo "Filetype: $filetype";
echo "<br>";
echo "Filesize: $filesize (Kb)";
echo "<br>";
echo "SE Location: $se_loc";
echo "<br>";
echo "Modality: $Modality";
echo "<br>";
echo "Bits Stored: $BitsStored";
echo "<br>";
echo "Patient Position: $PatientPosition";
echo "<br>";
echo "Photometric Interpretation: $PhotometricInterpretation";
echo "<br>";
echo "Slice Thickness: $SliceThickness";
echo "<br>";
echo "Transfer Syntax UID: $TransferSyntaxUID";
echo "<br>";
echo "Number Of Frames: $NumberOfFrames ";
echo "<br>";
echo "Image Comments: $ImageComments";
echo "<br>";
echo "Description: $description";
echo "<br><br/>";
$i++;
}
echo "</td><td>";
echo "<img src=\"upload/$exam_id/$newfilename\" alt=\"\" width=\"250\"
height=\"250\" id=\"image\"></td>";
echo "</tr>";
echo "</table>";
echo "<i>If the image is presented in inappropriate way means that there are
not enough information for its convertion.</i><br/><br/>";
//echo "<a href=\"display.php\">Go back</a>";
echo "</body>";
echo "</html>";



if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }

mysql_close();
?>
```

exportdicom.php

Εξάγει στοιχεία της επικεφαλίδας DICOM και τα αποθηκεύει στη βάση δεδομένων.

```
<?php
session_start();
$exam_id = $_SESSION['exam_id'];
$file_id = $_SESSION['file_id'];
$name = $_SESSION['name'];
require_once('DICOM.php');

include 'config.php';

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$dicom = new File_DICOM();
$res = $dicom->parse("upload/$exam_id/$name");

// check for errors
if (PEAR::isError($res)) {
    die("Error: ".$res->getMessage()."\n");
}

// show a few attributes of the DICOM file using group and element index
$bodypartexamined=$dicom->getValue(0x0018, 0x0015);
$compressioforce=$dicom->getValue(0x0018, 0x00A2);
$imagecomments=$dicom->getValue(0x0020, 0x4000);
$modality=$dicom->getValue(0x0008, 0x0060);
$photometricinter=$dicom->getValue(0x0028, 0x0004);
$slicethickness=$dicom->getValue(0x0018, 0x0050);
$transfersyntax=$dicom->getValue(0x0002, 0x0010);
$admitingdate=$dicom->getValue(0x0038, 0x0020);
$bitsStored = $dicom->getValue(0x0028, 0x0101);


$depthofscanfield = $dicom->getValue(0x0018, 0x5050);
$DetectorInformationSequence = $dicom->getValue(0x0054, 0x0022);
$DeviceDescription = $dicom->getValue(0x0050, 0x0020);
$HistogramData = $dicom->getValue(0x0060, 0x3020);
$ImageDisplayFormat = $dicom->getValue(0x2010, 0x0010);
$ImagePosition = $dicom->getValue(0x2020, 0x0010);
$PatientPosition = $dicom->getValue(0x0018, 0x5100);
$PixelSpacing = $dicom->getValue(0x0028, 0x0030);
$PixelRepresentation = $dicom->getValue(0x0028, 0x0103);
$BodyPartThickness = $dicom->getValue(0x0018, 0x11A0);



$query = "INSERT INTO `DICOM` (Did, PatientPosition, BitsStored,
ImageComments, Modality, PhotometricInterpretation, SliceThickness,
TransferSyntaxUID, NumberOfFrames, File_id) VALUES
('','".$PatientPosition."','".$bitsStored."','".$imagecomments."','".$modali
ty."','".$photometricinter."','".$slicethickness."','".$transfersyntax."','"
.$numberofframes."','".$file_id."')";
//echo $query;



if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }
mysql_close();
```

```
?>
```

imageConvert.php

Εδώ πραγματοποιείται η μετατροπή των εικόνων από DICOM σε JPEG για λόγους συμπίεσης και παρουσίασής τους.

```
<?php
session_start();

$server = 'localhost';
$username = 'ypapanti';
$password = 'ypapanti2009';
$database = "imagegrid";

require_once 'Image/Transform.php';

?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="REFRESH"
content="0;url=http://195.251.6.234/ypapanti/fileGrid_upload.html">
</head>
<body background="var/www/html/ypapanti/ob06.jpg">
</body>
</html>

<?php
$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}


$name = $_SESSION['name'];
$filename = $_SESSION['filename'];
$exam_id = $_SESSION['exam_id'];
$tmp = $_SESSION['tmp'];


// factory pattern - returns an object
$at = Image_Transform::factory('IM');
if (PEAR::isError($at)) {
    die($at->getMessage());
}

// load the image file
$at->load('/var/www/html/ypapanti/upload/' . $exam_id . '/' . $name);

if (PEAR::isError($at)) {
    die($at->getMessage());
}


$newfilename = str_replace(".dcm", ".jpg", $name);
```

```
// scale image by percentage - 20% of its original size
$at->scalebyPercentage(20);
if (PEAR::isError($at)) {
    die($at->getMessage());
}

$at->save('/var/www/html/ypapanti/upload/' . $exam_id . '/' . $newfilename,
'jpg', null);

if (PEAR::isError($at)) {
    die($at->getMessage());
}

//displays the image
//$at->display();


$query = "UPDATE `FILE` SET `newFilename` = '".$newfilename."' WHERE
`Filename` = '".$filename."'";
$result = mysql_query($query, $link);


if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }

mysql_close();
?>
```

allSearch.html

```
<form name="search" action="allSearch.php" method="get">
<table>

<tr>
<td align="left">
<select tabindex="10" name="table" onChange="selectTable()"
style="border:'border-width' 'border-style' 'color'; left:auto" />
<option value="" style="text-align:center">---Select---</option>
<option value="PATIENT">Patient</option>
<option value="DOCTOR">Doctor</option>
<option value="EXAMINATION">Examination</option>
<option value="FILE">File</option></select>
                      <br><br></td></tr>
<tr>
                           <td align="left">
<select tabindex="15" name="field" onChange="selectTable()"
style="border:'border-width' 'border-style' 'color'; left:auto" />
<option value="" style="text-align:center">---Select---</option>
<option value="" style="color:blue;text-align:center">-Patient-</option>
<option value="Surname">Surname</option>
<option value="Name">Name</option>
<option value="Sex">Sex</option>
<option value="Age">Age</option>
<option value="Insurance Fund">Insurance Fund</option>


<option value="" style="color:blue;text-align:center">-Doctor-</option>
<option value="Surname">Surname</option>
<option value="Name">Name</option>
<option value="MedicalSpecialty">Medical Specialty</option>
<option value="" style="color:blue;text-align:center">-Examination-</option>
<option value="Date">Date</option>
<option value="Machine_Handler">Machine Handler</option>
<option value="Examination_Type">Type of examination</option>
```

```
<option value="Type_of_incident">Type of incident</option>
<option value="Number_of_files">Number of files</option>
<option value="" style="color:blue;text-align:center">-File-</option>
<option value="Filename">Filename</option>
<option value="Filesize">Filesize</option>
</select>

                      <br><br></td>

                    </tr>
                    <tr>
                              <td>
<input type="text" name="key" id="key" size="30" maxlength="20" value=""
/></td>
                              </tr>  <tr>
<td>   <input type="submit" name="submit" value="Search" class="button" />
<input type="reset" name="clear" value="Clear" /></td>
                    </tr>

             </table></form>
```

allSearch.php

Αφού επιλεγούν τα κατάλληλα πεδία και τα κλειδιά της αναζήτησης καλείται αυτό το αρχείο για να πραγματοποιήσει την αναζήτηση.

```
<?php

include 'config.php';

$table=$_GET['table'];
$field=$_GET['field'];
$key=$_GET['key'];

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

if($table=='PATIENT'){
$query = "SELECT * FROM  `$table` WHERE `$field` LIKE '%".$key."%'";

$result = mysql_query($query, $link);
$num=mysql_numrows($result);

echo "<center>";
echo "<table border=\"1\">";
echo "<tr bgcolor=\"yellow\" class=\"alt\">";
echo
"<td>Surname</td><td>Name</td><td><b>ID</b></td><td>Age</td><td>Sex</td><td>
Insurance Fund</td><td>Address</td><td>Phone</td><td></td><td>Update</td>";
echo "</tr>";

$i=0;

if ($num !== 0){
while ($i < $num) {

$id = mysql_result($result,$i,"Patient_id");
$surname = mysql_result($result,$i,"Surname");
```

```
$name = mysql_result($result,$i,"Name");
$age = mysql_result($result,$i,"Age");
$sex = mysql_result($result,$i,"Sex");
$insurance = mysql_result($result,$i,"Insurance Fund");
$Address = mysql_result($result,$i,"Address");
$Phone = mysql_result($result,$i,"Phone");
$comm = mysql_result($result,$i,"comment");

echo "<tr>";
echo
"<td>$surname</td><td>$name</td><td><b>$id</b></td><td>$age</td><td>$sex</td
><td>$insurance</td><td>$Address</td><td>$Phone</td><td><a
href=\"javascript:void(0);\" NAME=\"INFO\" title=\" IMAGEGRID\"
onClick=window.open(\"morePatientInfo.php?id=$id\",\"INFO\",\"width=800,heig
ht=550,0,status=0,scrollbars=1\");>More information...</a></td><td><a
href=\"javascript:void(0);\" NAME=\"UPDATE\" title=\" IMAGEGRID\"
onClick=window.open(\"patient_update.php?id=$id\",\"UPDATE\",\"width=800,hei
ght=550,0,status=0\");>Update!</a></td>";
echo "</tr>";

$i++;
}
echo "</table>";
echo "</center>";
echo "<br><br><br><br><br><br><br><br><br><br><br><br><br>";
}
else{
echo "$name $surname has not carried tests yet.";
}
}
elseif($table=='DOCTOR'){

$query = "SELECT * FROM `$table` WHERE `$field` LIKE '%".$key."%'";
$result = mysql_query($query, $link);
//echo $query;

$num=mysql_numrows($result);
//echo $num;

$i=0;

echo "<center>";
echo "<table border=\"1\">";
echo "<tr bgcolor=\"yellow\" class=\"alt\">";
echo "<td>Surname</td><td>Name</td><td><b>ID</b></td><td>Medical
Specialty</td><td>Address</td><td>Phone</td><td></td><td>Update</td>";
echo "</tr>";

while ($i < $num) {

$doct_id = mysql_result($result,$i,"Doctor_id");
$dname = mysql_result($result,$i,"Name");
$dsurname = mysql_result($result,$i,"Surname");
$medsp = mysql_result($result,$i,"MedicalSpecialty");
$daddress = mysql_result($result,$i,"Address");
$dphone = mysql_result($result,$i,"Phone");

echo "<tr>";
echo
"<td>$dsurname</td><td>$dname</td><td><b>$doct_id</b></td><td>$medsp</td><td
>$daddress</td><td>$dphone</td><td><a href=\"javascript:void(0);\"
NAME=\"INFO\" title=\" IMAGEGRID\"
onClick=window.open(\"moreDoctorInfo.php?id=$doct_id\",\"INFO\",\"width=800,
height=550,0,status=0,scrollbars=1\");>More information...</a></td><td><a
href=\"javascript:void(0);\" NAME=\"UPDATE\" title=\" IMAGEGRID\"
onClick=window.open(\"doctor_update.php?id=$doct_id\",\"UPDATE\",\"width=800
,height=550,0,status=0\");>Update!</a></td>";
echo "</tr>";
```

```
$i++;
}
echo "</table>";
echo "</center>";
echo "<br><br><br><br><br><br><br><br><br><br><br><br><br>";
}

elseif($table=='EXAMINATION'){
$query= "SELECT * FROM `$table` WHERE `$field` LIKE '%".$key."%'";
$result = mysql_query($query, $link);

$num=mysql_numrows($result);

$i=0;
echo "<center>";
echo "<table border=\"1\">";
echo "<tr bgcolor=\"yellow\" class=\"alt\">";
echo "<td>Date</td><td>Machine Handler</td><td><b>ID</b></td><td>Examination
Type</td><td>Type of incident</td><td>Number of files</td><td>Any
comment</td>";
echo "</tr>";

while ($i < $num) {
$ex_id = mysql_result($result,$i,"Examination_id");
$pid = mysql_result($result,$i,"Patient_id");
$did = mysql_result($result,$i,"Doctor_id");
$date = mysql_result($result,$i,"Date");
$handler = mysql_result($result,$i,"Machine_Handler");
$type = mysql_result($result,$i,"Examination_Type");
$incident = mysql_result($result,$i,"Type_of_incident");
$numoffiles = mysql_result($result,$i,"Number_of_files");
$com = mysql_result($result,$i,"More_information");

echo "<tr>";
echo
"<td>$date</td><td>$handler</td><td><b>$ex_id</b></td><td>$type</td><td>$inc
ident</td><td>$numoffiles</td><td>$com</td>";
echo "</tr>";

$i++;
}
echo "</table>";
echo "</center>";
echo "<br><br><br><br><br><br><br><br><br><br><br><br><br>";
}
elseif($table=='FILE'){
$query= "SELECT * FROM `$table` WHERE `$field` LIKE '%".$key."%'";
$result = mysql_query($query, $link);

$num=mysql_numrows($result);

$i=0;
echo "<center>";
echo "<table border=\"1\">";
echo "<tr bgcolor=\"yellow\" class=\"alt\">";
echo "<td>Filename</td><td>Filesize (Kb)</td><td><b>ID</b></td><td>SE
Location</td><td>FileType</td><td>Description</td>";
echo "</tr>";

while ($i < $num) {
$fid = mysql_result($result,$i,"File_id");
$ex_id = mysql_result($result,$i,"Examination_id");
$fname = mysql_result($result,$i,"Filename");
$seLoc = mysql_result($result,$i,"SE_Location");
$type = mysql_result($result,$i,"FileType");
$size = mysql_result($result,$i,"FileSize");
$com = mysql_result($result,$i,"Description");
```

```
echo "<tr>";
echo
"<td>$fname</td><td>$size</td><td><b>$fid</b></td><td>$seLoc</td><td>$type</
td><td>$com</td>";
echo "</tr>";
$i++;
}
echo "</table>";
echo "</center>";
echo "<br><br><br><br><br><br><br><br><br><br><br><br><br>";
}


else{

echo "<b>Choose a table from the given in order to continue your
search.</b><br>";
echo "<b>Errors:</b><br>";
}
if($num == NULL){
       echo "There are no results!";
}
echo "<br>";
if(!$result){
       die('Can\'t view elements ' . mysql_error());
}

if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }

mysql_close();


?>
```

morePatientInfo.php

Σε περίπτωση που ο χρήστης θέλει να μάθει περισσότερες πληροφορίες για κάποιον ασθενή πέρα από τα δημογραφικά του στοιχεία καλεί μέσω ενός υπερσυνδέσμο αυτό το αρχείο και βλέπει το πλήθος των αρχείων και το είδος των εξετάσεων που έχει πραγματοποιήσει ο συγκεκριμ΄ρνος ασθενής.

```
 <?php
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<STYLE TYPE="text/css">

.menuheader {
       BORDER-COLOR : #000000 ;
       cursor : hand ;
       Border-Left : #000000 ;
       Border-Top : #000000 ;
       Padding-Left : 1px ;
       Padding-Top : 1px ;
       Background-Color : #FF0066 ;
}
```

```
.menu {
	Background-Color : white ;
}

.home {
	cursor : hand ;
}

.menulinks {
	text-decoration:none;
}

</STYLE>

</head>
<body background="images/ob06.jpg">

<h3 align="center">Search Results...</h3>
<?php

include 'config.php';

$id = $_GET['id'];

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$query = "SELECT DISTINCT `DOCTOR`.`Surname` , `DOCTOR`.`Name` ,
`DOCTOR`.`MedicalSpecialty` , `DOCTOR`.`Address` , `DOCTOR`.`Phone` ,
`EXAMINATION`.`Examination_id`, `EXAMINATION`.`Date` ,
`EXAMINATION`.`Machine_Handler` , `EXAMINATION`.`Examination_Type` ,
`EXAMINATION`.`Type_of_incident`
FROM `EXAMINATION` , `PATIENT` , `DOCTOR`
WHERE `EXAMINATION`.`Patient_id` = `PATIENT`.`Patient_id`
AND `EXAMINATION`.`Doctor_id` = `DOCTOR`.`Doctor_id`
AND `PATIENT`.`Patient_id`='".$id."'";

$result = mysql_query($query, $link);

$num=mysql_numrows($result);

echo "<br>";
$i=0;

echo "<center>";
echo "<table border=\"1\">";
echo "<tr bgcolor=\"yellow\" class=\"alt\">";
echo "<td>Date of examination</td><td>Type of examination</td><td>Type of
incident</td><td>Machine Handler</td><td>Doctor's name</td><td>Medical
Specialty</td><td>Address</td><td>Phone</td><Download the examination from
here.>";
echo "</tr>";

while ($i < $num) {

$dname = mysql_result($result,$i,"Name");
$dsurname = mysql_result($result,$i,"Surname");
$medsp = mysql_result($result,$i,"MedicalSpecialty");
$daddress = mysql_result($result,$i,"Address");
$dphone = mysql_result($result,$i,"Phone");
```

```
$ex_id = mysql_result($result,$i,"Examination_id");
$date = mysql_result($result,$i,"Date");
$handler = mysql_result($result,$i,"Machine_Handler");
$type = mysql_result($result,$i,"Examination_Type");
$incident = mysql_result($result,$i,"Type_of_incident");

echo "<tr>";
echo
"<td>$date</td><td>$type</td><td>$incident</td><td>$handler</td><td>$dname
$dsurname</td><td>$medsp</td><td>$daddress</td><td>$dphone</td><td><a
href=\"file.php?ex_id=$ex_id\">FILE</a></td>";
echo "</tr>";
$i++;
}
echo "</table>";
echo "</center>";

if($num == NULL){
       echo "There is not file!";
}


if(!$result){
       die('Can\'t view elements: ' . mysql_error());
}

if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }

mysql_close();
?>
</td></tr>
</table>
</body>
</html>
```

moreDoctorInfo.php

Με παρόμοιο τρόπο λειτουργεί και η διαδικασία για την αναζήτηση ενός ιατρού παρουσιάζοντας στοιχεία όπως πόσες εξετάσεις έχει παραγγείλει.

```
<?php
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<STYLE TYPE="text/css">

.menuheader {
       BORDER-COLOR : #000000 ;
       cursor : hand ;
       Border-Left : #000000 ;
       Border-Top : #000000 ;
       Padding-Left : 1px ;
       Padding-Top : 1px ;
       Background-Color : #FF0066 ;
}

.menu {
       Background-Color : white ;
}
```

```
.home {
       cursor : hand ;
}

.menulinks {
       text-decoration:none;
}

</STYLE>

</head>
<body background="images/ob06.jpg">

<h3 align="center">Search Results...</h3>
<?php

include 'config.php';

$id = $_GET['id'];

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$query = "SELECT `EXAMINATION`.`Date` , `EXAMINATION`.`Machine_Handler` ,
`EXAMINATION`.`Examination_Type` , `EXAMINATION`.`Type_of_incident`,
`PATIENT`.`Surname`, `PATIENT`.`Name`
FROM `EXAMINATION` , `PATIENT` , `DOCTOR`
WHERE `EXAMINATION`.`Patient_id` = `PATIENT`.`Patient_id`
AND `EXAMINATION`.`Doctor_id` = `DOCTOR`.`Doctor_id`
AND `DOCTOR`.`Doctor_id`='".$id."'";

$result = mysql_query($query, $link);

$num=mysql_numrows($result);

echo "<br>";
$i=0;

echo "<center>";
echo "<table border=\"1\">";
echo "<tr bgcolor=\"yellow\" class=\"alt\">";
echo "<td>Date of examination</td><td>Type of examination</td><td>Type of
incident</td><td>Machine Handler</td><td>Patient's name</td>";
echo "</tr>";

while ($i < $num) {

$pname = mysql_result($result,$i,"Name");
$psurname = mysql_result($result,$i,"Surname");

$date = mysql_result($result, $i, "Date");
$handler = mysql_result($result,$i,"Machine_Handler");
$type = mysql_result($result,$i,"Examination_Type");
$incident = mysql_result($result,$i,"Type_of_incident");

echo "<tr>";
echo
"<td>$date</td><td>$type</td><td>$incident</td><td>$handler</td><td>$pname
$psurname</td>";
```

```
echo "</tr>";


$i++;
}
echo "</table>";
echo "</center>";


if($num == NULL){
       echo "There are no results!";
}


if(!$result){
       die('Can\'t view elements: ' . mysql_error());
}

if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }

mysql_close();
?>
</td></tr>
</table>
</body>
</html>
```

file.php

Βρίσκει ένα συγκεκριμένο αρχείο και δίνει τη δυνατότητα λήψης του.

```
<?php

include 'config.php';
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<style type="text/css">
a:link {
  border-style: solid;
  border-width : 1px 4px 4px 1px;
  text-decoration : none;
  padding : 16px;
  border-color : #69f #00f #00f #69f;
}
a:hover { border-color: #ccc; }
</style>

</head>
<body background="images/ob06.jpg">

<h3 align="center">Download your file.</h3>
<center>

<?php
$ex_id=$_GET['ex_id'];

$link = mysql_connect($server, $username, $password);
if (!$link) {
```

```
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$query = "SELECT `FILE`.`File_id`, `FILE`.`Filename`, `FILE`.`SE_Location`
FROM `FILE`, `EXAMINATION` WHERE `EXAMINATION`.`Examination_id` =
'".$ex_id."' AND `EXAMINATION`.`Examination_id` = `FILE`.`Examination_id`";
$result = mysql_query($query, $link);

$num=mysql_numrows($result);
$j=0;

while ($j < $num) {
$file_id = mysql_result($result,$j,"File_id");
$filename = mysql_result($result,$j,"Filename");
$se = mysql_result($result,$j,"SE_Location");

$j++;
echo $filename . " is stored in " . $se . ".<br><br><br>";
echo "<a
href=\"download.php?download_file=$filename&file_id=$file_id\">DOWNLOAD</a>"
;
}
if($num == NULL){
       echo "<b>No file found!</b>";
}


if(!$result){
       die('Can\'t view elements: ' . mysql_error());
}

if (!mysql_query($query))
  {
  die('Error: ' . mysql_error());
  }

mysql_close();
?>

</center>
</body>
</html>
```

Patient update.php

Όταν ο χρήστης επιλέξει να κάνει ενημέρωση κάποιου στοιχείου το επιλέγει και κατόπιν καλεί το Pupdate.php.

```
<?php
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<STYLE TYPE="text/css">

.menuheader {
```

```
	BORDER-COLOR : #000000 ;
	cursor : hand ;
	Border-Left : #000000 ;
	Border-Top : #000000 ;
	Padding-Left : 1px ;
	Padding-Top : 1px ;
	Background-Color : #FF0066 ;
}

.menu {
	Background-Color : white ;
}

.home {
	cursor : hand ;
}

.menulinks {
	text-decoration:none;
}

</STYLE>

</head>
<body background="images/ob06.jpg">

<h3 align="center">Update...</h3>
<?php

include 'config.php';
$id =$_GET['id'];
$_SESSION['id'] = $id;

echo "<table border=\"0\" width=\"300\" align=\"center\">
<tr><td width=\"300\">
<form action=\"pupdate.php\" method=\"POST\">
</td></tr><tr><td>
<select tabindex=\"15\" name=\"field\" onChange=\"selectTable()\"
style=\"border:'border-width' 'border-style' 'color'; left:auto\" />
                                                <option value=\"\"
style=\"text-align:center\">---Select---</option>
                                                <option
value=\"Surname\">Surname</option>
                                                <option
value=\"Name\">Name</option>
                                                <option
value=\"Sex\">Sex</option>
                                                <option
value=\"Age\">Age</option>
                                                <option value=\"Insurance
Fund\">Insurance Fund</option>
                                                <option
value=\"Address\">Address</option>
                                                <option
value=\"Phone\">Phone</option>
</select><br>
New Value: <br><input type=\"text\" name=\"value\" id=\"value\"><br><br>
<input type=\"submit\" value=\"Update!\">
</form>
<br><br><br><br><br>
</td></tr>
</td></tr>

";

?>
</body>
</html>
```

Pupdate.php

```
<?php
session_start();
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>

<style type="text/css">
a:link {
  border-style: solid;
  border-width : 1px 4px 4px 1px;
  text-decoration : none;
  padding : 16px;
  border-color : #69f #00f #00f #69f;
}
a:hover { border-color: #ccc; }
</style>

</head>
<body background="images/ob06.jpg">
<?php
include 'config.php';

$id = $_SESSION['id'];
$field = $_POST['field']; //taken from the form
$value = $_POST['value']; //the new value taken from the form

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$sql = "UPDATE `PATIENT` SET $field = '".$value."' WHERE `Patient_id` =
'".$id."'";
$result = mysql_query($sql, $link);

mysql_close();
echo "<center>";
echo "<a href=\"\">CLOSE WINDOW</a>";
echo "<center>";
?>
</body>
</head>
```

Doctor_update.php

Έχει τον ίδιο ρόλο με το patient_update.php και ίδια λειτουργία.

```
<?php
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
<STYLE TYPE="text/css">

.menuheader {
	BORDER-COLOR : #000000 ;
	cursor : hand ;
	Border-Left : #000000 ;
	Border-Top : #000000 ;
	Padding-Left : 1px ;
	Padding-Top : 1px ;
	Background-Color : #FF0066 ;
}

.menu {
	Background-Color : white ;
}

.home {
	cursor : hand ;
}

.menulinks {
	text-decoration:none;
}

</STYLE>

</head>
<body background="images/ob06.jpg">

<h3 align="center">Update...</h3>
<?php

include 'config.php';

$doct_id = $_GET['id'];
$field = $_POST['field'];
$value = $_POST['value'];
$_SESSION['id'] = $doct_id;

echo "<table border=\"0\" width=\"300\" align=\"center\">
<tr><td width=\"300\">
<form action=\"dupdate.php\" method=\"POST\">
</td></tr><tr><td>
<select tabindex=\"15\" name=\"field\" onChange=\"selectTable()\"
style=\"border:'border-width' 'border-style' 'color'; left:auto\" />
	<option value=\"\" style=\"color:blue;text-align:center\">-Select-
</option>
	<option value=\"Surname\">Surname</option>
	<option value=\"Name\">Name</option>
	<option value=\"MedicalSpecialty\">Medical Specialty</option>
	<option value=\"Address\">Address</option>
	<option value=\"Phone\">Phone</option>
</select>
<br>
New Value: <br><input type=\"text\" name=\"value\" id=\"value\"><br><br>
<input type=\"submit\" value=\"Update!\">
</form>
<br><br><br><br><br>
</td></tr>
</td></tr>

";

?>
</body>
</html>
```

Dupdate.php

```
<?php
session_start();
?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>

<style type="text/css">
a:link {
  border-style: solid;
  border-width : 1px 4px 4px 1px;
  text-decoration : none;
  padding : 16px;
  border-color : #69f #00f #00f #69f;
}
a:hover { border-color: #ccc; }
</style>

</head>
<body background="images/ob06.jpg">
<?php
include 'config.php';

$doct_id = $_SESSION['id'];
$field = $_POST['field'];
$value = $_POST['value'];

$link = mysql_connect($server, $username, $password);
if (!$link) {
    die('Not connected : ' . mysql_error());
}

// make imagegrid the current db
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Can\'t use imagegrid : ' . mysql_error());
}

$sql = "UPDATE `DOCTOR` SET $field = '".$value."' WHERE `Doctor_id` =
'".$doct_id."'";
$result = mysql_query($sql, $link);


mysql_close();
echo "<center>";
echo "<a href=\"\">CLOSE WINDOW</a>";
echo "<center>";
?>
</body>
</head>
```

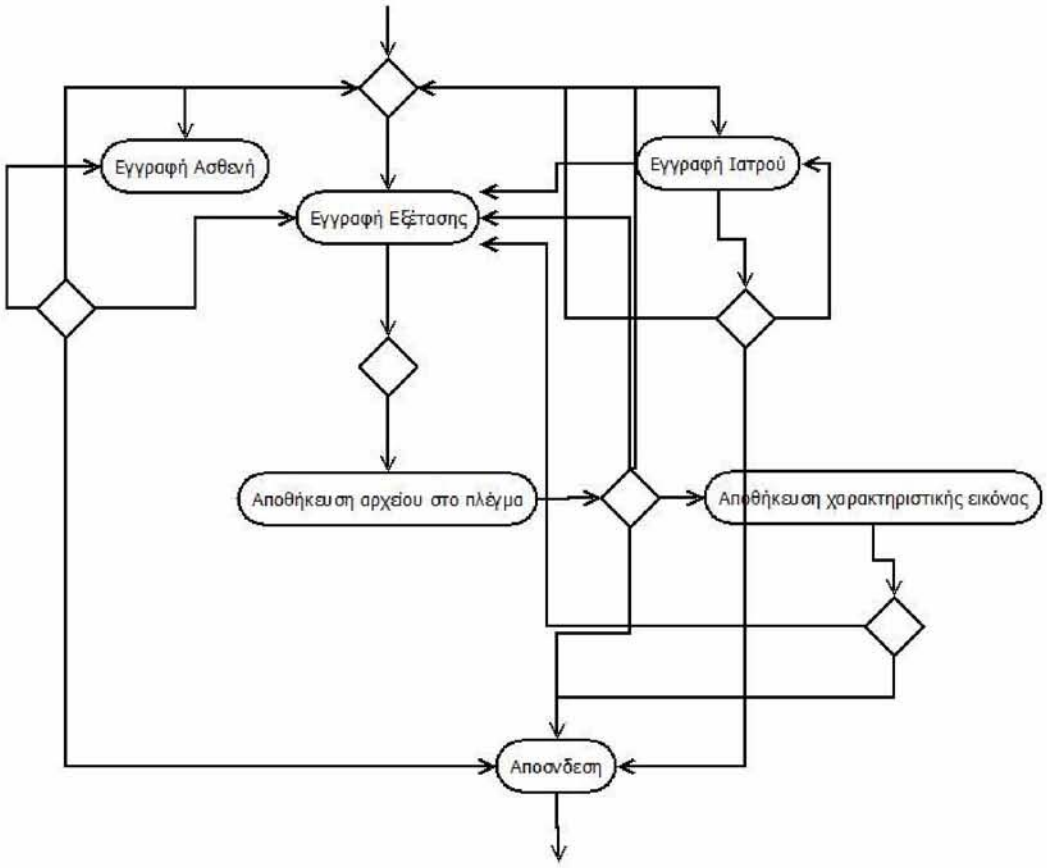
Logout.php

```
<?
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IMAGEGRID</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

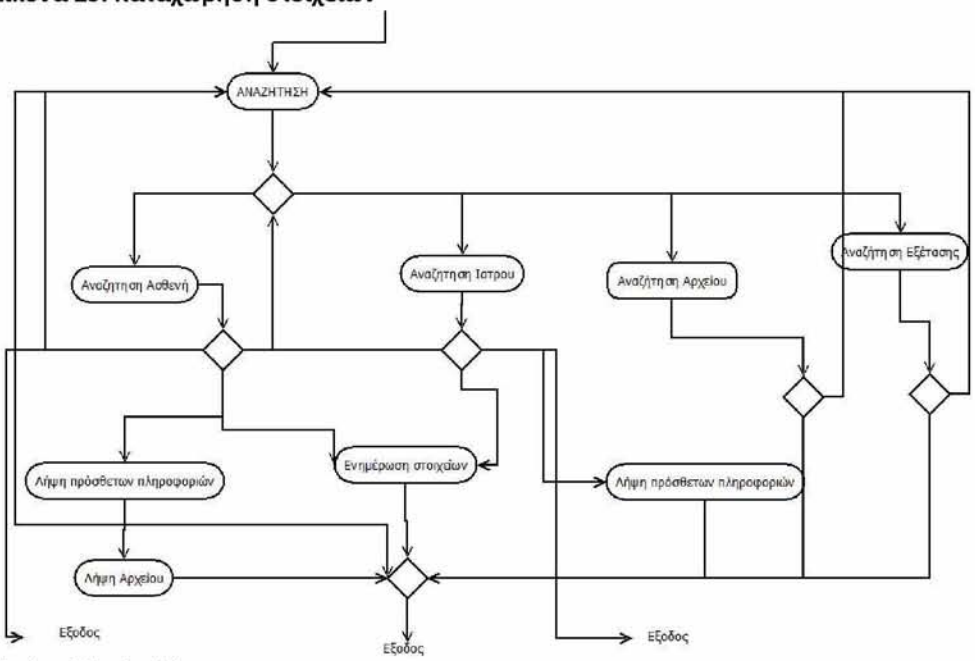
```
</head>
<body>
<?php
echo "<center>";
echo "Log out successfully... ";
echo "Please close your browser.";
echo "</center>";
include ("login.html");

?>

</body>
</html>
```

ΠΑΡΑΡΤΗΜΑ Γ: Διαγράμματα UML

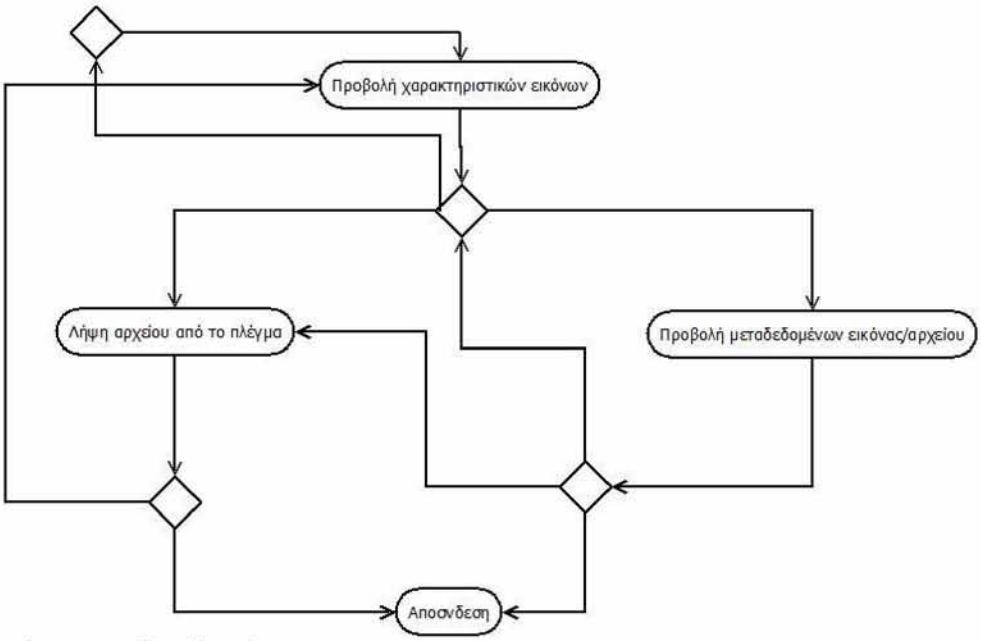
Τα διαγράμματα δραστηριοτήτων για τις περιπτώσεις καταχώρηση ασθενή, αναζήτηση και συλλογή εικόνων:



Εικόνα 25: Καταχώρηση στοιχείων



Εικόνα 26: Αναζήτηση



Εικόνα 27: Συλλογή εικόνων