ΙΑΤΡΙΚΗ ΣΧΟΛΗ
ΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ‘’ΑΤΤΙΚΟΝ’’
Β’ ΕΡΓΑΣΤΗΡΙΟ ΑΚΤΙΝΟΛΟΓΙΑΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Επεμβατική Ακτινολογία - Προεπεμβατικές και Μετεπεμβατικές Απεικονιστικές Μέθοδοι»

Αυτοματοποιημένη εξαγωγή μεταδεδομένων από πολλαπλές ιατρικές εικόνες μέσω τοπικού δικτύου

Διπλωματική εργασία

Δημήτρης Παπαμιχαήλ
Πυρηνικός Ιατρός

Επιβλέπων: Ε. Ευσταθόπουλος, Καθηγητής Ιατρικής Φυσικής-Ακτινοφυσικής

Αθήνα, 2016

Abstract

Explosive growth in the number of biomedical images in recent years requires new techniques to manage the information collected. Additional scan-related information, such as the acquisition parameters and filtration, is stored in DICOM metadata, while additional patient-related information, such as medical history or symptoms, is usually stored in a separated Database Management System (DBMS). Picture archiving and communication systems (PACS) can address the image data management issue, but PACS generally lack any methods for searching images and querying the metadata based on DBMS records. The objective of this study is to develop an automated method to address the problem of requiring biomedical image metadata from PACS over local network.

A client application has been designed to make requests to the database of the remote server. Algorithms have been developed to query the remote PACS server for the specified DICOM files and extract all metadata. The application allows users to enter queries and store the results in a Microsoft Excel spreadsheet. This project has placed increasing emphasis on the security aspect of the patient's personal data. The extracted information is automatically formatted and presented to the authorized end user as a Microsoft Excel file for data and trend analysis. The integrity and time efficacy of the method has been evaluated with a test sample of 171 DICOM files, while the accuracy of the data has been manually validated.

Περίληψη

Η εκρηκτική αύξηση του αριθμού των βιοϊατρικών εικόνων τα τελευταία χρόνια απαιτεί νέες τεχνικές για τη διαχείριση των πληροφοριών που συλλέγονται. Πρόσθετες απεικονιστικές πληροφορίες, όπως οι παράμετροι σάρωσης και φιλτραρίσματος, αποθηκεύονται ως DICOM μεταδεδομένα, ενώ πρόσθετες πληροφορίες που σχετίζονται με τον ασθενή, όπως το ιατρικό ιστορικό, αποθηκεύονται συνήθως σε ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS). Τα συστήματα επικοινωνίας και αρχειοθέτησης εικόνων (PACS) αντιμετωπίζουν το ζήτημα της διαχείρισης των δεδομένων, αλλά αδυνατούν να πραγματοποιήσουν την αναζήτηση εικόνων και την εξαγωγή των αντίστοιχων μεταδεδομένων με βάση τις εγγραφές στο σύστημα DBMS. Ο στόχος αυτής της μελέτης είναι να αναπτυχθεί μια αυτοματοποιημένη μέθοδος για την αντιμετώπιση του προβλήματος της εξαγωγής μεταδεδομένων εικόνων από το σύστημα PACS μέσω τοπικού δικτύου.

Σχεδιάστηκε μια εφαρμογή για απομακρυσμένο υπολογιστή, η οποία δημιουργεί και αποστέλλει αιτήματα στη βάση δεδομένων του απομακρυσμένου διακομιστή. Αναπτύχθηκαν αλγόριθμοι για την αναζήτηση των επιθυμητών αρχείων DICOM στον απομακρυσμένο διακομιστή PACS και την εξαγωγή των αντίστοιχων μεταδεδομένων. Η εφαρμογή επιτρέπει στους χρήστες να διαμορφώσουν αιτήματα και να αποθηκεύσουν τα αποτελέσματα σε ένα υπολογιστικό φύλλο εργασίας Microsoft Excel. Κατά τη διεκπεραίωση της μελέτης δόθηκε ιδιαίτερη έμφαση στην διασφάλιση των προσωπικών δεδομένων των ασθενών. Οι πληροφορίες που εξάγονται ταξινομούνται αυτόματα σε αρχείο Microsoft Excel και διατίθενται στον εξουσιοδοτημένο τελικό χρήστη σε ηλεκτρονική μορφή για περαιτέρω ανάλυση. Η ακεραιότητα και η αποτελεσματικότητα της μεθόδου αξιολογήθηκε μετά την αίτηση 171 αρχείων DICOM, ενώ η ακρίβεια των στοιχείων επιβεβαιώθηκε ποιοτικά.

Περιεχόμενα

Εισαγωγή

Τα σύγχρονα τμήματα ιατρικής απεικόνισης διαθέτουν ηλεκτρονικά συστήματα αποθήκευσης των προσωπικών στοιχείων και ιατρικών εικόνων του ασθενή. Η μεταφορά και η αποθήκευση μιας ιατρικής εικόνας γίνεται σύμφωνα με το ευρέως διαδεδομένο σύστημα αρχειοθέτησης εικόνων και επικοινωνίας (Picture Archiving & Communication System – PACS) [1]. Τα δίκτυα PACS επιτρέπουν την αρχειοθέτηση των εικόνων, χρησιμοποιώντας συσκευές προσωρινής ή μακροπρόθεσμης αποθήκευσης, ενώ παράλληλα επιτρέπουν την επικοινωνία και τη μεταφορά πληροφοριών μεταξύ τερματικών σταθμών υπολογιστών και απομακρυσμένων απεικονιστικών συστημάτων. Το πρωτόκολλο που χρησιμοποιείται κυρίως στα δίκτυα PACS είναι το πρότυπο Ψηφιακής Απεικόνισης και Επικοινωνίας στην Ιατρική (DICOM - Digital Imaging and Communications in Medicine). Σύμφωνα με αυτό, προσωπικά στοιχεία του ασθενή, στοιχεία της απεικονιστικής μεθόδου και του απεικονιστικού συστήματος αποθηκεύονται ως μεταδεδομένα (metadata) και αντιμετωπίζονται από το πρότυπο ως προτυποποιημένα αντικείμενα με αντίστοιχες ιδιότητες ή γνωρίσματα.

Όταν προκύπτει η ανάγκη εξαγωγής των μεταδεδομένων από την απεικόνιση ενός ασθενή, για παράδειγμα σε μία μελέτη δοσιμετρίας [2,3,4,5] ο ενδιαφερόμενος ανατρέχει στη βάση δεδομένων του PACS και αναζητά τον ασθενή. Στη συνέχεια, εξάγει τα μεταδεδομένα της αντίστοιχης εικόνας ή των αντίστοιχων τομών (εάν πρόκειται για τομογραφική απεικόνιση) και τα καταγράφει σε ηλεκτρονική μορφή για περαιτέρω επεξεργασία. Τα δίκτυα PACS δεν προσφέρουν αυτοματοποιημένα εργαλεία εξαγωγής μεταδεδομένων [6], με αποτέλεσμα η παραπάνω διαδικασία να γίνεται ιδιαίτερα περίπλοκη και χρονοβόρα, όταν προκύπτει η ανάγκη εξαγωγής των μεταδεδομένων πολλών ασθενών.

Σκοπός του εγχειρήματος ήταν η ανάπτυξη μιας μεθόδου για την αυτοματοποιημένη εξαγωγή μεταδεδομένων από ένα σύνολο ασθενών με κοινά στοιχεία εγγραφής, μέσω τοπικού δικτύου.

Μέθοδος

Γλώσσα προγραμματισμού JAVA

Η Java είναι μια σύγχρονη αντικειμενοστραφής (object oriented) γλώσσα προγραμματισμού, η οποία αναπτύχθηκε το 1991 από την εταιρεία Sun Microsystems. Η ανάπτυξή της πυροδοτήθηκε από τη ραγδαία διείσδυση του διαδικτύου και, ως ένα βαθμό, στόχευε στην κάλυψη των προγραμματιστικών αναγκών που άρχισαν να δημιουργούνται. Σε σχέση με τις άλλες γλώσσες προγραμματισμού, είχε δυο σημαντικούς νεοτερισμούς: αφενός μεν μπορούσε να τη χρησιμοποιήσει κανείς ελεύθερα χωρίς περιορισμούς και κόστος, αφετέρου δε, χάρη στη χρήση της εικονικής μηχανής, μπορούσε να εγκατασταθεί σε οποιοδήποτε λειτουργικό σύστημα.

Η Java προσφέρει ένα ολοκληρωμένο περιβάλλον εκτέλεσης των εφαρμογών, που η Sun ονομάζει νοητή μηχανή ή αλλιώς Java Virtual Machine – JVM. Η νοητή αυτή μηχανή παρεμβάλλεται μεταξύ του προγράμματος σε Java και του λειτουργικού συστήματος, προσφέροντας έτσι στο πρόγραμμα έναν εικονικό υπολογιστή. Ο κώδικας της Java εκτελείται σε αυτόν τον νοητό υπολογιστή εγγυώντας μια σταθερή και προκαθορισμένη συμπεριφορά ανεξάρτητα από το λειτουργικό σύστημα και τον επεξεργαστή. Τα αρχεία πηγαίου κώδικα Java (.java) μεταγλωττίζονται σε αρχεία bytecode (.class) και όχι σε κώδικα μηχανής (machine code).Το Java Virtual Machine (JVM), μεταφράζει τις εντολές του κώδικα σε πραγματικό χρόνο και τις εκτελεί χρησιμοποιώντας τις βιβλιοθήκες του συγκεκριμένου λειτουργικού συστήματος.

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, δηλαδή ο χειρισμός σχετιζόμενων δεδομένων και των διαδικασιών που επενεργούν σε μια εφαρμογή γίνεται μέσω μίας δομής δεδομένων που τα περιβάλλει ως αυτόνομη οντότητα με ταυτότητα και δικά της χαρακτηριστικά. Αυτή η δομή δεδομένων καλείται αντικείμενο και αποτελεί ένα στιγμιότυπο στη μνήμη ενός σύνθετου τύπου δεδομένων, ονόματι κλάση. Οι κλάσεις στην Java αποτελούνται από τις μεθόδους

και τις μεταβλητές. Μέθοδοι λέγονται οι συναρτήσεις στις οποίες μπορεί να απευθύνεται το αντικείμενο μίας κλάσης. Μεταβλητές λέγονται τα δεδομένα που καθορίζουν την κατάσταση ενός αντικειμένου. Ένα άλλο βασικό χαρακτηριστικό της Java είναι η αυτόματη διαχείριση της μνήμης, γνωστή και ως "μονάδα συλλογής σκουπιδιών" (garbage collector). Η εικονική μηχανή της Java απελευθερώνει αυτόματα τη μνήμη που δεσμεύει η εφαρμογή.

Η γλώσσα Java επιλέχθηκε ως εργαλείο ανάπτυξης στην παρούσα μελέτη, γιατί είναι μια γλώσσα προγραμματισμού απλή, συμβατή με δικτυακά πρωτόκολλα, ασφαλής, ουδέτερη της υποκείμενης αρχιτεκτονικής και υψηλής απόδοσης [7].

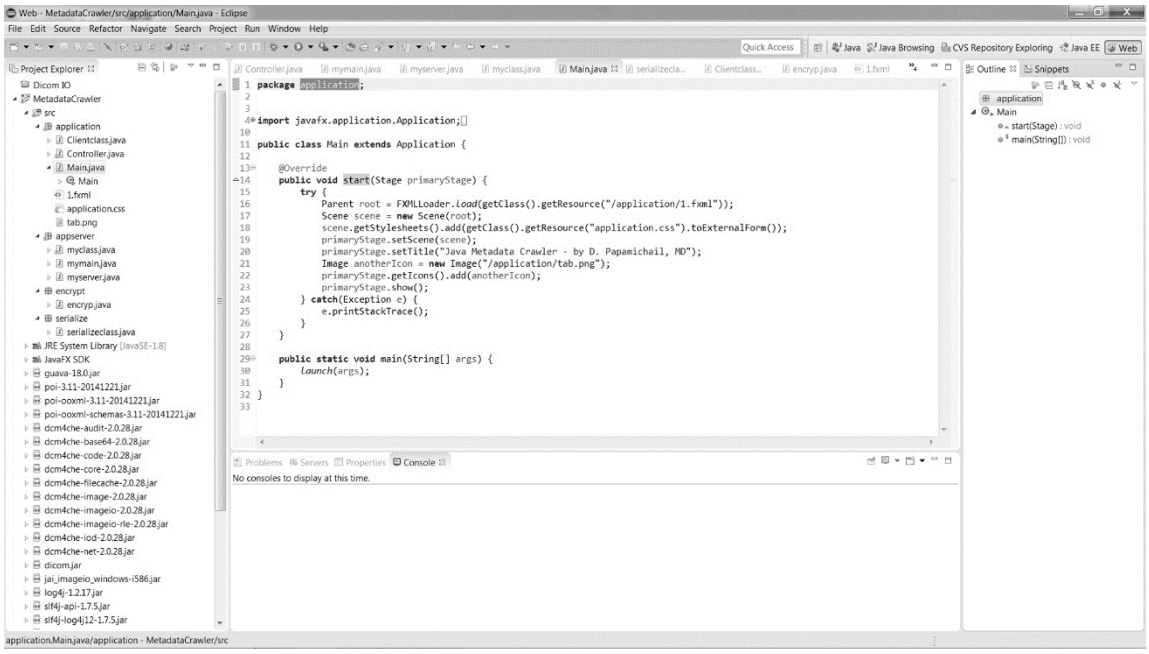
Περιβάλλον ανάπτυξης Eclipse

Το λογισμικό Eclipse είναι ένα πλήρως επεκτάσιμο, ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών (Integrated Development Environment- IDE). Το περιβάλλον αυτό καθώς και όλα τα υπόλοιπα προγράμματα που χρειάζονται για να αναπτυχθεί κώδικας σε Java είναι ελεύθερης διανομής (freeware) και ανοικτού κώδικα (open source), μετατρέποντας τελικά το Eclipse σε ένα δυναμικό και συνεχώς αναπτυσσόμενο εργαλείο.

Το πλαίσιο εργασίας (framework) του Eclipse [8], σχεδιάστηκε ως μια ανοιχτή πλατφόρμα με σκοπό την ανάπτυξη πλούσιων εφαρμογών «rich client applications». Η κεντρική ιδέα της αρχιτεκτονικής του Eclipse που συνέβαλε σημαντικά στην ανάπτυξη του, είναι τα βύσματα (plugins), μια ιδιαίτερη αρχιτεκτονική λογισμικού η οποία συμβάλει σε μια αρθρωτή και σταδιακή σχεδίαση που έχει ως στόχο να εμπλουτίσει την πλατφόρμα ανάπτυξης (IDE). Κάθε βύσμα χρησιμοποιεί και βελτιώνει τη λειτουργικότητα των άλλων βυσμάτων, μέσω ενός μηχανισμού, που είναι υπεύθυνο για τον καθορισμό των συσχετίσεων μεταξύ των διαφορετικών βυσμάτων καθώς και για το πώς και πότε αυτά θα εκτελεστούν. Με αυτόν τον τρόπο, δίνεται η δυνατότητα σε όσους αναπτύσσουν εργαλεία να προσφέρουν

επεκτάσεις στο Eclipse και να δημιουργήσουν τελικά ένα συνεπές και ολοκληρωμένο περιβάλλον για τους υπόλοιπους χρήστες.

Το γραφικό περιβάλλον του Eclipse (Εικόνα 1) είναι οργανωμένο σε όψεις (views), οδηγούς (wizards) και συντάκτες (editors). Τα στοιχεία αυτά παρέχουν τη βασική λειτουργικότητα που απαιτείται, προκειμένου να είναι εύκολη η ενσωμάτωση νέων στοιχείων γραφικών στην υπάρχουσα πλατφόρμα, δημιουργώντας επιπλέον τις προοπτικές (perspective), οι οποίες ουσιαστικά αποτελούν μια συλλογή από όψεις, συντάκτες κτλ, που απαιτούνται για να εκτελέσουν ένα συγκεκριμένο στόχο όπως για παράδειγμα η ανάπτυξη ενός προγράμματος. Το Eclipse προσφέρεται με προκαθορισμένες προοπτικές για την εξερεύνηση των πόρων, την ανάπτυξη εφαρμογών, την ανάπτυξη βυσμάτων και τη διόρθωση (debugging), ενώ νέες προοπτικές μπορούν να σχεδιαστούν στα πλαίσια των νέων βυσμάτων που δημιουργούνται.



Εικόνα 1. Το γραφικό περιβάλλον του Eclipse.

Για τον σκοπό της εργασίας εγκαταστάθηκε και χρησιμοποιήθηκε το περιβάλλον Eclipse IDE (έκδοση 4.4) σε λειτουργικό σύστημα Windows (έκδοση 7.1).

Πρωτόκολλο DICOM

Το πρωτόκολλο που χρησιμοποιείται κυρίως στα δίκτυα PACS είναι το πρότυπο Ψηφιακής Απεικόνισης και Επικοινωνίας στην Ιατρική (DICOM - Digital Imaging and Communications in Medicine). Το DICOM είναι ένα πρότυπο, το οποίο αναπτύχθηκε από κοινού από το American College of Radiology (ACR) και την National Electronics Manufacturer's Association (NEMA), λόγω της ανάγκης για την μεταφορά της πληροφορίας των ιατρικών εικόνων σε μια πρότυπη και συμβατή μορφή με όλους τους τύπους των απεικονιστικών συστημάτων [9]. Η πρώτη έκδοση του προτύπου (ACR-NEMA 1.0) ανακοινώθηκε το 1982 και ακολούθησε η δεύτερη έκδοση το 1989 (ACR-NEMA 2.0). Τα πρότυπα αυτά επανασχεδιάστηκαν, ώστε το 1993 να εκδοθεί το διεθνές πρότυπο Ψηφιακής Απεικόνισης και Επικοινωνίας στην Ιατρική (Digital Imaging and Communications in Medicine – DICOM).

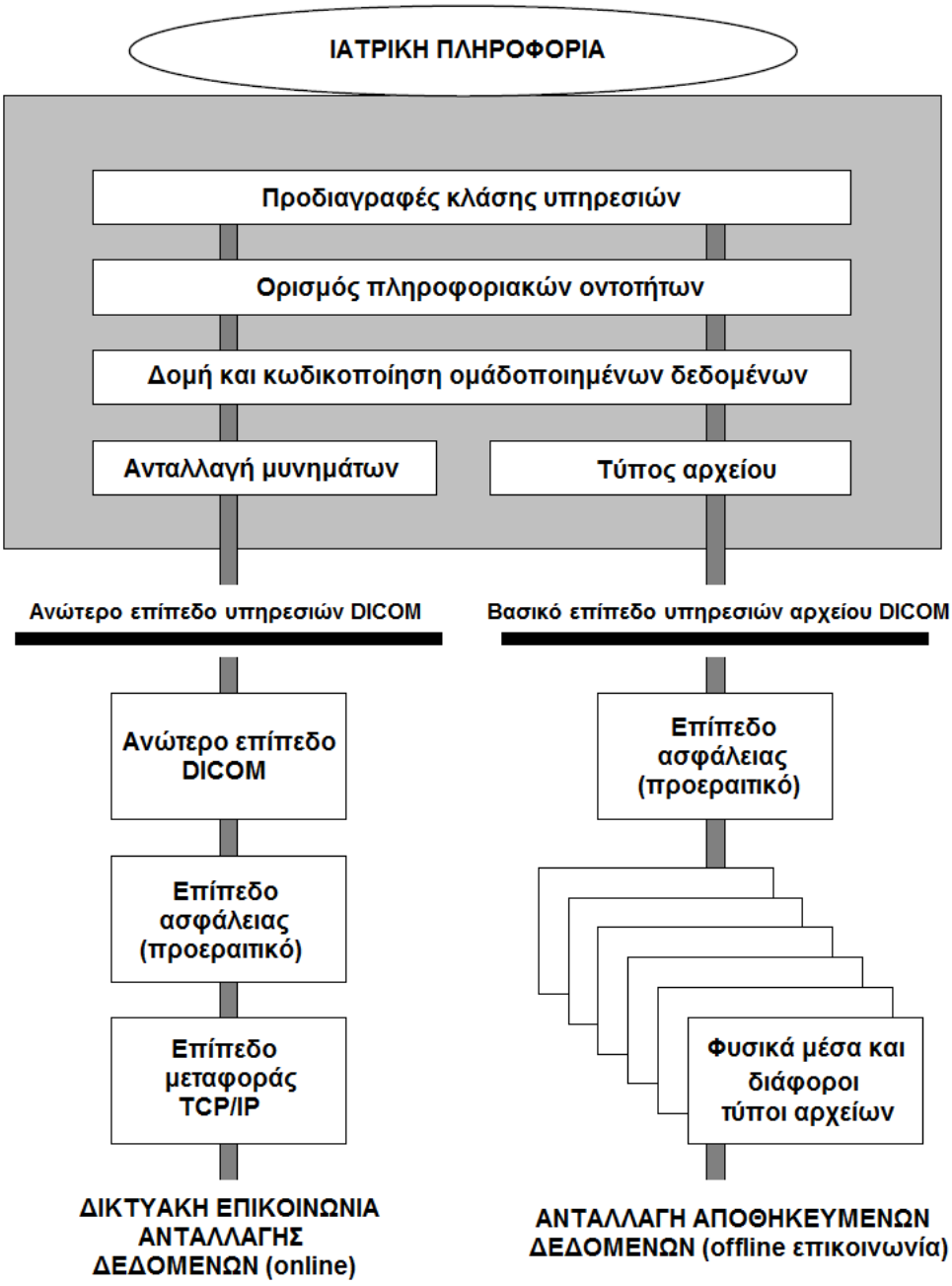
Το πρότυπο DICOM ορίζει ένα σύνολο κοινών κανόνων για την αποθήκευση και τη μεταφορά της πληροφορίας από μία μελέτη ιατρικής απεικόνισης [10]. Το πρότυπο DICOM περιλαμβάνει ένα σύνολο εντολών λογισμικού και ένα σύνολο υπηρεσιών επικοινωνίας, ασφάλειας και αποθήκευσης, τόσο της ιατρικής εικόνας όσο και των υπόλοιπων πληροφοριών μιας απεικονιστικής μελέτης (Εικόνα 1.9). Χαρακτηριστικό του προτύπου DICOM είναι η κατηγοριοποίηση των πληροφοριών σε ομάδες δεδομένων (data sets), επιτρέποντας με αυτόν τον τρόπο τη μεταφορά συνοδευτικών πληροφοριών με χαρακτηριστικές ετικέτες (tags) και την ενσωμάτωσή τους στο ίδιο αρχείο DICOM. Για παράδειγμα, μια σπινθηρογραφική εικόνα αποθηκεύεται σε ένα αρχείο DICOM μαζί με την ταυτότητα του ασθενούς, έτσι ώστε να αποφευχθούν τυχόν λάθη.

Σημαντικά στοιχεία του προτύπου DICOM είναι:

- οι πληροφοριακές οντότητες (Information Entities - IEs), οι οποίες ακολουθούν συγκεκριμένη ιεραρχία:
  - Ασθενής (Patient)
  - Διαγνωστική εξέταση (Study)

- Σειρά εξέτασης (Series)
- Στοιχείο εξέτασης (Object)

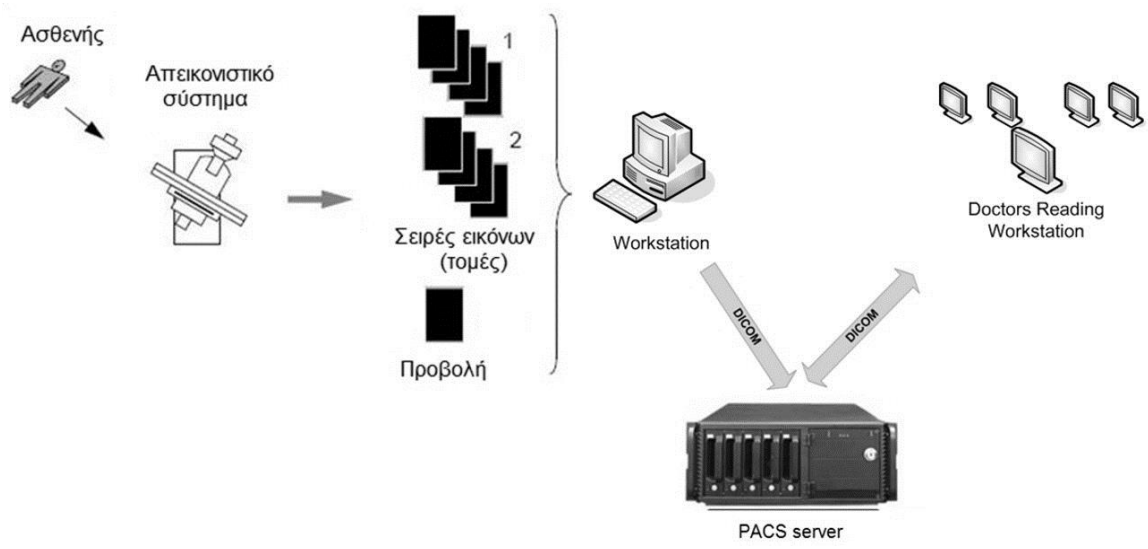
- οι ειδικές πληροφοριακές οντότητες, για παράδειγμα οι τελικές αναφορές (reports)
- τα μοναδικά αναγνωριστικά ταυτοποίησης (Unique Identifiers - UIDs), τα οποία ορίζουν τη μοναδικότητα μιας ιδιότητας ή ενός χαρακτηριστικού σε διαφορετικές χώρες, διαφορετικούς προμηθευτές και εξοπλισμούς
- η συμβατότητα με το Πρωτόκολλο Ελέγχου Μετάδοσης / Πρωτόκολλο του Internet (Transmission Control Protocol / Internet Protocol – TCP / IP)
- η υποστήριξη του εφαρμογής του σε περιβάλλον εκτός δικτύου (off-line) με τη χρήση εμπορικά διαθέσιμων συστημάτων αρχειοθέτησης, όπως το CDFS (Compact Disc File System) και το FAT (File Allocation Table) σε προσωπικούς υπολογιστές
- ο καθορισμός των εντολών και της συνάφειας των δεδομένων, χρησιμοποιώντας κλάσεις υπηρεσιών (Service Classes)
- η εξέλιξη της δομής του, σύμφωνα με τις οδηγίες του Διεθνούς Οργανισμού Τυποποίησης (International Organization for Standardization – ISO)
- η συνεχής εξέλιξη και ενημέρωση του προτύπου με διατήρηση της συμβατότητάς του με τις προηγούμενες εκδόσεις [11].



Εικόνα 2. Δομή προτύπου DICOM (Digital Imaging and Communications in Medicine (DICOM): Introduction and Overview. National Electrical Manufacturers Association. PS 3.1, 2011.)

Όταν επιθυμείται η επικοινωνία μεταξύ δύο υπολογιστικών συστημάτων και η ανταλλαγή δεδομένων με τη χρήση του προτύπου DICOM, αρχικά αποστέλλεται το αίτημα επικοινωνίας στο δίκτυο. Το πρωτόκολλο δικτύου ενημερώνει για την διαθεσιμότητα του δικτύου. Εάν το δίκτυο είναι διαθέσιμο, τότε ξεκινά μια συγκεκριμένη σειρά ενεργειών για να πραγματοποιηθεί η επικοινωνία μεταξύ των

υπολογιστικών συστημάτων. Το σύστημα, το οποίο αιτείται την επικοινωνία, ενημερώνει για το είδος των επιθυμητών ενεργειών, ενώ το σύστημα, το οποίο λαμβάνει την αίτηση, ενημερώνει για τη δυνατότητα υλοποίησης των επιθυμητών ενεργειών. Με την αρχική αυτή διαπραγμάτευση καθορίζονται οι δυνατότητες κάθε συστήματος και με ποιον τρόπο είναι εφικτή η ανταλλαγή των δεδομένων, ανάλογα με τα χαρακτηριστικά και το λογισμικό του κάθε συστήματος. Το πρότυπο DICOM διασφαλίζει την ομαλή ανταλλαγή πληροφοριών μεταξύ δύο, πιθανώς διαφορετικών λογισμικών. Ένα απεικονιστικό σύστημα ενσωματώνει τις εικόνες και τις πληροφοριακές οντότητες του ασθενή σε νέα αρχεία με βάση το πρότυπο DICOM και στη συνέχεια αποστέλλει τα αρχεία αυτά στο σύστημα αποθήκευσης PACS (Εικόνα 3).

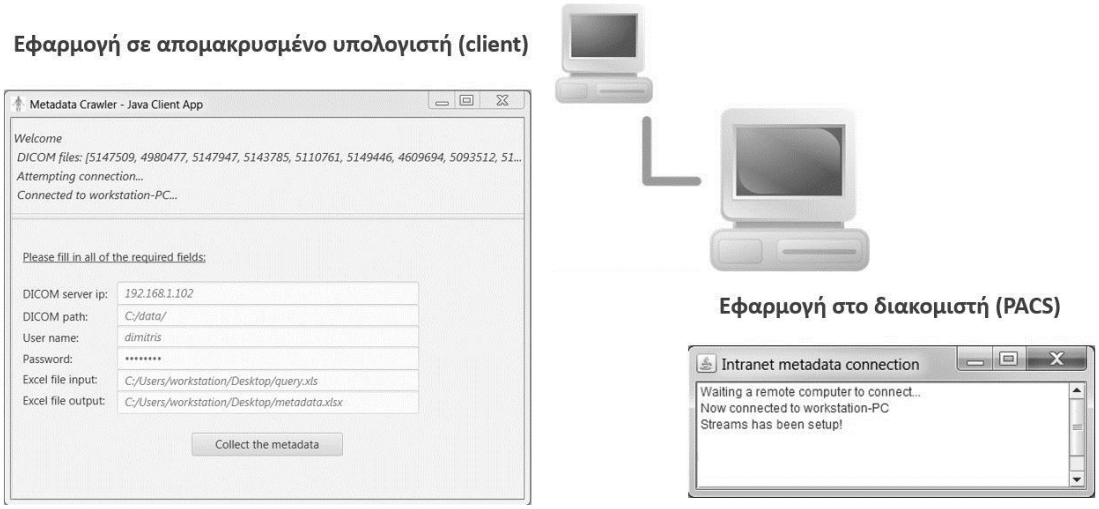


Εικόνα 3. Δημιουργία και διανομή των αρχείων απεικόνισης με βάση το πρωτόκολλο DICOM.

Η γλώσσα προγραμματισμού Java δεν αναγνωρίζει το πρωτόκολλο DICOM, αλλά υπάρχουν διαθέσιμα βύσματα (plugins) για το περιβάλλον Eclipse που καθιστούν το πρωτόκολλο DICOM συμβατό. Κατά την ανάπτυξη των εφαρμογών εξόρυξης μεταδεδομένων χρησιμοποιήθηκε το πακέτο dcm4che (έκδοση 2.0.28).

Σχεδιασμός της εφαρμογής για τον απομακρυσμένο υπολογιστή

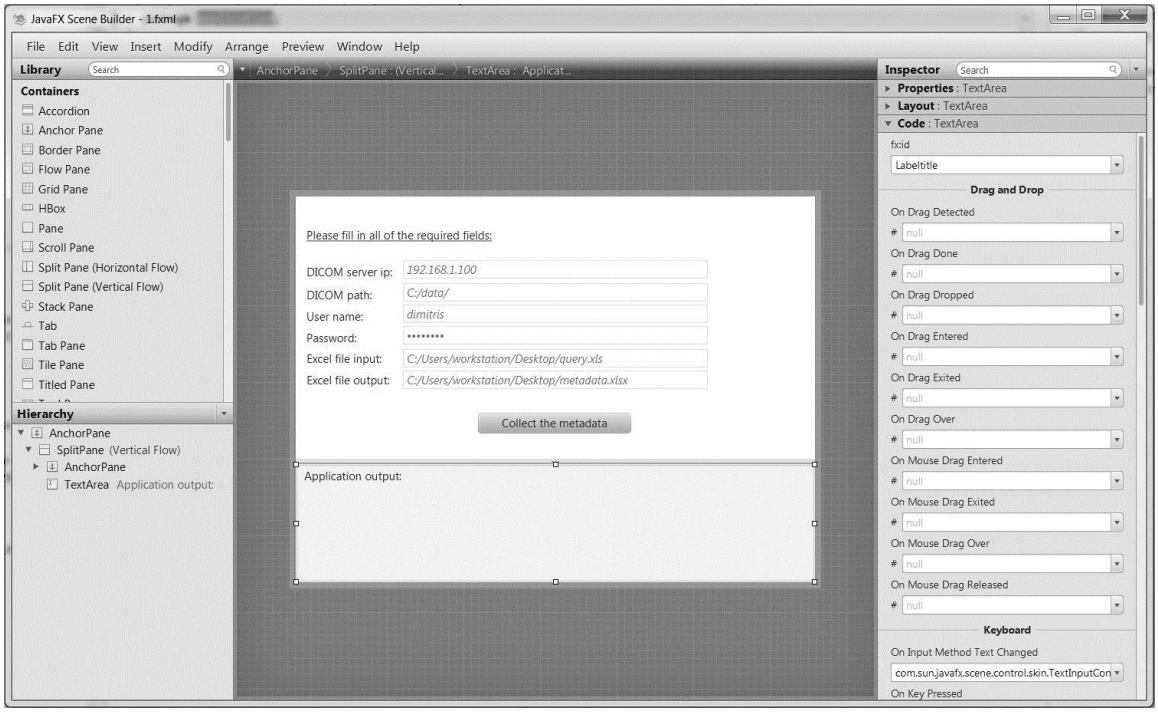
Σχεδιάστηκε ένα σύστημα επικοινωνίας μεταξύ PACS και ενός απομακρυσμένου υπολογιστή (client) μέσω τοπικού δικτύου (intranet), έτσι ώστε ο χρήστης να αναζητά τα επιθυμητά μεταδεδομένα από έναν απομακρυσμένο υπολογιστή, χωρίς την ανάγκη άμεσης πρόσβασης στον PACS (Εικόνα 4).



Εικόνα 4. Αναπαράσταση επικοινωνίας μεταξύ client και PACS.

Η εφαρμογή του απομακρυσμένου υπολογιστή (client) στέλνει στην εφαρμογή του PACS μια λίστα ασθενών και αναμένει την απάντηση από τον PACS, δηλαδή ένα υπολογιστικό φύλλο με όλα τα επιθυμητά μεταδεδομένα, ταξινομημένα ανά ασθενή. Εφόσον χρειάζεται να εισάγει ο χρήστης ορισμένα στοιχεία στην εφαρμογή client, όπως τη διεύθηνση IP (Internet Protocol) του PACS, σχεδιάστηκε μια διαδραστική γραφική διεπαφή χρήστη (Graphical User Interface - GUI) με τη βοήθεια της τεχνολογίας JavaFX. Η τεχνολογία JavaFX εγκαθίσταται πάνω στην υπάρχουσα αρχιτεκτονική της Java, επιτρέποντας τη δημιουργία πολυμεσικών εφαρμογών υψηλής ποιότητας [12]. Η αρχιτεκτονική της δομή περιλαμβάνει το

κοινό πλαίσιο εκτέλεσης των προγραμμάτων της Java, τις απαραίτητες επεκτάσεις για την εκτέλεση πολυμεσικών εφαρμογών (JavaFX Runtime) και το κοινό προγραμματιστικό περιβάλλον (Application Programming Interface - API) όπου, με τη χρήση του, όλα τα προγράμματα μπορούν να εκτελούνται σε οποιαδήποτε συσκευή υποστηρίζει την Java. Η εφαρμογή client ενσωματώνει την τεχνολογία JavaFX και σχεδιάστηκε με τη βοήθεια του υπολογιστικού εργαλείου Scene Builder (Εικόνα 5).



Εικόνα 5. Σχεδιασμός της εφαρμογής client με την τεχνολογία JavaFX Scene Builder.

Δημιουργία αιτήματος

Αφού εισάγει ο χρήστης τα απαραίτητα στοιχεία στο διαδραστική γραφική διεπαφή της εφαρμογή client και πατήσει το κουμπί της εφαρμογής, ενεργοποιείται ο προγραμματισμένος αλγόριθμος στον απομακρυσμένο υπολογιστή.

Η εφαρμογή θα εξάγει από ένα υπολογιστικό φύλλο εργασίας Excel τα ονόματα των ασθενών με τον κωδικό που αποθηκεύονται στον PACS. Ένα τέτοιο υπολογιστικό φύλλο εργασίας δύναται να δημιουργηθεί είτε από το χρήστη, είτε αυτόματα από ένα σύστημα διαχείρισης βάσης δεδομένων των ασθενών (Database Management System - DBMS), όπως είναι το λογισμικό (Evorad SA), το οποίο και χρησιμοποιήθηκε στην παρούσα μελέτη. Για την εξαγωγή της λίστας ασθενών σχεδιάστηκε μια κλάση, ονόματι ReadExcelFile:

```
public HashSet<String> ReadExcelFile (String Excelpath){

   HashSet<String> patientID =new HashSet<String>();
   FileInputStream fis = null;
   try {
        fis = new FileInputStream(Excelpath);
        HSSFWorkbook workbook = new HSSFWorkbook(fis);
        HSSFSheet sheet = workbook.getSheetAt(0);
        String columnWanted1 = "Sec. Patient ID";
        Integer columnNo1 = null;
        Row firstRow = sheet.getRow(0);
   for(Cell cell:firstRow){
    if (cell.getStringCellValue().equals(columnWanted1)){
            columnNo1 = cell.getColumnIndex();}
   for (Row row : sheet) {
    Cell c = row.getCell(columnNo1);
    String content = c.getRichStringCellValue().getString();
    patientID.add(content);}}}
            workbook.close();
        }        catch        (FileNotFoundException        e)
{e.printStackTrace();
          } catch (IOException e) {e.printStackTrace();
          } return patientID;
}
```

Σχεδιασμός της εφαρμογής για τον PACS

Η εφαρμογή στον PACS σχεδιάστηκε έτσι ώστε να δημιουργείται ένας δίαυλος επικοινωνίας μεταξύ του PACS και του απομακρυσμένου υπολογιστή μέσω τοπικού δικτύου. Σκοπός της εφαρμογής είναι καταρχάς η λήψη της λίστας ασθενών από την εφαρμογή client. Στη συνέχεια, η εφαρμογή στον PACS αναζητά τα αντίστοιχα αρχεία των ασθενών και όταν τα εντοπίσει, τα αποσυμπιέζει σε ένα νέο φάκελο του συστήματος. Για να ολοκληρωθεί η παραπάνω διαδικασία, δημιουργήθηκαν δύο κλάσεις ονόματι SearchPatients και Decompress:

```
public  Map<String, String> SearchPatients(String rootfolder){

File root = new File(rootfolder);
 File[] list = root.listFiles();
 if (list != null){
        for ( File f : list ) {
            if ( f.isDirectory() ) {
                searchdir(f.getAbsolutePath());
      dirlist.put(f.getName(),f.getAbsoluteFile().toString());
            }}}
return dirlist;
}

public void Decompress (String fullpath){

try {
Files.walk(Paths.get(fullpath)).forEach(filePath ->
if (Files.isRegularFile(filePath)) {
   String source = filePath.toString();
   String extension = source.substring(source.lastIndexOf(".")
   + 1, source.length());
if (extension.equals("gz")) {
    try {
```

```
    GZIPInputStream in = new GZIPInputStream(new
    FileInputStream(source));
    String target = fullpath+"/dicom/"
    +filePath.getFileName().toString();
if (target.indexOf(".") > 0)
    target = target.substring(0, target.lastIndexOf("."));
    OutputStream out = new FileOutputStream(target);
    byte[] buf = new byte[1024];
    int len;
while ((len = in.read(buf)) > 0) {
    out.write(buf, 0, len);}
in.close();
out.close();}
catch (IOException e) {e.printStackTrace();}
}}});}
catch (IOException e) {e.printStackTrace();}
     }
```

Εφόσον δημιουργηθεί ο φάκελος με τα επιθυμητά αρχεία DICOM, η εφαρμογή εξάγει τα μεταδεδομένα από κάθε μία εικόνα χωριστά και εισάγει τα μεταδεδομένα σε μια συλλογή διεπαφών Multimap. Η συλλογή αυτή της Java αποτελεί ένα σύνολο ζευγών κλειδιού-τιμών (key-value pairs) και επιτρέπει την αναζήτηση ορισμένων τιμών με βάση κάποιο κλειδί. Ως κλειδιά ορίστηκαν οι χαρακτηριστικές ετικέτες (tags) που βρέθηκαν σε κάθε αρχείο DICOM. Η κλάση που σχηματίστηκε για την εξαγωγή των μεταδεδομένων ονομάζεται ReadDicom:

```
public Multimap<String, String> ReadDicom (String fullpath){

DicomObject object = null;
String dcmfile = "Filename";
File root = new File(fullpath+"/dicom/");
File[] list = root.listFiles();
if (list == null) return null;
```

```
Multimap<String, String> mymap = LinkedListMultimap.create();
   for (int filenum=0;filenum<list.length;filenum++) {
   try {
    mymap.put(dcmfile , list[filenum].getName());
    DicomInputStream            dis            =            new
DicomInputStream(list[filenum]);
    object = dis.readDicomObject();
    dis.close();
    object.size();
    Iterator<DicomElement> iter = object.datasetIterator();
    while(iter.hasNext()) {
      DicomElement element = iter.next();
      int tag = element.tag();
      try {
        String tagName = object.nameOf(tag);
        String tagAddr = TagUtils.toString(tag);
        String tagVR = object.vrOf(tag).toString();
       if (tagVR.equals("SQ")) {
       if (element.hasItems()) {
       continue;
      }}
       String tagValue = object.getString(tag);
       if (!(tagVR.equals("UN"))) {
int blank=Math.max(0,(filenum+1) - mymap.get(tagName).size());
       for (int count=1;count<blank;count++)
       {mymap.put(tagName,"");}
        mymap.put(tagName,tagValue);}
} catch (Exception e) {
e.printStackTrace();
System.out.println(e.getMessage());
System.exit(0);}}
} catch (Exception e) {
System.out.println(e.getMessage());
System.exit(0);
}}
return mymap;
}
```

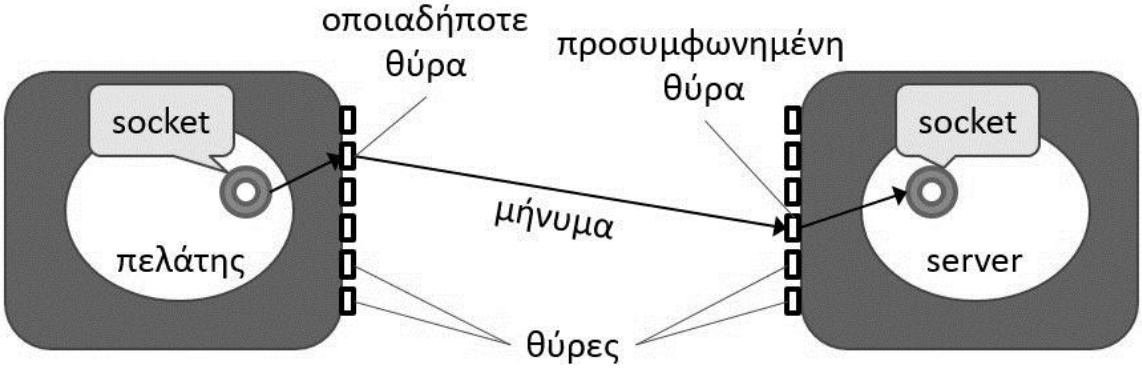
Η συλλογή Multimap με τα μεταδεδομένα από όλες τις εικόνες DICOM ενός ασθενή δημιουργεί μια νέα καρτέλα υπολογιστικού φύλλου εργασίας Excel (Microsoft Corporation, Redmond, WA). Όταν ολοκληρωθεί η διαδικασία αυτή για κάθε ασθενή, η εφαρμογή του PACS αποθηκεύει προσωρινά στον σκληρό δίσκο ένα αρχείο Excel με όλες τις καρτέλες των ασθενών. Οι υπεύθυνες κλάσεις για τη δημιουργία του αρχείου Excel ονομάζονται WriteSheet και WriteExcelFile:

```
public void WriteSheet (Multimap<String, String> mymap,
XSSFWorkbook workbook, String datasetname){

XSSFSheet sheet = workbook.createSheet(datasetname);
XSSFRow row;
Set<String> keyset = mymap.keySet();
int rowid = 0;
   for (String key : keyset){
     row = sheet.createRow(rowid++);
     int cellid = 0;
     Cell cellkey = row.createCell(cellid++);
     cellkey.setCellValue((String)key);
     Collection<String> keyvalues = mymap.get(key);
          for (String tagValue : keyvalues){
            Cell cellvalues = row.createCell(cellid++);
            cellvalues.setCellValue((String)tagValue);
           }}}

public void WriteExcelFile(String Outputfile, XSSFWorkbook
workbook){

try{
   FileOutputStream out = new FileOutputStream(new
   File(Outputfile));
   workbook.write(out);
```

```
   out.close();
   workbook.close();
}catch (Exception e)
{e.printStackTrace();}
}
```

Επικοινωνία μέσω τοπικού δικτύου

Το ευρύτερα διαδεδομένο μοντέλο ανάπτυξης δικτυακών εφαρμογών είναι το μοντέλο του πελάτη - εξυπηρετητή (client - server) [13]. Ο εξυπηρετητής αποτελεί έναν υπολογιστή, ο οποίος αναμένει να συνδεθεί σε αυτόν ένας απομακρυσμένος υπολογιστής (client) και να του παράσχει υπηρεσίες. Όταν μια διεργασία στον απομακρυσμένο υπολογιστή αρχίζει να εκτελείται, αποστέλλεται ένα αίτημα μέσω του δικτύου στον εξυπηρετητή, ζητώντας του κάποιου είδους υπηρεσία (π.χ. μεταφορά αρχείου). Όταν ο εξυπηρετητής απαντήσει στο αίτημα, τότε εισέρχεται σε κατάσταση αναμονής, περιμένοντας για ένα καινούριο αίτημα. Ως σύνδεση ορίζεται ο επικοινωνιακός δίαυλος μεταξύ δύο διεργασιών.



Εικόνα 6. Δίαυλος επικοινωνίας μεταξύ πελάτη και εξυπηρετητή (client - server).

Η τοπική διεύθυνση και απομακρυσμένη διεύθυνση δικτύου, προσδιορίζουν την ταυτότητα των υποδικτύων και των υπολογιστών, στους οποίους εκτελούνται οι επικοινωνούσες διεργασίες. Η τοπική διεργασία και απομακρυσμένη διεργασία προσδιορίζουν την ταυτότητα των διεργασιών που θα επικοινωνούν. Ως υποδοχή (socket) ορίζεται το ένα άκρο, από έναν επικοινωνιακό δίαυλο διπλής κατεύθυνσης, μεταξύ δύο διεργασιών που εκτελούνται στο δίκτυο. Το socket προσδιορίζει το πρωτόκολλο, τη διεύθυνση και τον αριθμό θύρας του άκρου (Εικόνα 6). Ο επικοινωνιακός δίαυλος δημιουργείται από την πλευρά του εξυπηρετητή με την κλάση ServerConnection και από την πλευρά του client με την κλάση ClientConnection:

```
public void ServerConnection(){

try{
server = new ServerSocket(7010, 1);
  while(true){
  try{
    showMessage(" Waiting a remote computer to connect...
\n");
    connection = server.accept();
    showMessage("Connected to " + connection.getInetAddress()
    .getHostName());
    output = new ObjectOutputStream(connection
    .getOutputStream());
    output.flush();
     input = new ObjectInputStream(connection
     .getInputStream());
     showMessage("\n Streams has been setup! \n");
     whileConnect();
      }catch(EOFException eofException){
      showMessage("\n Server Ended Connection!");
      }finally{
      try{
        output.close();
```

```
        input.close();
         connection.close();
           }catch(IOException ioException){
                ioException.printStackTrace();
           }}}
           }catch(IOException ioException){
           ioException.printStackTrace();}
}

public void ClientConnection(serializeclass p){

String inputexcel = Textoutput.getText();
try{
 connection = new Socket(InetAddress.getByName(Textip
 .getText()),7010);
 Labeltitle.setText(Labeltitle.getText() + "\n Connected to "+
 connection.getInetAddress().getHostName()+"...");
 output                          =                          new
ObjectOutputStream(connection.getOutputStream());
 Labeltitle.setText(Labeltitle.getText()+"Streams set up...");
 output.writeObject(p);
 output.flush();
 Labeltitle.setText(Labeltitle.getText()+ "\n Object sent..."
 input = new ObjectInputStream(connection.getInputStream());
 …
 } catch (Exception e) {
 e.printStackTrace();}
 } catch (SocketException se) {
 se.printStackTrace();
 } catch (IOException e) {
 e.printStackTrace();
 }
 finally{
 try {
   output.close();
   input.close();
   connection.close();
```

```
    } catch (IOException e) {
    e.printStackTrace();}
}}
```

Σειριοποίηση αντικειμένων

Στην Java, όπως και στις γλώσσες προγραμματισμού C και C++, η εισαγωγή και η επεξεργασία των δεδομένων δεν υποστηρίζονται απευθείας από τη γλώσσα προγραμματισμού, αλλά από ένα εξωτερικό πακέτο κλάσεων. Το πακέτο java.io διαθέτει ένα σύνολο κλάσεων, γνωστές με το όνομα streams (ροές) και pipes (διασωληνώσεις), οι οποίες παρέχουν λειτουργικότητα στην ανάγνωση και στην εγγραφή των δεδομένων. Πρόκειται ουσιαστικά για ροές δεδομένων ή ένα κανάλι επικοινωνίας μεταξύ της κύριας πηγής κάποιας πληροφορίας και του προορισμού της. Η πληροφορία αυτή δύναται να προέρχεται από ένα αντικείμενο. Για την απόκτηση μιας πληροφορίας το πρόγραμμα ανοίγει μια ροή σε μια πηγή πληροφοριών και διαβάζει σειριακά την πληροφορία του αντικειμένου.

Η Java παρέχει ένα ενδογενή μηχανισμό σειριοποίησης, ο οποίος αναπαριστά την κατάσταση του αντικειμένου σε σειριοποιημένη μορφή με αρκετά λεπτομερή τρόπο, έτσι ώστε το αντικείμενο να μπορεί να ανακατασκευαστεί αργότερα [14]. Για τη σειριοποίηση του αντικειμένου εφαρμόζεται η διασύνδεση java.io.Serializable και ορίζεται ένας μοναδικός ενδείκτης serial versionID σαν κωδικοποιημένο αποτύπωμα του ονόματος της τάξης και όλων των χαρακτηριστικών των εφαρμοσμένων διασυνδέσεων που περιλαμβάνει.

Η εφαρμογή client σχηματίζει ένα αντικείμενο με τη λίστα ασθενών και τη διαδρομή του φακέλου με όλους τους ασθενείς (path) στον PACS. Η εφαρμογή στον εξυπηρετητή λαμβάνει το σειριοποιημένο αντικείμενο και το ανασυνθέτει. Εφόσον η υπεύθυνη κλάση για τη σειριοποίηση και την ανασύνθεση του αντικειμένου περιέχει μια ορισμένη μέθοδο κατασκευής (constructor) και μια ορισμένη μέθοδο καταστροφής (destructor), οφείλει να είναι η ίδια τόσο στην εφαρμογή client όσο και στην εφαρμογή του εξυπηρετητή. Για το σκοπό αυτό δημιουργήθηκε ένα αρχείο

JAR (Java ARchive) με την υπεύθυνη κλάση, προσβάσιμη και από τις δύο εφαρμογές. Η κλάση για τη σειριοποίηση και την ανασύνθεση του αντικειμένου ονομάστηκε SerializeClass:

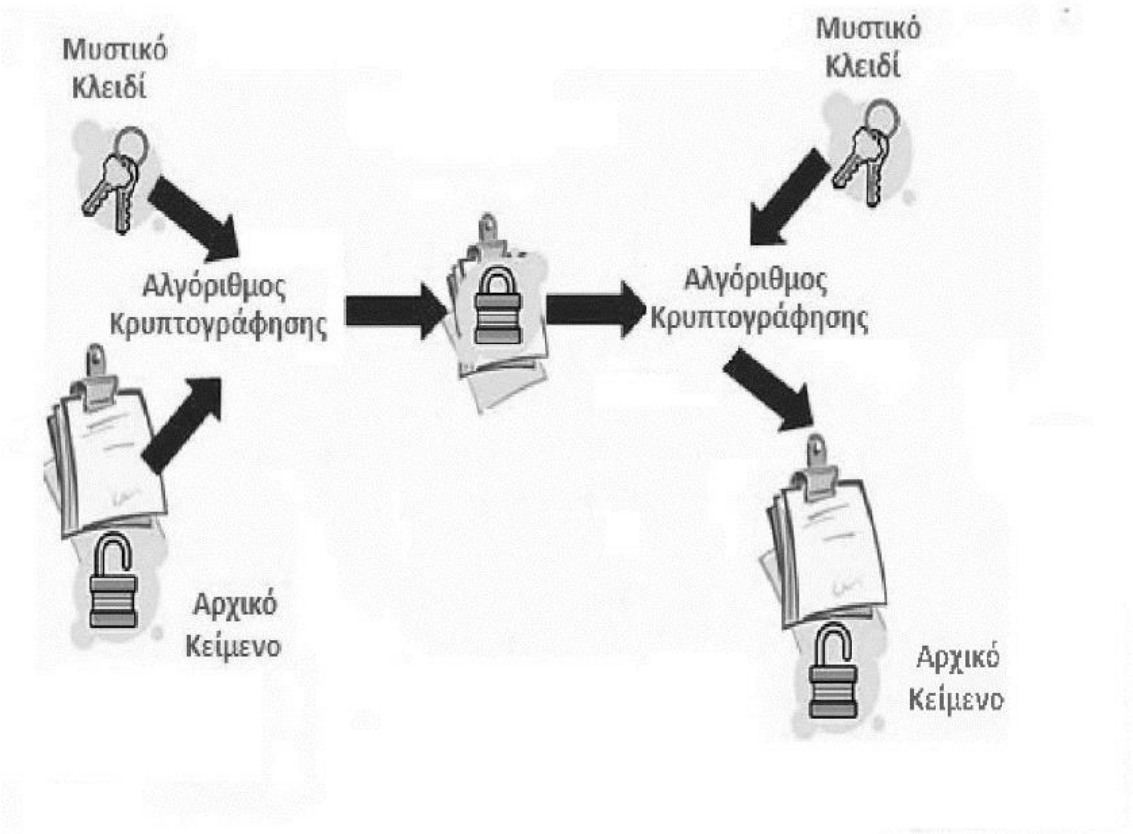
```
public class SerializeClass implements Serializable {

private static final long serialVersionUID = *****;
private String myroot;
private HashSet<String> mylist;

public SerializeClass(String myroot,HashSet<String> mylist){
this.myroot = myroot;
this.mylist = mylist;
}
public String getmyroot(){
return myroot;
}
public void setmyroot(String myroot){
this.myroot = myroot;
}
public HashSet <String> getmylist(){
return mylist;
}
public void setmylist(HashSet <String> mylist){
this.mylist = mylist;
}}
```

Κρυπτογράφηση δεδομένων

Τα δεδομένα σχετικά με την υγεία του ασθενή αποτελούν μέρος της προσωπικότητας του ατόμου και όχι ιδιοκτησία του φορέα που τα συλλέγει ή τα επεξεργάζεται [15]. Επομένως, η επεξεργασία των δεδομένων οφείλει να συνάδει

με τις σχετικές διατάξεις για την προστασία των προσωπικών ευαίσθητων δεδομένων και του ιατρονοσηλευτικού απορρήτου [16]. Εφόσον οι εφαρμογές που αναπτύχθηκαν διακινούν προσωπικά ευαίσθητα δεδομένα μέσω τοπικού δικτύου [17], εφαρμόστηκαν μέθοδοι κρυπτογράφησης για την διασφάλιση της εμπιστευτικότητας και της ακεραιότητας της πληροφορίας.

Κρυπτογράφηση (encryption) είναι η διεργασία μετασχηματισμού ενός μηνύματος σε μια ακατανόητη μορφή με τη χρήση ενός κρυπτογραφικού αλγόριθμου, έτσι ώστε αυτό να μην είναι αναγνώσιμο από τρίτα μέρη (εκτός του νόμιμου παραλήπτη). Η αποκρυπτογράφηση (decryption) είναι η διεργασία ανάκτησης του αρχικού μηνύματος (αναγνώσιμη μορφή) από μια ακατανόητη έκδοσή του που είχε παραχθεί μέσω μιας διεργασίας κρυπτογράφησης. Η κρυπτογράφηση και αποκρυπτογράφηση ενός μηνύματος γίνεται με βάση τη βοήθεια ενός αλγορίθμου κρυπτογράφησης και ενός κλειδιού κρυπτογράφησης (Εικόνα 7).



Εικόνα 7. Κρυπτογράφηση και αποκρυπτογράφηση δεδομένων.

Στην παρούσα μελέτη χρησιμοποιήθηκε το πρότυπο AES για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων. Το πρότυπο AES περιγράφει μια συμμετρική διαδικασία κρυπτογράφησης μυστικού κλειδιού, δηλαδή ο πομπός και ο δέκτης χρησιμοποιούν το ίδιο κλειδί για κρυπτογράφηση και αποκρυπτογράφηση [18]. Ανεξάρτητα από το μήκος του κλειδιού, ο αλγόριθμος επενεργεί πάνω σε τμήματα δεδομένων μεγέθους 128 bits. Η διαδικασία κρυπτογράφησης είναι επαναληπτική, δηλαδή σε κάθε τμήμα δεδομένων γίνεται μια επεξεργασία, η οποία επαναλαμβάνεται τόσες φορές όσες και το μέγεθος του κλειδιού. Σε κάθε επανάληψη επεξεργασίας ως είσοδος είναι το τμήμα δεδομένων, όπως έχει προκύψει από τον προηγούμενο γύρο επεξεργασίας καθώς και ένα κλειδί που έχει παραχθεί από το αρχικό με βάση κάποια διαδικασία που ορίζει ο αλγόριθμος. Το τελικό προϊόν της επεξεργασίας είναι το κρυπτογραφημένο τμήμα δεδομένων, διατηρώντας το αρχικό μέγεθος (128 bits). Σημειώνεται ότι η ασφάλεια της συμβατικής κρυπτογραφίας στηρίζεται μόνον στη μυστικότητα του κλειδιού και όχι στη μυστικότητα του αλγορίθμου που χρησιμοποιείται. Στην παρούσα μελέτη δημιουργήθηκε μία κλάση με το όνομα Encryption για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων που διακινούνται μέσω τοπικού δικτύου:

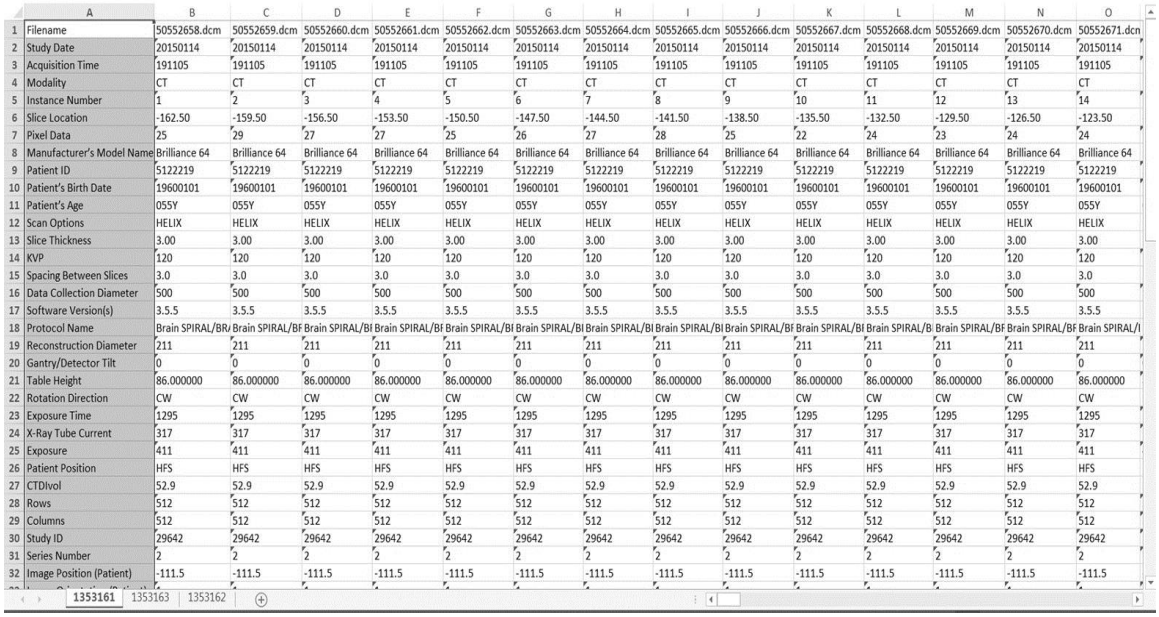
```
public class Encryption {

            private static final String ALGO = "AES";
            private static final byte[] keyValue =
            new byte[] { ********** };

     public static byte[] encrypt(byte[] Data) throws
Exception {            Key key = generateKey();
             Cipher c = Cipher.getInstance(ALGO);
             c.init(Cipher.ENCRYPT_MODE, key);
             byte[] encVal = c.doFinal(Data);
             return encVal;
```

```
        }
    public static byte[] decrypt(byte[] encryptedData) {
        Key key = generateKey();
        Cipher c = Cipher.getInstance(ALGO);
        c.init(Cipher.DECRYPT_MODE, key);
        byte[] decValue = c.doFinal(encryptedData);
        return decValue;
    }
    private static Key generateKey() throws Exception {
        Key key = new SecretKeySpec(keyValue, ALGO);
        return key;}
}
```

Αποτελέσματα

Δύο εφαρμογές σχεδιάστηκαν και δοκιμάστηκαν με επιτυχία για την εξαξωγή μεταδεδομένων από πολλαπλές ιατρικές εικόνες μέσω τοπικού δικτύου. Η εφαρμογή client εγκαταστάθηκε σε έναν απομακρυσμένο υπολογιστή, συνδεδεμένο με το τοπικό δίκτυο του νοσοκομείου. Η εφαρμογή server εγκαταστάθηκε στο κεντρικό σύστημα PACS του νοσοκομείου.

Ο χρήστης εισάγει τα απαραίτητα στοιχεία στην εφαρμογή client και πατώντας το κουμπί της εφαρμογής, αποστέλλει ένα αίτημα με τα ονόματα των ασθενών στην εφαρμογή server. Η εφαρμογή στον PACS λαμβάνει το αίτημα και αναζητά τα αντίστοιχα αρχεία DICOM στον τοπικό δίσκο. Εφόσον εντοπιστούν οι φάκελοι με τα ζητούμενα αρχεία, η εφαρμογή client διαβάζει τα μεταδεδομένα από κάθε μία εικόνα χωριστά και εισάγει τα μεταδεδομένα σε ένα υπολογιστικό φύλλο, ταξινομημένα ανά ασθενή (Εικόνα 8). Στη συνέχεια, το υπολογιστικό φύλλο με όλα τα επιθυμητά μεταδεδομένα κρυπτογραφείται και αποστέλλεται στην εφαρμογή client. Η εφαρμογή client αποκρυπτογραφεί το υπολογιστικό φύλλο και το αποθηκεύει στον τοπικό δίσκο του απομακρυσμένου υπολογιστή.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Filename	50552658.dcm	50552659.dcm	50552660.dcm	50552661.dcm	50552662.dcm	50552663.dcm	50552664.dcm	50552665.dcm	50552666.dcm	50552667.dcm	50552668.dcm	50552669.dcm	50552670.dcm	50552671.dcn
2	Study Date	20150114	20150114	20150114	20150114	20150114	20150114	20150114	20150114	20150114	20150114	20150114	20150114	20150114	20150114
3	Acquisition Time	191105	191105	191105	191105	191105	191105	191105	191105	191105	191105	191105	191105	191105	191105
4	Modality	CT	CT	CT	CT	CT	CT	CT	CT	CT	CT	CT	CT	CT	CT
5	Instance Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
6	Slice Location	-162.50	-159.50	-156.50	-153.50	-150.50	-147.50	-144.50	-141.50	-138.50	-135.50	-132.50	-129.50	-126.50	-123.50
7	Pixel Data	25	29	27	27	25	26	27	28	25	22	24	23	24	24
8	Manufacturer's Model Name	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64	Brilliance 64
9	Patient ID	5122219	5122219	5122219	5122219	5122219	5122219	5122219	5122219	5122219	5122219	5122219	5122219	5122219	5122219
10	Patient's Birth Date	19600101	19600101	19600101	19600101	19600101	19600101	19600101	19600101	19600101	19600101	19600101	19600101	19600101	19600101
11	Patient's Age	055Y	055Y	055Y	055Y	055Y	055Y	055Y	055Y	055Y	055Y	055Y	055Y	055Y	055Y
12	Scan Options	HELIX	HELIX	HELIX	HELIX	HELIX	HELIX	HELIX	HELIX	HELIX	HELIX	HELIX	HELIX	HELIX	HELIX
13	Slice Thickness	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
14	KVP	120	120	120	120	120	120	120	120	120	120	120	120	120	120
15	Spacing Between Slices	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
16	Data Collection Diameter	500	500	500	500	500	500	500	500	500	500	500	500	500	500
17	Software Version(s)	3.5.5	3.5.5	3.5.5	3.5.5	3.5.5	3.5.5	3.5.5	3.5.5	3.5.5	3.5.5	3.5.5	3.5.5	3.5.5	3.5.5
18	Protocol Name	Brain SPIRAL/BR/	Brain SPIRAL/BF	Brain SPIRAL/BI	Brain SPIRAL/BI	Brain SPIRAL/BI	Brain SPIRAL/BI	Brain SPIRAL/BI	Brain SPIRAL/BI	Brain SPIRAL/BI	Brain SPIRAL/BI	Brain SPIRAL/B	Brain SPIRAL/BF	Brain SPIRAL/BF	Brain SPIRAL/I
19	Reconstruction Diameter	211	211	211	211	211	211	211	211	211	211	211	211	211	211
20	Gantry/Detector Tilt	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	Table Height	86.000000	86.000000	86.000000	86.000000	86.000000	86.000000	86.000000	86.000000	86.000000	86.000000	86.000000	86.000000	86.000000	86.000000
22	Rotation Direction	CW	CW	CW	CW	CW	CW	CW	CW	CW	CW	CW	CW	CW	CW
23	Exposure Time	1295	1295	1295	1295	1295	1295	1295	1295	1295	1295	1295	1295	1295	1295
24	X-Ray Tube Current	317	317	317	317	317	317	317	317	317	317	317	317	317	317
25	Exposure	411	411	411	411	411	411	411	411	411	411	411	411	411	411
26	Patient Position	HFS	HFS	HFS	HFS	HFS	HFS	HFS	HFS	HFS	HFS	HFS	HFS	HFS	HFS
27	CTDIvol	52.9	52.9	52.9	52.9	52.9	52.9	52.9	52.9	52.9	52.9	52.9	52.9	52.9	52.9
28	Rows	512	512	512	512	512	512	512	512	512	512	512	512	512	512
29	Columns	512	512	512	512	512	512	512	512	512	512	512	512	512	512
30	Study ID	29642	29642	29642	29642	29642	29642	29642	29642	29642	29642	29642	29642	29642	29642
31	Series Number	2	2	2	2	2	2	2	2	2	2	2	2	2	2
32	Image Position (Patient)	-111.5	-111.5	-111.5	-111.5	-111.5	-111.5	-111.5	-111.5	-111.5	-111.5	-111.5	-111.5	-111.5	-111.5

1353161 1353163 1353162

Εικόνα 8. Υπολογιστικό φύλλο εργασίας με τα μεταδεδομένα από όλα τα αρχεία DICOM τριών ασθενών.

Προκειμένου να εξακριβωθεί η πιστότητα και η ακεραιότητα των εφαρμογών, δημιουργήθηκε μια λίστα τριών ασθενών, οι οποίοι είχαν ολοκληρώσει απεικόνιση εγκεφάλου με αξονικό τομογράφο. Η λίστα ασθενών στάλθηκε στην εφαρμογή server ως κρυπτογραφημένο αίτημα για την εξαγωγή των αντίστοιχων μεταδεδομένων. Η εφαρμογή server απέστειλε με επιτυχία το υπολογιστικό φύλλο στην εφαρμογή client. Το υπολογιστικό φύλλο περιείχε τα μεταδεδομένα 57 αρχείων DICOM για κάθε ασθενή, περιλαμβανομένου το αρχείο αναφοράς της δόσης (dose report). Στη συνέχεια, πραγματοποιήθηκε η αναζήτηση των τριών ασθενών και η εξαγωγή των αντίστοιχων μεταδεδομένων με άμεση πρόσβαση στο σύστημα PACS, χρησιμοποιώντας κατάλληλες υπολογιστικές ρουτίνες του λογισμικού MATLAB (MathWorks MatLab, έκδοση R2012b). Τα μεταδεδομένα των αρχείων από την άμεση πρόσβαση στο σύστημα PACS συγκρίθηκαν και ταυτοποιήθηκαν με τα μεταδεδομένα στο υπολογιστικό φύλλο ένα προς ένα. Με αυτόν τον τρόπο εξακριβώθηκε η ακεραιότητα των εφαρμογών και η πιστότητα των αποτελεσμάτων τους.

Συζήτηση

Όπως προαναφέρθηκε στην εισαγωγή, όταν προκύπτει η ανάγκη εξαγωγής των μεταδεδομένων ενός ασθενή, για παράδειγμα σε μία μελέτη δοσιμετρίας, ο ενδιαφερόμενος ανατρέχει στη βάση δεδομένων του PACS και αναζητά τον ασθενή. Στη συνέχεια, εξάγει τα μεταδεδομένα της αντίστοιχης εικόνας ή των αντίστοιχων τομών και τα καταγράφει σε ηλεκτρονική μορφή για περαιτέρω επεξεργασία (Εικόνα 9).

Εικόνα 9. Καθιερωμένη μέθοδος αναζήτησης και εξαγωγής μεταδεδομένων από την απεικόνιση ενός ασθενή.

Τα δίκτυα PACS δεν προσφέρουν αυτοματοποιημένα εργαλεία εξαγωγής μεταδεδομένων, με αποτέλεσμα η παραπάνω διαδικασία να γίνεται ιδιαίτερα περίπλοκη και χρονοβόρα, όταν προκύπτει η ανάγκη εξαγωγής των μεταδεδομένων πολλών ασθενών.

Οι εφαρμογές που δημιουργήθηκαν στην παρούσα μελέτη επιτρέπουν στον ενδιαφερόμενο να εξάγει τα επιθυμητά μεταδεδομένα από την απεικόνιση πολλών ασθενών με μία πιο απλή και σύντομη μέθοδο, χωρίς να απαιτείται η άμεση

πρόσβασή του στο σύστημα PACS. Το μόνο που χρειάζεται είναι η αναζήτηση των ασθενών με τα ζητούμενα κοινά στοιχεία εγγραφής, π.χ. εύρος ηλικίας, στο σύστημα διαχείρισης βάσης δεδομένων των ασθενών. Εφόσον η λίστα αυτή εισαχθεί στην εφαρμογή client, η εφαρμογή server θα αναζητήσει και θα εξάγει όλα τα επιθυμητά μεταδεδομένα αυτόματα. Ο ενδιαφερόμενος θα έχει στη διάθεσή του τα επιθυμητά δεδομένα, ταξινομημένα σε ένα υπολογιστικό φύλλο εργασίας, μέσα σε ελάχιστο χρονικό διάστημα.

Βιβλιογραφία

1. Liu BJ, Cao F, Zhou MZ, Mogel G, Documet L. Trends in PACS image storage and archive. Comput Med Imaging Graph. 2003;27(2-3):165-74.

2. Jahnen A, Kohler S, Hermen J, Tack D, Back C. Automatic computed tomography patient dose calculation using DICOM header metadata. Radiat Prot Dosimetry. 2011 Sep;147(1-2):317-20.

3. Källman HE, Halsius E, Folkesson M, Larsson Y, Stenström M, Båth M. Automated detection of changes in patient exposure in digital projection radiography using exposure index from DICOM header metadata. Acta Oncol. 2011 Aug;50(6):960-5.

4. Emanuelli S, Rizzi E, Amerio S, Fasano C, Cesarani F. Dosimetric and image quality comparison of two digital mammography units with different target/filter combinations: Mo/Mo, Mo/Rh, W/Rh, W/Ag. Radiol Med. 2011 Mar;116(2):310-8.

5. Mercuri M, Rehani MM, Einstein AJ. Tracking patient radiation exposure: challenges to integrating nuclear medicine with other modalities. J Nucl Cardiol.
2012 Oct;19(5):895-900.

6. Korenblum D, Rubin D, Napel S, Rodriguez C, Beaulieu C. Managing biomedical image metadata for search and retrieval of similar images. J Digit Imaging. 2011 Aug;24(4):739-48.

7. Jitngernmadan P, Miesenberger K. A Comparative Study on Java Technologies for Focus and Cursor Handling in Accessible Dynamic Interactions. Stud Health Technol Inform. 2015;217:267-73.

8. Gamma E, Beck K (eds). Contributing to Eclipse: Principles, Patterns, and Plugins. Addison-Wesley Longman. Amsterdam, 2003.

9. Spilker C. The ACR-NEMA Digital Imaging and communications Standard: a nontechnical description. J Digit Imaging. 1989 Aug;2(3):127-31.

10. Lynes J, Riha C. Learning the fundamentals of DICOM. Biomed Instrum Technol. 2004 Jan-Feb;38(1):35-8.

11. Horn RJ. DICOM additions and advances: the standards committee works to keep up with technology. Healthc Inform. 2004 Sep;21(9):48, 50.

12. Ignatchenko V, Ignatchenko A, Sinha A, Boutros PC, Kislinger T. VennDIS: a JavaFX-based Venn and Euler diagram software to generate publication quality figures. Proteomics. 2015 Apr;15(7):1239-44.

13. Koutelakis GV, Anastassopoulos GK, Lymberopoulos DK. Application of Multiprotocol Medical Imaging Communications and an Extended DICOM WADO Service in a Teleradiology Architecture. Int J Telemed Appl. 2012;2012:271758.

14. Knoll P, Höll K, Mirzaei S, Koriska K, Köhn H. Distributed nuclear medicine applications using World Wide Web and Java technology. Eur Radiol. 2000;10(9):1483-6.

15. Choe J, Yoo SK. Web-based secure access from multiple patient repositories. Int J Med Inform. 2008 Apr;77(4):242-8.

16. Pruski C, Wisniewski F. Efficient medical information retrieval in encrypted Electronic Health Records. Stud Health Technol Inform. 2012;180:225-9.

17. Ballance D. The network and its role in digital imaging and communications in medicine imaging. Vet Radiol Ultrasound. 2008 Jan-Feb;49(1 Suppl 1):S29-32.

18. Oh JY, Yang DI, Chon KH. A Selective Encryption Algorithm Based on AES for Medical Information. Healthc Inform Res. 2010 Mar;16(1):22-9.