



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**

**ΜΕΤΑΠΤΥΧΙΑΚΟ ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΨΗΦΙΑΚΩΝ  
ΣΥΣΤΗΜΑΤΩΝ**

**ΚΑΤΕΥΘΥΝΣΗ: ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ**

## **ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Ανάπτυξη πληροφοριακού συστήματος με τεχνολογία Workflow

«Διαδικασία υποβολής και εκτέλεσης  
ακτινολογικών εντολών σε υγειονομική περιφέρεια»

**ΦΟΙΤΗΤΡΙΑ: ΔΕΣΠΟΙΝΑ ΠΑΡΤΣΑΚΟΥΛΑΚΗ**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΓΕΩΡΓΙΟΣ ΒΑΣΙΛΑΚΟΠΟΥΛΟΣ**

Πειραιάς, Σεπτέμβριος 2009

## ΠΡΟΛΟΓΟΣ

Στα πλαίσια της παρούσας εργασίας, θα αναλυθεί ο σχεδιασμός και η κατασκευή ενός πληροφοριακού συστήματος, το οποίο υλοποιεί την διαδικασία υποβολής και εκτέλεσης ακτινολογικών εντολών σε υγειονομική περιφέρεια.

Αρχικά θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου, κύριο Γεώργιο Βασιλακόπουλο, για τις γνώσεις και συμβουλές που μου παρείχε όλο αυτό το διάστημα φοίτησης στο τμήμα, καθώς και όλους τους αρμόδιους καθηγητές που με επέλεξαν αρχικά να φοιτήσω σε αυτό το μεταπτυχιακό τμήμα.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑΣ

---

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ.....	7
ΕΙΣΑΓΩΓΗ.....	8
<b>1 ΣΥΣΤΗΜΑ ΡΟΗΣ ΕΡΓΑΣΙΑΣ – WORKFLOW SYSTEM.....</b>	<b>11</b>
1.1 Εισαγωγή.....	11
1.2 Οργανωτική Δομή Υγειονομικής Περιφέρειας.....	11
1.3 Διαδικασία – Δραστηριότητες – Ροή Συστήματος.....	13
1.3.1 Issue_Rad_request (Εισαγωγή Εντολής Ακτινολογικής Εξέτασης) .....	15
1.3.2 Lookup Rad Request (Διάβασμα – Έλεγχος Εντολών Ακτινολογικών Εξετάσεων)... ..	16
1.3.3 Check pending requests (Αναζήτηση εκκρεμών εντολών- εντολές που δεν έχουν καθοριστεί ραντεβού) .....	16
1.3.4 Schedule_Rad_Request (Καθορισμός ραντεβού και ακτινολόγου που θα εκτελέσει την εξέταση).....	16
1.3.5 Notify_patient (Ειδοποίηση ασθενούς και θεράποντος ιατρού μέσω e-mail).....	17
1.3.6 Search_Appointments (Έλεγχος των ραντεβού ακτινολόγου γιατρού) .....	17
1.3.7 Perform Rad Request (Εκτέλεση της ακτινογραφίας).....	18
1.3.8 Read_Old_Rad_Reports (Επισκόπηση παλιότερων ακτινολογικών αναφορών) .....	18
1.3.9 Charge Patient Account (Πληρωμή).....	19
1.3.10 Interpret Rad Results (Εισαγωγή τελικής γνωμάτευσης θεράποντος ιατρού) .....	19
<b>2 ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ – ΠΕΡΙΓΡΑΦΗ &amp; ΔΙΑΓΡΑΜΜΑΤΑ .....</b>	<b>21</b>
2.1 Εισαγωγή.....	21
2.2 Διαγράμματα μοντέλων της Βάσης Δεδομένων.....	22
2.2.1 Διάγραμμα μοντέλου βάσης δεδομένων συστήματος υποβολής και εκτέλεσης ακτινολογικής εντολής.....	22
2.2.2 Διάγραμμα μοντέλου βάσης δεδομένων ασφάλειας συστήματος.....	24
2.3 Σχεσιακό μοντέλο δεδομένων .....	25
2.3.1 Σχεσιακό μοντέλο συστήματος υποβολής και εκτέλεσης ακτινολογικής εντολής.....	25
2.3.2 Σχεσιακό μοντέλο συστήματος ασφάλειας .....	30
<b>3 ΕΓΚΑΤΑΣΤΑΣΗ – ΔΙΑΜΟΡΦΩΣΗ ΣΥΣΤΗΜΑΤΟΣ ΡΟΗΣ ΕΡΓΑΣΙΑΣ .....</b>	<b>33</b>
3.1 Εισαγωγή.....	33
3.2 Διαμόρφωση και αρχικοποίηση του συστήματος ροής εργασίας.....	34
3.3 Καθορισμός των δραστηριοτήτων στην buildtime βάση δεδομένων.....	44
3.3.1 Διαδικασίες ( PROCESSES) .....	44
3.3.2 Χρήστες και ρόλοι συστήματος (STAFF).....	50
3.3.3 Υλοποιήσεις (Implementations).....	52
3.4 EXPORT.....	58
3.5 Websphere MQ Workflow Client.....	59
<b>4 ΠΑΡΟΥΣΙΑΣΗ – ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ .....</b>	<b>61</b>
4.1 Εισαγωγή.....	61
4.2 Δομή προγραμμάτων – κρίσιμα σημεία ενδιαφέροντος.....	61
4.2.1 Σύνδεση με την βάση δεδομένων.....	61
4.2.2 Κλήση του workflow engine.....	62
4.2.3 Έλεγχος ορθής και επιτρεπτής πρόσβασης.....	63

4.2.4 Μεταφορά των δεδομένων μεταξύ των δραστηριοτήτων του συστήματος ροής εργασίας.....	64
4.2.5 Έξοδος προγράμματος.....	65
<b>5 ΑΝΑΛΥΣΗ ΦΟΡΜΩΝ.....</b>	<b>67</b>
5.1 Εισαγωγή εντολής ακτινολογικής εξέτασης.....	67
5.2 Λίστα Εντολών Ακτινολογικών Εξετάσεων.....	68
5.3 Εκκρεμείς Εντολές Ακτινολογικών Εξετάσεων.....	69
5.4 Καθορισμός Ραντεβού Εξέτασης.....	70
5.5 Ειδοποίηση ασθενούς – θεράπωντος ιατρού.....	72
5.6 Λίστα Ραντεβού Ακτινολογικών Εξετάσεων.....	75
5.7 Πληρωμή Ακτινολογικής Εξέτασης.....	76
5.8 Εκτέλεση ακτινολογικής Εξέτασης.....	77
5.9 Έλεγχος Ακτινολογικής Εξέτασης.....	80
5.10 Έλεγχος Παλαιότερων Ακτινολογικών Αναφορών Ασθενούς.....	82

## ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 0-1 Διαδικασία υποβολής και εκτέλεσης ακτινολογικών εντολών σε υγειονομική περιφέρεια.....	9
Εικόνα 1-1 Διάγραμμα υγειονομική περιφέρειας.....	12
Εικόνα 2-1 : E-R Diagram (1).....	22
Εικόνα 2-2 : E-R Diagram (2).....	24
Εικόνα 2-3: Σχισιακό Μοντέλο Υποσυστήματος.....	29
Εικόνα 2-4: Σχισιακό μοντέλο ασφάλειας συστήματος.....	31
Εικόνα 3-1: Εμπλεκόμενοι ενός συστήματος ροής εργασίας.....	34
Εικόνα 3-2: Βασικά μέρη του MQ Workflow.....	35
Εικόνα 3-3: Νέο Configuration ID.....	35
Εικόνα 3-4: Επιλογή όλων των components.....	36
Εικόνα 3-5: Configuration of Runtime Database.....	37
Εικόνα 3-6: Νέα σύνδεση με την βάση δεδομένων του workflow.....	37
Εικόνα 3-7: Connect parameters for runtime database.....	38
Εικόνα 3-8: Configuration του Queue Manager.....	39
Εικόνα 3-9: Configuration of Cluster.....	39
Εικόνα 3-10: Ρύθμιση των Client Connections.....	40
Εικόνα 3-11: Buildtime configuration.....	41
Εικόνα 3-12: Buildtime Database Configuration.....	41
Εικόνα 3-13: Configuration of Runtime Client.....	42
Εικόνα 3-14: Configuration of Web Client.....	42
Εικόνα 3-15: Configuration of JDK/JRE.....	43
Εικόνα 3-16: New process category.....	44
Εικόνα 3-17: Category properties.....	45
Εικόνα 3-18: New Process.....	45
Εικόνα 3-19: New process properties.....	46
Εικόνα 3-20: Σύστημα ροής εργασίας έκδοσης και εκτέλεσης ακτινολογικής εξέτασης.....	47
Εικόνα 3-21: Program Activity Properties-General Εικόνα 3-22: Program Activity Properties Start.....	48
Εικόνα 3-23: Program Activity Properties-Exit.....	48
Εικόνα 3-24: Program Activity Properties-Data Εικόνα 3-25: Program Activity Properties-Staff1.....	49
Εικόνα 3-26: Program Activity Properties-Staff2.....	49
Εικόνα 3-27: STAFF elements.....	50
Εικόνα 3-28: User U100 – Person Properties - Staff.....	51

Εικόνα 3-29: Department data structure .....	52
Εικόνα 3-30: DOCTOR_DS Data Structure	Εικόνα 3-31: patient_data Data structure. 53
Εικόνα 3-32: radissueDS Data Structure	Εικόνα 3-33: radissueDS member tree..... 53
Εικόνα 3-34: Data Structure - rereport .....	54
Εικόνα 3-35: issueradrequest – Program properties .....	55
Εικόνα 3-36: issueradrequest- program properties - data .....	56
Εικόνα 3-37: issueradrequest-Program properties- WindowsNT .....	57
Εικόνα 3-38: Export (1) .....	58
Εικόνα 3-39: Export (2) .....	59
Εικόνα 5-1: ΕΙΣΑΓΩΓΗ ΕΝΤΟΛΗΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ.....	67
Εικόνα 5-2: ΕΙΣΑΓΩΓΗ ΕΝΤΟΛΗΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ- Μήνυμα OK .....	68
Εικόνα 5-3: ΛΙΣΤΑ ΕΝΤΟΛΩΝ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ .....	69
Εικόνα 5-4: ΕΚΚΡΕΜΕΙΣ ΕΝΤΟΛΕΣ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ.....	70
Εικόνα 5-5: ΚΑΘΟΡΙΣΜΟΣ ΡΑΝΤΕΒΟΥ ΕΞΕΤΑΣΗΣ - Ημερομηνία.....	71
Εικόνα 5-6: ΚΑΘΟΡΙΣΜΟΣ ΡΑΝΤΕΒΟΥ ΕΞΕΤΑΣΗΣ - Ώρα.....	71
Εικόνα 5-7: ΚΑΘΟΡΙΣΜΟΣ ΡΑΝΤΕΒΟΥ ΕΞΕΤΑΣΗΣ - Καταχώρηση.....	72
Εικόνα 5-8: Αποστολή e-mail.....	73
Εικόνα 5-9: Παραλαβή e-mail από τον ασθενή.....	73
Εικόνα 5-10: Παραλαβή e-mail από τον θεράποντα ιατρό.....	74
Εικόνα 5-11: ΛΙΣΤΑ ΡΑΝΤΕΒΟΥ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ.....	75
Εικόνα 5-12: ΠΛΗΡΩΜΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ.....	76
Εικόνα 5-13: ΠΛΗΡΩΜΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ - Καταχώρηση .....	77
Εικόνα 5-14: ΕΚΤΕΛΕΣΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ .....	78
Εικόνα 5-15: ΕΚΤΕΛΕΣΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ-επισύναψη ακτινογραφίας.....	78
Εικόνα 5-16: ΕΚΤΕΛΕΣΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ- σχόλια γιατρού.....	79
Εικόνα 5-17: ΕΚΤΕΛΕΣΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ- καταχώρηση εγγραφής.....	79
Εικόνα 5-18: ΕΛΕΓΧΟΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ.....	80
Εικόνα 5-19: ΕΛΕΓΧΟΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ – Άντληση Ακτινογραφίας.....	81
Εικόνα 5-20: ΕΛΕΓΧΟΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ – Κλείσιμο εξέτασης .....	81
Εικόνα 5-21: Διάγραμμα ροής εκτέλεσης της διαδικασίας .....	84

## ΛΙΣΤΑ ΠΙΝΑΚΩΝ

Πίνακας 0-1 Απαιτήσεις ασφάλειας για την διαδικασία της εικόνας 0-1. ....	9
Πίνακας 1-1 : Input and Output of Issue_Rad_Request .....	15
Πίνακας 1-2: Input and Output of Lookup_Rad_Request .....	16
Πίνακας 1-3: Input and Output of Check_pending_requests .....	16
Πίνακας 1-4: Input and Output of Schedule_Rad_Request .....	17
Πίνακας 1-5: Input and Output of Notify_patient.....	17
Πίνακας 1-6: Input and Output of Search_Appointments.....	18
Πίνακας 1-7: Input and Output of Perform_Rad_Request .....	18
Πίνακας 1-8: Input and Output of Charge_Patient_Account.....	19
Πίνακας 1-9: Input and Output of Interpret_Rad_Results .....	19
Πίνακας 2-1: Περιγραφή πεδίων του πίνακα ΙΔΡΥΜΑ .....	25
Πίνακας 2-2: Περιγραφή πεδίων του πίνακα ΤΜΗΜΑΤΑ.....	26
Πίνακας 2-3: Περιγραφή πεδίων του πίνακα ΓΙΑΤΡΟΣ.....	26
Πίνακας 2-4: Περιγραφή πεδίων του πίνακα ΕΙΔΙΚΟΤΗΤΑ_ΓΙΑΤΡΟΥ.....	26
Πίνακας 2-5: Περιγραφή πεδίων του πίνακα ΑΣΘΕΝΗΣ .....	27
Πίνακας 2-6: Περιγραφή πεδίων του πίνακα ΑΣΘΕΝΕΙΣ_ΓΙΑΤΡΟΥ.....	27
Πίνακας 2-7: Περιγραφή πεδίων του πίνακα ΛΟΓΑΡΙΑΣΜΟΣ_ΑΣΘΕΝΟΥΣ.....	27
Πίνακας 2-8: Περιγραφή πεδίων του πίνακα ΕΝΤΟΛΕΣ_ΑΚΤΙΝΟΛΟΓΙΚΩΝ_ΕΞΕΤΑΣΕΩΝ.....	27
Πίνακας 2-9: Περιγραφή πεδίων του πίνακα ΑΚΤΙΝΟΛΟΓΙΚΕΣ_ΑΝΑΦΟΡΕΣ.....	27
Πίνακας 2-10: Περιγραφή πεδίων του πίνακα ΙΑΤΡΙΚΟΣ_ΦΑΚΕΛΟΣ_ΑΣΘΕΝΟΥΣ.....	28
Πίνακας 2-11: Περιγραφή πεδίων του πίνακα ΧΡΗΣΤΕΣ_ΣΥΣΤΗΜΑΤΟΣ.....	30
Πίνακας 2-12: Περιγραφή πεδίων του πίνακα ΡΟΛΟΙ_ΣΥΣΤΗΜΑΤΟΣ .....	30

Πίνακας 2-13: Περιγραφή πεδίων του πίνακα ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ_ΣΥΣΤΗΜΑΤΟΣ .....	30
Πίνακας 2-14: Περιγραφή πεδίων του πίνακα ΕΙΔΗ_ΠΡΟΣΒΑΣΕΩΝ.....	30
Πίνακας 2-15: Περιγραφή πεδίων του πίνακα ΧΡΗΣΤΕΣ_ΡΟΛΟΙ .....	31
Πίνακας 2-16: Περιγραφή πεδίων του πίνακα ΡΟΛΟΙ_ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ_ΠΡΟΣΒΑΣΕΙΣ	31

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

---

## ΠΕΡΙΛΗΨΗ

Οι ολοένα νέες εργασιακές ευκαιρίες και στρατηγικές, καθιστούν την ολοκλήρωση των επιχειρησιακών εφαρμογών κρίσιμο επίτευγμα για μια ανταγωνιστική επιχείρηση.

Η διαδικασία υποβολής ακτινολογικών εντολών, αλλά και η εκτέλεσή τους στην συνέχεια σε μια υγειονομική περιφέρεια, θα πρέπει να υποστηρίζεται από ένα κατάλληλο πληροφοριακό σύστημα, το οποίο θα είναι ευέλικτο, εύχρηστο, ικανό να τηρεί στο ακέραιο την όλη διαδικασία, αλλά και τους απαραίτητους κανόνες λειτουργίας ενός τέτοιου συστήματος.

Ένα τέτοιο σύστημα θα παρουσιαστεί στην παρούσα εργασία, το οποίο αποσκοπεί να ικανοποιεί τόσο την λειτουργική δομή του συστήματος όσο και τους κανόνες ασφαλείας που διέπουν τέτοια συστήματα υγείας.

New business opportunities and new strategies make the integration of business applications critical for a competitive business.

The procedure of issuing and executing radiological examinations in the boundaries of a health region, should be supported by a suitable information system, that would be flexible, easy to use and capable of delivering the whole procedure and operation rules of such a system.

A presentation of such a system will take place in this exam, which is aiming to satisfy the needs of the operational structure and the security issues that rule such health systems.

---

## ΕΙΣΑΓΩΓΗ

Μια Υγειονομική περιφέρεια περιλαμβάνει νοσοκομεία και κέντρα υγείας, ενώ ένα από τα νοσοκομεία αποτελεί το περιφερειακό γενικό νοσοκομείο της περιφέρειας. Στο πλαίσιο της Υγειονομικής Περιφέρειας, είναι δυνατή η έκδοση ιατρικών εντολών για εκτέλεση ακτινολογικών πράξεων από τους θεράποντες γιατρούς των ασθενών κατά τους ακόλουθους τρόπους:

- Από ένα κλινικό τμήμα του ΠΓΝ (συμπεριλαμβανομένων των εξωτερικών ιατρείων και του τμήματος επειγόντων περιστατικών) προς το ακτινολογικό τμήμα του ΠΓΝ
- Από ένα κλινικό τμήμα ενός νοσοκομείου (συμπεριλαμβανομένων των εξωτερικών ιατρείων και του τμήματος επειγόντων περιστατικών) προς το ακτινολογικό τμήμα του ίδιου νοσοκομείου.
- Από ένα κλινικό τμήμα ενός νοσοκομείου (συμπεριλαμβανομένων των εξωτερικών ιατρείων και του τμήματος επειγόντων περιστατικών) προς το ακτινολογικό τμήμα του ΠΓΝ
- Από ένα ΚΥ προς το ακτινολογικό τμήμα του ΠΓΝ

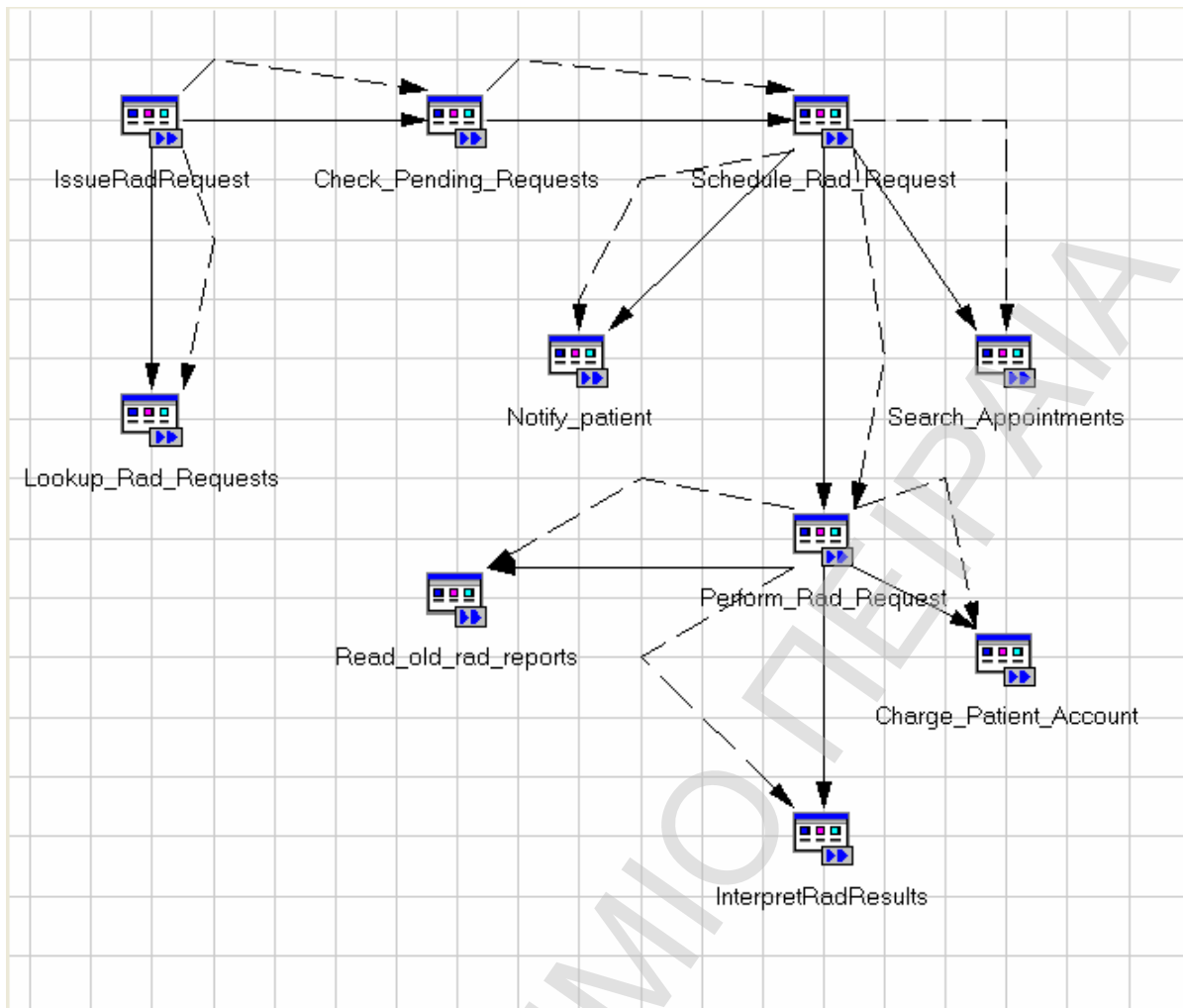
Μόλις παραληφθεί μια εντολή από ένα ακτινολογικό τμήμα, προγραμματίζεται (ανατίθεται σε ακτινολόγο και ορίζεται ημέρα και ώρα εκτέλεσης) και ενημερώνεται ο εντέλλων θεράπων γιατρός και/ή ο ασθενής (εφόσον είναι δυνατό). Όταν εκτελεστεί μια ακτινολογική πράξη, ο υπεύθυνος ακτινολόγος γιατρός συγγράφει και αποστέλλει στον θεράποντα γιατρό του ασθενούς μια ακτινολογική αναφορά που περιλαμβάνει τις εικόνες και την ακτινολογική γνωμάτευση. Η εντολή και η ακτινολογική αναφορά αποτελούν μέρη του ιατρικού φακέλου του ασθενούς.

Η υλοποίηση της παραπάνω διαδικασίας έχει πραγματοποιηθεί με την βοήθεια της τεχνολογίας workflow. Παλαιότερα πληροφοριακά συστήματα, συγχώνευαν την ροή των δραστηριοτήτων μιας διαδικασίας μέσα στα προγράμματα, γράφοντας επιπλέον κώδικα ώστε να εξασφαλίζεται η ορθή σειρά εκτέλεσης των βημάτων μιας διαδικασίας. Αυτό καθιστούσε επίπονη την κάθε αλλαγή που θα χρειαζόταν να γίνει στην τυχόν αλλαγή της επιχειρησιακής ροής, αλλά και την κατανόηση της διαδικασίας πολύ δύσκολη, μιας και θα έπρεπε να διαβάσει κανείς πολλές γραμμές κώδικα ώστε να διαπιστώσει ποια είναι η διαδικασία που ακολουθείται. Ακόμη χειρότερα, εάν αυτό δεν εξασφαλιζόταν πλήρως από το πρόγραμμα, να γίνονται λάθη στην εκτέλεση της διαδικασίας από τους χρήστες του συστήματος και να παραβιάζονται ενδεχομένως σημαντικοί κανόνες της όλης διαδικασίας.

Με την χρήση όμως σήμερα της τεχνολογίας workflow, ο προγραμματιστής με απλά βήματα και ακόμα πιο απλούστερο κώδικα, μπορεί να εξασφαλίσει την ορθή και αποτελεσματική λειτουργία του συστήματος, βοηθώντας παράλληλα τον χρήστη να εκτελεί την εργασία του χωρίς να πρέπει να γνωρίζει απαραίτητα την όλη διαδικασία. Το γραφικό δε περιβάλλον απεικόνισης των δραστηριοτήτων μιας διαδικασίας, διευκολύνει την κατανόηση της ροής της αλλά και την εύκολη αλλαγή της διαδικασίας, εάν αυτό ζητηθεί.

Η περιγραφείσα διαδικασία φαίνεται στην [εικόνα 0-1](#).





**Εικόνα 0-1** Διαδικασία υποβολής και εκτέλεσης ακτινολογικών εντολών σε υγειονομική περιφέρεια

Μερικές από τις απαιτήσεις ασφάλειας του συστήματος φαίνονται στον Πίνακα 0-1.

**Πίνακας 0-1** Απαιτήσεις ασφάλειας για την διαδικασία της εικόνας 0-1.

1.	Οι θεράποντες γιατροί μπορούν να εκδίδουν ακτινολογικές εντολές μόνο για τους ασθενείς τους (write, read)
2.	Οι θεράποντες γιατροί μπορούν να διαβάζουν στοιχεία που έχουν καταχωρηθεί στους ιατρικούς φακέλους των ασθενών τους και μόνο (write, read)
3.	Οι ακτινολόγοι γιατροί μπορούν να εκδίδουν ακτινολογικές αναφορές μόνο για τους ασθενείς στους οποίους εκτέλεσαν κάποια ακτινολογική πράξη (write, edit, send).
4.	Οι ακτινολόγοι γιατροί μπορούν να διαβάζουν ορισμένα από τα στοιχεία που έχουν καταχωρηθεί στους ιατρικούς φακέλους των ασθενών στους οποίους εκτέλεσαν ή θα εκτελέσουν κάποια ακτινολογική πράξη τους (read).
5.	Οι ακτινολόγοι γιατροί μπορούν να διαβάζουν παλαιές ακτινολογικές αναφορές μόνο για τους ασθενείς στους οποίους εκτέλεσαν κάποια ακτινολογική πράξη (read).

Επομένως, στα πλαίσια της εκπόνησης της παρούσας εργασίας, θα αναπτυχθεί και κατασκευαστεί ένα πληροφοριακό σύστημα, που θα ικανοποιεί την προαναφερθείσα διαδικασία, χρησιμοποιώντας την σημερινή τεχνολογία workflow.

# 1<sup>ο</sup> ΚΕΦΑΛΑΙΟ

ΣΥΣΤΗΜΑ ΡΟΗΣ ΕΡΓΑΣΙΑΣ

WORKFLOW SYSTEM

---

# 1 ΣΥΣΤΗΜΑ ΡΟΗΣ ΕΡΓΑΣΙΑΣ – WORKFLOW SYSTEM

## 1.1 Εισαγωγή

Στο κεφάλαιο που ακολουθεί, θα περιγραφεί πιο αναλυτικά η διαδικασία που υποστηρίζει το σύστημα ροής εργασιών: «Έκδοση - παρακολούθηση και εκτέλεση μιας ακτινολογικής εξέτασης σε μια υγειονομική περιφέρεια». Θα γίνει αναφορά στις δραστηριότητες που θα εκτελούνται στο σύστημα με την σειρά που εκτελούνται.

## 1.2 Οργανωτική Δομή Υγειονομικής Περιφέρειας

Αρχικά για να κατανοήσουμε την δομή μιας υγειονομικής περιφέρειας και επομένως την ροή που θα ακολουθεί κάθε πληροφορία του συστήματος, αναφέρουμε επιγραμματικά ότι περιλαμβάνει :

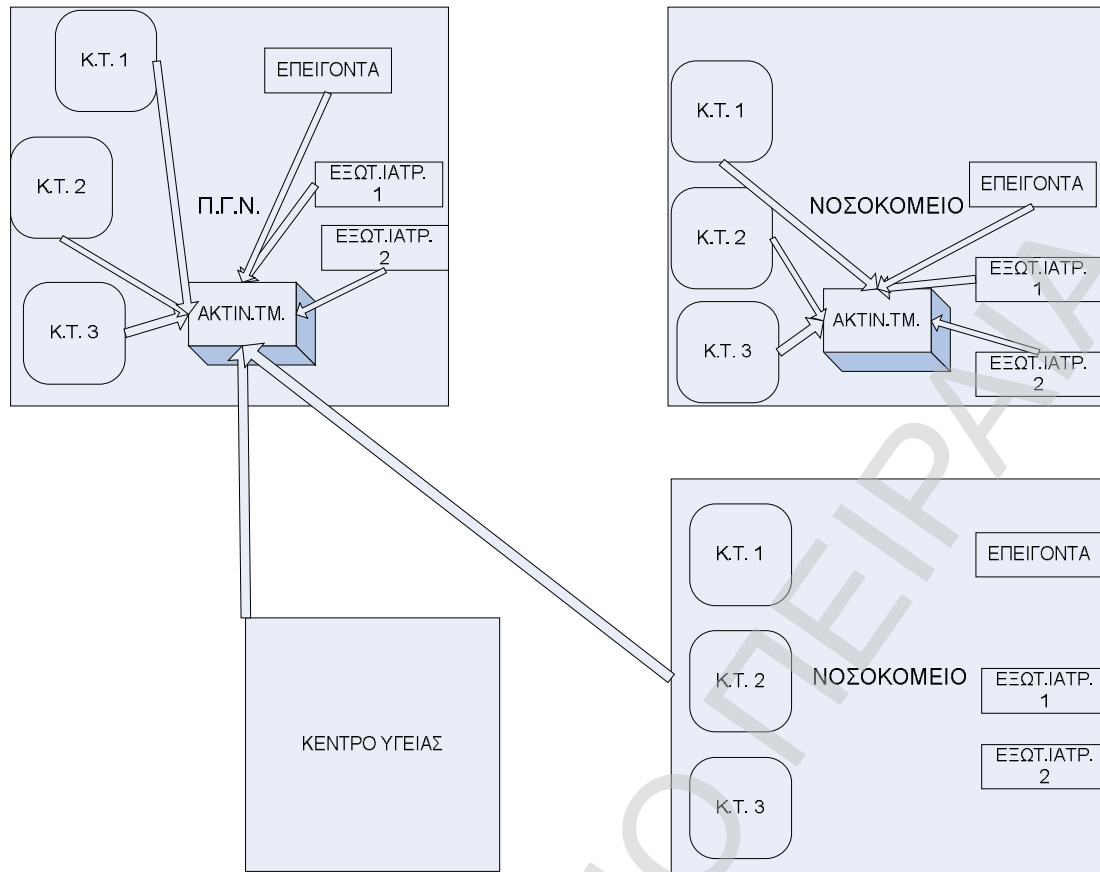
- 1) ΝΟΣΟΚΟΜΕΙΑ
- 2) ΚΕΝΤΡΑ ΥΓΕΙΑΣ
- 3) 1 ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ

Στην συνέχεια κάθε ΝΟΣΟΚΟΜΕΙΟ περιέχει:

- 1) πολλά διαφορετικά κλινικά τμήματα
- 2) πολλά εξωτερικά ιατρεία
- 3) τμήμα επειγόντων περιστατικών
- 4) 0 ή 1 ακτινολογικό τμήμα

Τα Κέντρα Υγείας δεν περιλαμβάνουν ακτινολογικό τμήμα.

Το Π.Γ.Ν. διαθέτει πάντα ακτινολογικό τμήμα, το οποίο και δέχεται περιστατικά των Κέντρων Υγείας ή των Νοσοκομείων που δεν διαθέτουν ακτινολογικό τμήμα.



**Εικόνα 1-1** Διάγραμμα υγειονομική περιφέρειας

Στην εικόνα 1-1, διαφάνεται η ανάθεση του ακτινολογικού τμήματος σε μια ακτινολογική εξέταση, από όπου και αν αυτή ξεκινάει ως απαίτηση. Επομένως, «ιδρύματα» τα οποία δεν διαθέτουν τα ίδια ακτινολογικό τμήμα, στέλνουν τους ασθενείς τους στο Π.Γ.Ν., ενώ «ιδρύματα» που διαθέτουν εξυπηρετούν τους ασθενείς τους στο ίδιο το νοσοκομείο.

Η βάση δεδομένων του συστήματος που πρόκειται να αναπτυχθεί, καθώς και το σύστημα ροής εργασιών, θα βρίσκεται κεντρικά στο Π.Γ.Ν., όπου όλα τα υπόλοιπα τμήματα, νοσοκομεία και κέντρα υγείας θα έχουν πρόσβαση. Επομένως, κάθε «τμήμα» διασυνδέεται στην κεντρική βάση δεδομένων του Π.Γ.Ν. και διαβάζει στοιχεία π.χ. του Ιατρικού φακέλου του ασθενούς σύμφωνα με τα δικά του δικαιώματα πρόσβασης στο σύστημα.

Χρήστες του συστήματος είναι όλοι οι θεράποντες ιατροί διαφόρων ειδικοτήτων, ακτινολόγοι, η γραμματεία του ακτινολογικού τμήματος και φυσικά κάποιος administrator του συστήματος για το setup του συστήματος.

Επομένως οι ρόλοι που ορίζονται είναι οι εξής:

- 1) Administrator. Είναι αυτός που διαχειρίζεται το σύστημα και έχει πρόσβαση στο να δημιουργεί νέους χρήστες στο σύστημα, να δίνει προσβάσεις χρηστών σε ρόλους του συστήματος, να καταργεί και να αλλάζει τους χρήστες του συστήματος κ.τ.λ.
- 2) Θεράπων Ιατρός.
- 3) Ακτινολόγος
- 4) Γραμματεία ακτινολογικού

### 1.3 Διαδικασία – Δραστηριότητες – Ροή Συστήματος

Αναλύοντας την διαδικασία της **Εικόνας 0-1** (ΕΙΣΑΓΩΓΗ), θα περιγράψουμε μια-μια τις δραστηριότητες που εμφανίζονται.

Ο ασθενής αρχικά επισκέπτεται κάποιο νοσοκομείο ή κέντρο υγείας της περιφέρειας, είτε το Περιφερειακό Γενικό Νοσοκομείο και ανάλογα με την υπόθεση που τον απασχολεί απευθύνεται σε γιατρό, είτε κλινικού τμήματος είτε επειγόντων περιστατικών είτε εξωτερικών ιατρείων. Αυτός είτε ήταν ξανά ασθενής του είτε είναι νέος ασθενής για αυτόν τον γιατρό. Ο γιατρός αυτός θα αποτελεί τον θεράπων ιατρό του ασθενούς. Επειδή το σύστημα ζητάει ο θεράπων ιατρός να εκδίδει ακτινολογική εξέταση μόνο για τους ασθενείς του, η διαδικασία που γίνεται ασθενής του είναι εκτός συστήματος και υλοποίησης.

Η πρώτη δραστηριότητα του συστήματος είναι η εισαγωγή του κάθε χρήστη στο σύστημα (Login). Με αυτή την διαδικασία θα γνωρίζουν οι επόμενες δραστηριότητες ποιος είναι ο χρήστης και τι δικαιώματα πρόσβασης έχει. Η δραστηριότητα αυτή όμως δεν φαίνεται στο σχήμα της διαδικασίας που περιγράφεται διότι γίνεται εσωτερικά από το πρόγραμμα “MQSeries Workflow Client”. Οι χρήστες του συστήματος έχουν καθοριστεί στην βάση δεδομένων του workflow, όπως θα δούμε και παρακάτω αναλυτικότερα, ώστε να μπορούν να χρησιμοποιηθούν από το σύστημα που αναπτύσσουμε. Οι ρόλοι του κάθε χρήστη καθορίζονται στην βάση δεδομένων του workflow, παράλληλα όμως και επιπρόσθετα όπως θα δούμε παρακάτω, ορίζονται και στην βάση δεδομένων του συστήματος ασφαλείας του συστήματος.

Εν συνεχεία ξεκινάνε οι δραστηριότητες του συστήματος έκδοσης ακτινολογικής εξέτασης, οι οποίες ελέγχονται μέσω του κεντρικού συστήματος workflow.

Σε κάθε δραστηριότητα (**program activity**), θα γίνεται αναφορά στα εξής:

- I. Λειτουργία της δραστηριότητας και τι αποσκοπεί να ικανοποιήσει από την όλη διαδικασία
- II. **Actors** : Ρόλοι που θα μπορούν να εκτελέσουν την συγκεκριμένη διαδικασία
- III. **Input** : Ποια πληροφορία θα χρειάζεται κάθε διαδικασία προκειμένου να λειτουργήσει.
- IV. **Output** : Ποια θα είναι η πληροφορία που θα επιστρέφει το σύστημα μετά την εκτέλεση της διαδικασίας.
- V. **Constraints** : Ποιοι θα είναι οι περιορισμοί της διαδικασίας όπου δεν τις επιτρέπουν να λειτουργήσει και να προχωρήσει προς το αποτέλεσμα.
- VI. **Exceptions** : Τα ενδεχόμενα λάθη, προειδοποιητικά μηνύματα τα οποία θα πρέπει να επιστρέφονται προς τον χρήστη σε περιπτώσεις καταπάτησης των περιορισμών.

Οι δραστηριότητες που αποτελούν την διαδικασία είναι οι εξής:

- **Issue Rad Request** (Εισαγωγή Εντολής Ακτινολογικής Εξέτασης)
- **Lookup Rad Request** (διάβασμα – έλεγχος εντολών ακτινολογικών εξετάσεων)
- **Check pending requests** (Αναζήτηση εκκρεμών εντολών- εντολές που δεν έχουν καθοριστεί ραντεβού)
- **Schedule Rad Request** (Καθορισμός ραντεβού και ακτινολόγου που θα εκτελέσει την εξέταση).
- **Notify patient** (Ειδοποίηση του ασθενούς και του θεράποντος ιατρού με το ραντεβού του ασθενή μέσω e-mails)
- **Search Appointments** (έλεγχος των ραντεβού που έχει ένας ακτινολόγος γιατρός).

- **Perform Rad Request** (Εκτέλεση της ακτινογραφίας)
- **Read\_Old\_Rad\_Reports** (Επισκόπηση παλιότερων ακτινολογικών αναφορών)
- **Charge Patient Account** (Πληρωμή)
- **Interpret Rad Results** (Διαδικασία εισαγωγής τελικής γνωμάτευσης του θεράποντος ιατρού στην εγγραφή της ακτινολογικής αναφοράς της ακτινολογικής εξέτασης)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

### 1.3.1 Issue\_Rad\_request (Εισαγωγή Εντολής Ακτινολογικής Εξέτασης)

Την δεδομένη στιγμή ξεκινάει η διαδικασία όπου θα χρειαστεί ο ασθενής να κάνει ακτινολογική εξέταση. Ο γιατρός λοιπόν (θεράπων ιατρός), εκδίδει εντολή εξέτασης του ασθενούς του είτε προς το ακτινολογικό τμήμα του νοσοκομείου που βρίσκονται εφόσον έχει τέτοιο τμήμα, είτε στο ακτινολογικό τμήμα του Π.Γ.Ν.

<b>Actor:</b>	Θεράπων ιατρός του ασθενούς
<b>Input:</b>	<ul style="list-style-type: none"><li>ο Στοιχεία ασθενούς (κωδικός ασθενούς).</li><li>ο Ακτινολογικό τμήμα που θα σταλεί και θα εκτελεστεί η ακτινογραφία. (Εύρεση μέσω εσωτερικής αυτόματης διαδικασίας που περιγράφεται παρακάτω)</li><li>ο Κωδικός θεράποντος ιατρού. Παρέχεται προκειμένου να γνωρίζει ο ακτινολόγος ποιον γιατρό θα ειδοποιήσει με το ραντεβού και την αναφορά.</li></ul>
<b>Output:</b>	Μια εγγραφή εντολής ακτινολογικής εξέτασης για τον ασθενή. Μια νέα εγγραφή στον ιατρικό φάκελο του ασθενούς με τον κωδικό της ακτινολογικής εξέτασης.
<b>Constraints:</b>	Ένας γιατρός δεν μπορεί να εκδώσει εντολή ακτινολογικής εξέτασης για ασθενή που δεν είναι δικός του.
<b>Exceptions</b>	Αν ο χρήστης εισαγωγής δεν έχει πρόσβαση στο συγκεκριμένο πρόγραμμα, να μην του επιτρέπεται η πρόσβαση με μήνυμα λάθους. Αν ο ασθενής δεν είναι ασθενής του θεράποντος ιατρού, να μην επιτρέπεται η καταχώρηση της εντολής και η διαδικασία να τερματίζεται με μήνυμα λάθους προς τον χρήστη.

Πίνακας 1-1 : Input and Output of Issue\_Rad\_Request

Κατά την εκτέλεση της διαδικασίας, το σύστημα αυτόματα θα εκτελεί τις παρακάτω ενέργειες:

#### 1<sup>α</sup>) Εσωτερική διαδικασία εύρεσης του τμήματος στο οποίο εργάζεται ο θεράπων ιατρός.

Γίνεται εύρεση από την βάση δεδομένων, βάσει του κωδικού του θεράποντος ιατρού, το τμήμα στο οποίο αυτός εργάζεται.

#### 1<sup>β</sup>) Εσωτερική διαδικασία εύρεσης ακτινολογικού τμήματος

Η function αυτή, η οποία έχει αναπτυχθεί σε Oracle plsql στην βάση δεδομένων, παίρνει ως όρισμα το τμήμα στο οποίο εργάζεται ο θεράπων ιατρός, από την εσωτερική διαδικασία 1<sup>α</sup>). Εάν αυτό το τμήμα ανήκει σε νοσοκομείο που διαθέτει ακτινολογικό τμήμα, τότε επιστρέφεται το ακτινολογικό τμήμα αυτού του νοσοκομείου. Εάν όχι, τότε αποστέλλεται στο ακτινολογικό τμήμα του Π.Γ.Ν. (plsql function find\_aktinologiko. Παράρτημα 2).

### 1.3.2 Lookup Rad Request (Διάβασμα – Έλεγχος Εντολών Ακτινολογικών Εξετάσεων)

Επόμενη διαδικασία είναι αυτή όπου ο χρήστης έχει την ευχέρεια να βλέπει όλα τα στοιχεία των ακτινολογικών εντολών και να βλέπει σε τι κατάσταση είναι. Χρήστης της διαδικασίας είναι ο θεράπων ιατρός και μπορεί να βλέπει μόνο τις εντολές που έχουν ως θεράπων ιατρό τον ίδιο. Επίσης έχει την δυνατότητα να βλέπει αν έχει καθορισθεί ραντεβού για την εξέταση αυτή και σε ποιόν ακτινολόγο και νοσοκομείο θα γίνει η εξέταση.

<b>Actors:</b>	A) Θεράπων ιατρός του ασθενούς
<b>Input:</b>	Ο κωδικός του θεράποντος ιατρού.
<b>Output:</b>	Επιστρέφονται όλες οι εντολές για ακτινολογική εξέταση που έχει εκδώσει ο συγκεκριμένος ιατρός.
<b>Constraints:</b>	Επιλέγονται αυτόματα μόνο εντολές του θεράποντος ιατρού. (USER = DOCT_KOD)
<b>Exceptions</b>	Σε περίπτωση εκτέλεσης του προγράμματος από μη εξουσιοδοτημένο χρήστη, μήνυμα λάθους πρόσβασης στοιχείων.

Πίνακας 1-2: Input and Output of Lookup\_Rad\_Request

### 1.3.3 Check pending requests (Αναζήτηση εκκρεμών εντολών- εντολές που δεν έχουν καθοριστεί ραντεβού)

Η προηγούμενη εντολή ακτινολογικής εξέτασης του θεράποντος ιατρού, προωθείται προς την γραμματεία του νοσοκομείου που αποστάληκε και είναι υποχρεωμένη να ορίσει άμεσα το ραντεβού του ασθενούς.

Η γραμματεία, με το που εισαχθεί στο πρόγραμμα αυτό βλέπει τα στοιχεία του ασθενούς που έχει παραπεμφθεί σε ακτινολογική εξέταση και θα πρέπει αυτή στην συνέχεια να συνεχίσει στην επόμενη δραστηριότητα του καθορισμού ραντεβού που περιγράφεται παρακάτω.

<b>Actors:</b>	A) Γραμματεία του Ακτινολογικού Τμήματος
<b>Input:</b>	A) Κωδικός Ασθενή, Κωδικός Ακτινολογικού τμήματος, Θεράπων Ιατρός
<b>Output:</b>	A) Κωδικός Ασθενή, Κωδικός Ακτινολογικού τμήματος, Θεράπων Ιατρός
<b>Constraints:</b>	Η γραμματεία βλέπει μόνο τις εγγραφές που έχουν καθοριστεί να γίνουν στο τμήμα της.
<b>Exceptions</b>	Σε περίπτωση εκτέλεσης του προγράμματος από μη εξουσιοδοτημένο χρήστη, μήνυμα λάθους πρόσβασης στοιχείων.

Πίνακας 1-3: Input and Output of Check\_pending\_requests

### 1.3.4 Schedule Rad Request (Καθορισμός ραντεβού και ακτινολόγου που θα εκτελέσει την εξέταση)

Η γραμματεία του νοσοκομείου που αποστάληκε η εντολή της εξέτασης, είναι εδώ υπεύθυνη να καθορίσει το ραντεβού του ασθενούς και τον ακτινολόγο ο οποίος θα τον εξετάσει. Έχει την δυνατότητα να ζητήσει αυτόματα από το σύστημα να καθορίσει το ραντεβού καθώς και τον ακτινολόγο γιατρό, αλλά παράλληλα έχει και την δυνατότητα να προβεί σε αλλαγές σύμφωνα με τις δικές της συνιστώσες.

<b>Actors:</b>	A) Γραμματεία Ακτινολογικού Τμήματος
<b>Input:</b>	A) Κωδικός Ασθενή, Κωδικός Ακτινολογικού τμήματος, Θεράπων Ιατρός
<b>Output:</b>	A) Ημερομηνία ραντεβού B) Ώρα ραντεβού



	Γ) Κωδικός Ακτινολόγου Ιατρού
<b>Constraints:</b>	Η γραμματεία βλέπει μόνο τις εγγραφές που έχουν καθοριστεί να γίνουν στο τμήμα της.
<b>Exceptions</b>	Σε περίπτωση εκτέλεσης του προγράμματος από μη εξουσιοδοτημένο χρήστη, μήνυμα λάθους πρόσβασης στοιχείων.

**Πίνακας 1-4:** Input and Output of Schedule\_Rad\_Request

Στην δραστηριότητα αυτή, ο χρήστης επιλέγοντας την εύρεση του προτεινόμενου ραντεβού από το σύστημα, εκτελείται εσωτερική διαδικασία αυτόματης εύρεσης του ακτινολόγου Ιατρού που θα εκτελέσει την εξέταση.

Η plsql function hospital.set\_appointment\_doctor (παράρτημα 2) της βάσης δεδομένων, βρίσκει τον ακτινολόγο του συγκεκριμένου τμήματος ο οποίος έχει μέχρι στιγμής τα λιγότερα ραντεβού, προκειμένου να μην απευθύνονται όλα τα ραντεβού στον ίδιο ακτινολόγο και να γίνεται δίκαιος καταμερισμός εργασίας.

Εν συνεχεία το σύστημα εκτελεί εσωτερική αυτόματη διαδικασία εύρεσης της μέρας του ραντεβού. Η plsql function hospital.set\_appointment\_date (παράρτημα 2) της βάσης δεδομένων, βρίσκει για τον συγκεκριμένο ακτινολόγο ποια μέρα μπορεί να αναλάβει το επόμενο ραντεβού του. Έχει γίνει παραδοχή ότι ένας γιατρός μπορεί να φέρει εις πέρας μέχρι και 10 ραντεβού την ημέρα. Ελέγχει λοιπόν για την σημερινή μέρα πόσα ραντεβού έχουν καθοριστεί για τον γιατρό αυτό που εκκρεμούν και αν είναι <10 τότε καθορίζεται για σήμερα, αλλιώς πάει για την επόμενη εργάσιμη μέρα (χωρίς Σάββατα και Κυριακές).

Τέλος, το σύστημα εκτελεί εσωτερική αυτόματη διαδικασία εύρεσης της ώρας του ραντεβού εκείνης της μέρας και για αυτόν τον γιατρό. Έχει γίνει παραδοχή ότι κάθε ραντεβού διαρκεί μισή ώρα ξεκινώντας από τις 8:00 π.μ. Η plsql function hospital.set\_appointment\_time (παράρτημα 2) της βάσης δεδομένων, επιστρέφει την ώρα του ραντεβού για τον συγκεκριμένο γιατρό και την συγκεκριμένη μέρα.

Η γραμματεία εάν δεν συμφωνεί με το προτεινόμενο ραντεβού έχει την δυνατότητα να προβεί σε οποιαδήποτε αλλαγή.

### 1.3.5 Notify\_patient (Ειδοποίηση ασθενούς και θεράποντος Ιατρού μέσω e-mail)

Εν συνεχεία, αφού καταχωρηθεί το ραντεβού, το σύστημα εκτελεί αυτόματη διαδικασία ενημέρωσης τόσο του ασθενούς όσο και του θεράπων Ιατρού για το καθορισμένο ραντεβού, εφόσον υπάρχουν οι απαραίτητες e-mail διευθύνσεις, μέσω ηλεκτρονικού ταχυδρομείου.

<b>Actors:</b>	- αυτόματη διαδικασία -
<b>Input:</b>	e-mail ασθενή, e-mail θεράποντος Ιατρού
<b>Output:</b>	Δημιουργία e-mail και αποστολή του στον ασθενή και στον γιατρό
<b>Constraints:</b>	-
<b>Exceptions</b>	Εάν δεν υπάρχει το e-mail του ασθενή, δεν αποστέλλεται το μήνυμα

**Πίνακας 1-5:** Input and Output of Notify\_patient

Η ενέργεια γίνεται αμέσως μετά, αυτόματα από το σύστημα με e-mail ώστε να ειδοποιηθεί τελικά ο ασθενής με το ραντεβού. Χρησιμοποιείται ο smtp server του google (gmail)

### 1.3.6 Search\_Appointments (Έλεγχος των ραντεβού ακτινολόγου Ιατρού)

Κάθε ακτινολόγος γιατρός, έχει την δυνατότητα να βλέπει όλα τα ραντεβού που έχουν καθοριστεί για αυτόν, συνολικά σε έναν πίνακα. Το σύστημα γνωρίζοντας τον κωδικό του

χρήστη με τον οποίο ο ακτινολόγος έχει εισαχθεί στο σύστημα, αντλεί από την βάση δεδομένων όλα τα ραντεβού που έχουν καθοριστεί για τον συγκεκριμένο γιατρό.

<b>Actors:</b>	Ακτινολόγος γιατρός
<b>Input:</b>	Κωδικός ακτινολόγου γιατρού
<b>Output:</b>	Πίνακας με όλες τις εξετάσεις που έχουν καθοριστεί με ραντεβού για αυτόν τον ακτινολόγο.
<b>Constraints:</b>	-
<b>Exceptions</b>	Σε περίπτωση εκτέλεσης του προγράμματος από μη εξουσιοδοτημένο χρήστη, μήνυμα λάθους πρόσβασης στοιχείων.

Πίνακας 1-6: Input and Output of Search\_Appointments

### 1.3.7 Perform Rad Request (Εκτέλεση της ακτινογραφίας)

Ο ασθενής πηγαίνει στο ακτινολογικό τμήμα για να πραγματοποιήσει την εξέταση την ημέρα και ώρα που ορίζει το ραντεβού.

Αφού τελειώσει η κάθε εξέταση από τον βοηθό ακτινολόγου (διαδικασία εκτός συστήματος), η ακτινογραφία δίνεται (είναι διαθέσιμη) στον ακτινολόγο γιατρό και αυτός με την σειρά του, συμπληρώνει τα σχόλιά του και επισυνάπτει την εικόνα της ακτινογραφίας στην εγγραφή της εξέτασης του ασθενούς προς αποθήκευσή της στην βάση δεδομένων του συστήματος. Όταν ολοκληρωθεί η εκτέλεση της εξέτασης, η εξέταση «κλείνει» και έτσι αποστέλλεται πίσω στον θεράπων ιατρό.

<b>Actors:</b>	Ακτινολόγος ιατρός
<b>Input:</b>	Κωδικός Ασθενούς Rad_request ID (κλειδί εγγραφής εντολής) Ακτινογραφία Σχόλια Ακτινολόγου
<b>Output:</b>	Κλείσιμο της εγγραφής εντολής εξέτασης με την ημερομηνία εκτέλεσης της εξέτασης και παράλληλα συμπλήρωση σχολίων του ακτινολόγου γιατρού και δημιουργία της εγγραφής ακτινολογικής αναφοράς με επισύναψη της ακτινογραφίας και ιατρικών σχολίων του.
<b>Constraints:</b>	Η εκτέλεση της εξέτασης γίνεται μόνο από τον ακτινολόγο που έχει καθοριστεί στο ραντεβού. Η ακτινογραφία είναι υποχρεωτική.
<b>Exceptions</b>	Σε περίπτωση εκτέλεσης του προγράμματος από μη εξουσιοδοτημένο χρήστη, μήνυμα λάθους πρόσβασης στοιχείων.

Πίνακας 1-7: Input and Output of Perform\_Rad\_Request

Η επισύναψη της ακτινογραφίας γίνεται με άντληση του αρχείου από το file system και η αποθήκευσή της θα γίνει σε μορφή BLOB αρχείο στην βάση δεδομένων.

### 1.3.8 Read\_Old\_Rad\_Reports (Επισκόπηση παλιότερων ακτινολογικών αναφορών)

Ο κάθε ακτινολόγος γιατρός μπορεί να δει παλαιότερες εξετάσεις που έχει εκτελέσει ο ίδιος σε ασθενείς του και μόνο. Επιλέγοντας έναν από τους ασθενείς του, του δίνεται η δυνατότητα ελέγχου όλων των εξετάσεων που είχε πραγματοποιήσει αυτός ο ασθενής και είχαν ως ακτινολόγο γιατρό τον ίδιο. Έχει την δυνατότητα να δει τις ιατρικές γνωματεύσεις καθώς και την ακτινογραφία της κάθε εξέτασης.

<b>Actors:</b>	Ακτινολόγος ιατρός
<b>Input:</b>	Κωδικός χρήστη και Ιατρού Κωδικός Ασθενούς Rep_ ID (κλειδί εγγραφής ακτινολογικής αναφοράς)
<b>Output:</b>	-

<b>Constraints:</b>	Η εκτέλεση της εξέτασης γίνεται μόνο από ακτινολόγο και μπορεί να επιλέξει ασθενή δικό του και μόνο.
<b>Exceptions</b>	Σε περίπτωση εκτέλεσης του προγράμματος από μη εξουσιοδοτημένο χρήστη, μήνυμα λάθους πρόσβασης στοιχείων.

Πίνακας 1-8: Input and Output of Read\_Old\_Rad\_Reports

### 1.3.9 Charge Patient Account (Πληρωμή)

Εφόσον ολοκληρωθεί η προηγούμενη δραστηριότητα εκτέλεσης της ακτινογραφίας, η γραμματεία του ακτινολογικού τμήματος ζητάει πληρωμή της εξέτασης από τον ασθενή. Εφόσον ο ασθενής πληρώσει, ο ταμίας ενημερώνει τον λογαριασμό του ασθενούς (χρέωση λογαριασμού).

<b>Actors:</b>	Γραμματεία Ακτινολογικού Τμήματος
<b>Input:</b>	Κωδικός Ασθενούς Ποσό χρέωσης
<b>Output:</b>	Χρέωση του λογαριασμού του ασθενούς με την πληρωμή της ακτινογραφίας
<b>Constraints:</b>	Μόνο η γραμματεία του ακτινολογικού που έχει καθοριστεί να γίνει η εξέταση έχει πρόσβαση στην πληρωμή.
<b>Exceptions</b>	Σε περίπτωση εκτέλεσης του προγράμματος από μη εξουσιοδοτημένο χρήστη, μήνυμα λάθους πρόσβασης στοιχείων.

Πίνακας 1-8: Input and Output of Charge\_Patient\_Account

### 1.3.10 Interpret Rad Results (Εισαγωγή τελικής γνωμάτευσης θεράποντος ιατρού)

Ο θεράπων ιατρός ενημερώνεται για την εκτέλεση της εξέτασης και τα αποτελέσματά της. Ο θεράπων ιατρός συμπληρώνει την γνωμάτευσή του αφού ελέγξει την ακτινογραφία του ασθενούς, αντλώντας την από την βάση δεδομένων.

<b>Actors:</b>	Θεράπων ιατρός
<b>Input:</b>	<ul style="list-style-type: none"> <li>• Εγγραφή ακτινολογικής αναφοράς.</li> <li>• Ιατρική γνωμάτευση του ακτινολόγου ιατρού.</li> <li>• Ακτινογραφία (φωτογραφία).</li> </ul>
<b>Output:</b>	Ενημέρωση εγγραφής ακτινολογικής αναφοράς με την ιατρική γνωμάτευση και την ημερομηνία που έκλεισε η εξέταση. Τέλος, ενημερώνεται και ο ιατρικός φάκελος του ασθενούς με την ιατρική γνωμάτευση του γιατρού και τον κωδικό της ακτινολογικής αναφοράς.
<b>Constraints:</b>	Τα 3 input πεδία είναι υποχρεωτικά Ο χρήστης που εισάγεται στο σύστημα θα πρέπει να είναι ο Θεράπων ιατρός της συγκεκριμένης εξέτασης.
<b>Exceptions</b>	Σε περίπτωση εκτέλεσης του προγράμματος από μη εξουσιοδοτημένο χρήστη, μήνυμα λάθους πρόσβασης στοιχείων.

Πίνακας 1-9: Input and Output of Interpret\_Rad\_Results

## 2<sup>ο</sup> ΚΕΦΑΛΑΙΟ

### ΣΧΕΔΙΑΣΜΟΣ - ΥΛΟΠΟΙΗΣΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

### DATABASE DESIGN AND IMPLEMENTATION

---

## 2 ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ – ΠΕΡΙΓΡΑΦΗ & ΔΙΑΓΡΑΜΜΑΤΑ

### 2.1 Εισαγωγή

Τα μοντέλα δεδομένων που περιλαμβάνονται στο κεφάλαιο αυτό αφορούν στη Βάση Δεδομένων του συστήματος υποβολής και εκτέλεσης ακτινολογικών εντολών σε υγειονομική περιφέρεια και εν συνεχεία στην Βάση Δεδομένων του συστήματος ασφάλειας του παραπάνω συστήματος, σύμφωνα με το μοντέλο RBAC (Role Base Access Control), το οποίο βασίζεται σε ρόλους χρηστών και αρμοδιότητες που δίνονται σε αυτούς τους ρόλους.

Κάθε μοντέλο δεδομένων διακρίνεται σε δυο μέρη :

- § Το πρώτο περιγράφεται σε μορφή διαγραμμάτων οντοτήτων – σχέσεων, και δείχνει τις οντότητες και τις σχέσεις μεταξύ τους, και
- § Το δεύτερο περιγράφει τους πίνακες της σχεσιακής βάσης δεδομένων που θα υλοποιηθούν. Επιπλέον περιγράφονται τα ονόματα, η ερμηνεία και οι τύποι των πεδίων που περιλαμβάνονται σε κάθε πίνακα.

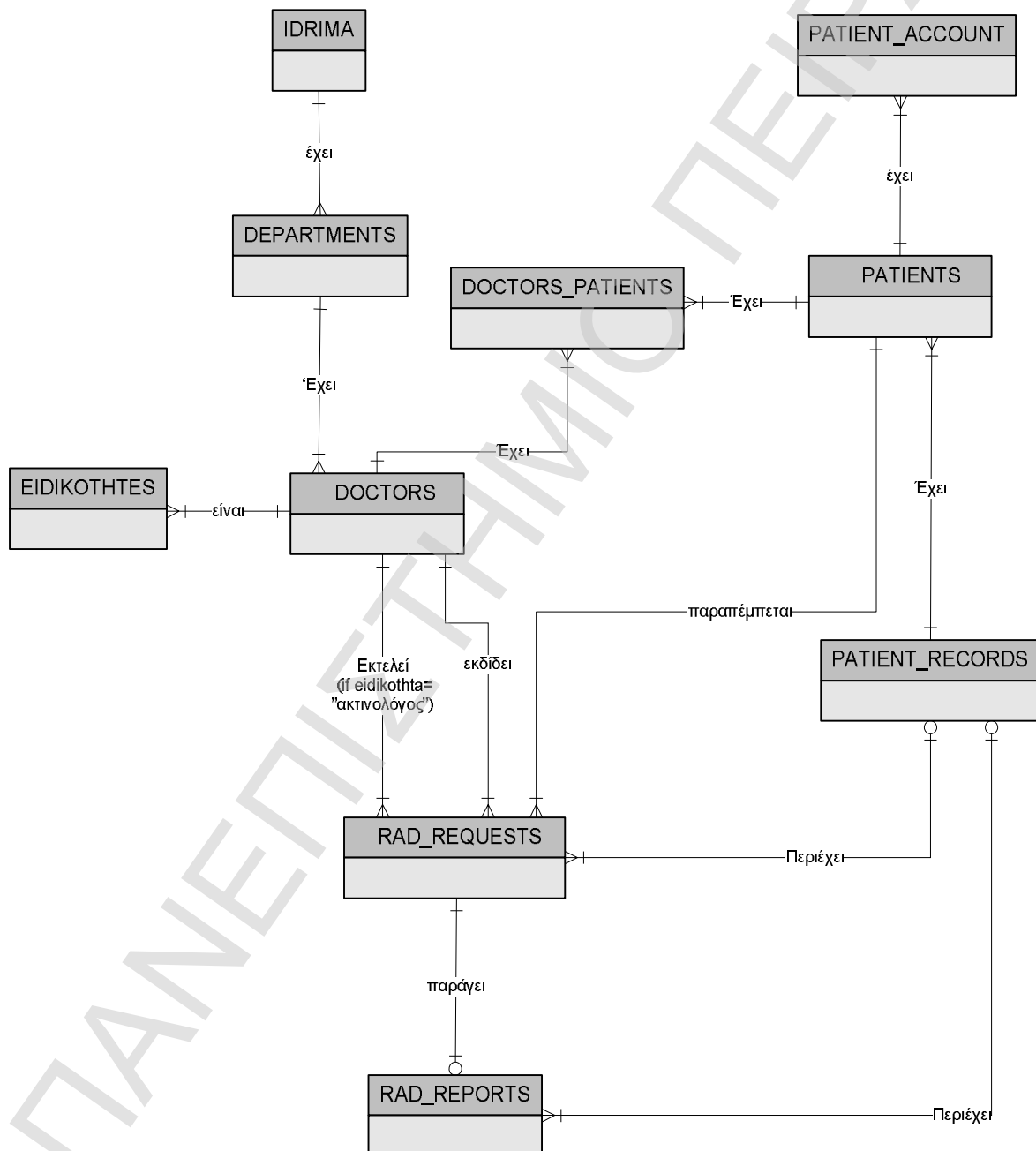
Η βάση δεδομένων που έχει επιλεγεί είναι Oracle 10g Release 2.

## 2.2 Διαγράμματα μοντέλων της Βάσης Δεδομένων

Στην ενότητα αυτή παρατίθενται τα διαγράμματα μοντέλων (Entity Relationship Diagrams) και των δύο βάσεων δεδομένων.

Σε αυτά διακρίνουμε τις οντότητες που λαμβάνουν μέρος στο κάθε σύστημα και την σχέση που έχουν μεταξύ τους. Εν συνεχεία περιγράφονται αναλυτικά οι σχέσεις αυτές μεταξύ τους.

### 2.2.1 Διάγραμμα μοντέλου βάσης δεδομένων συστήματος υποβολής και εκτέλεσης ακτινολογικής εντολής



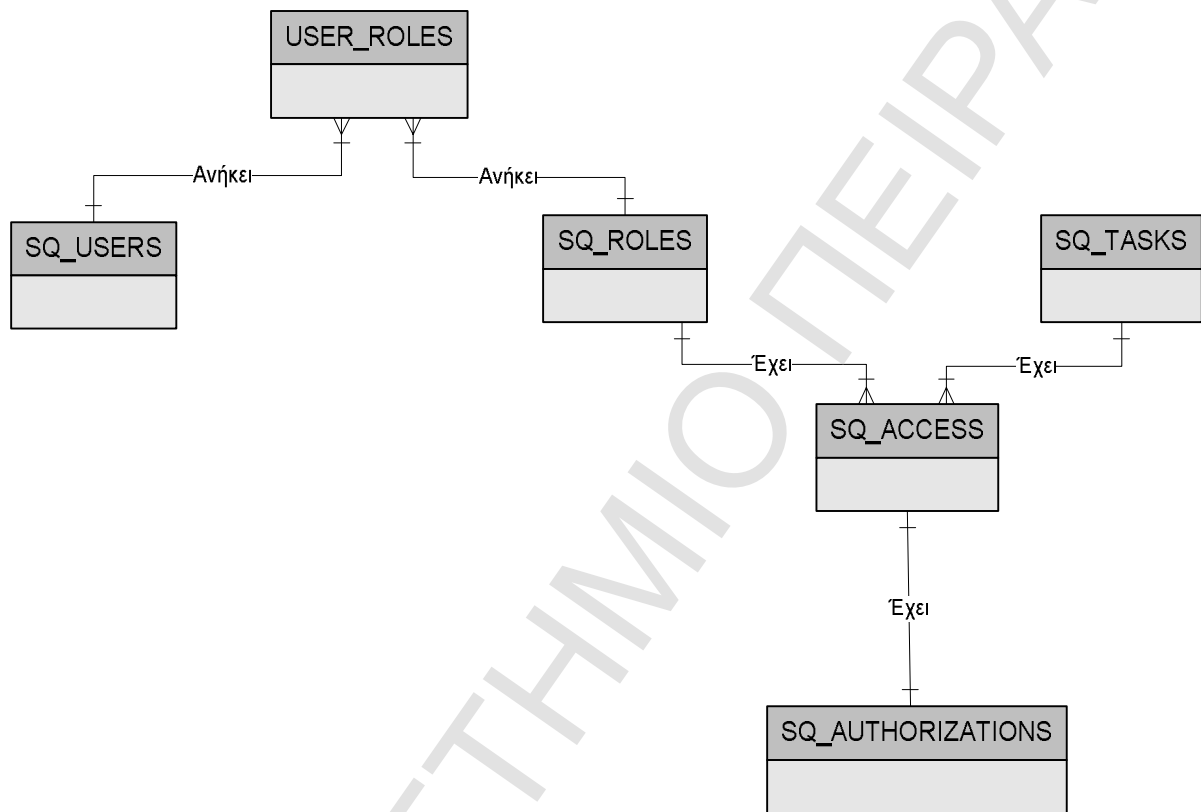
Εικόνα 2-1 : E-R Diagram (1)

Στο μοντέλο οντοτήτων – σχέσεων που βλέπουμε στην [εικόνα 2-1](#), περιγράφονται οι παρακάτω οντότητες και σχέσεις μεταξύ τους:

- 1) Ένα ίδρυμα (IDRIMA) π.χ. Νοσοκομείο, αποτελείται από πολλά τμήματα και κάθε τμήμα (DEPARTMENTS) ανήκει σε ένα μόνο ίδρυμα. Ένα από τα χαρακτηριστικά του ιδρύματος όπως θα δούμε και στο σχεσιακό μοντέλο, είναι ο τύπος του. Αυτό (idrima-type), μπορεί να πάρει τις τιμές:
  - '1' = Νοσοκομείο
  - '2' = Κέντρο Υγείας
  - '3' = Π.Γ.ΝΑντίστοιχα το χαρακτηριστικό του τμήματος (dept\_type) είναι :
  - ΚΤ = κλινικό τμήμα
  - ΑΤ = ακτινολογικό τμήμα
  - ΕΞ = Εξωτερικό ιατρείο
  - ΕΠ = Επείγοντα.
- 2) Ένας γιατρός (DOCTORS) εργάζεται σε ένα μόνο τμήμα και αντίστοιχα ένα τμήμα (DEPARTMENTS) έχει τουλάχιστον ένα γιατρό.
- 3) Κάθε γιατρός (DOCTORS) έχει μία ειδικότητα (EIDIKOTHTES) π.χ. παθολόγος, ακτινολόγος, καρδιολόγος κ.τ.λ., ενώ μια ειδικότητα (EIDIKOTHTES) έχει πολλούς γιατρούς (DOCTORS).
- 4) Κάθε ασθενής (PATIENTS) έχει πολλούς λογαριασμούς (PATIENT\_ACCOUNT) και κάθε λογαριασμός ανήκει σε έναν μόνο ασθενή (PATIENTS).
- 5) Κάθε ασθενής (PATIENTS) έχει έναν ιατρικό φάκελο (PATIENT\_RECORDS) που ξεκινάει από την πρώτη επίσκεψη σε κάποιο γιατρό και οι ιατρικοί φάκελοι (PATIENT\_RECORDS) ανήκουν σε πολλούς ασθενείς (PATIENTS).
- 6) Ένας γιατρός (DOCTORS) έχει πολλούς ασθενείς (PATIENTS) και αντίστοιχα ένας ασθενής (PATIENTS) έχει κατά καιρούς πολλούς διαφορετικούς γιατρούς (DOCTORS). Επομένως αυτή η σχέση δημιουργεί μια νέα οντότητα-πίνακα: (DOCTORS\_PATIENTS) η οποία περιέχει τα κλειδιά των δύο παραπάνω οντοτήτων.
- 7) Ένας γιατρός (DOCTORS) εκδίδει πολλές εντολές ακτινολογικών εξετάσεων (RAD\_REQUESTS), αλλά κάθε εντολή ακτινολογικής εξέτασης (RAD\_REQUESTS) έχει εκδοθεί από έναν μόνο γιατρό (DOCTORS).
- 8) Κάθε γιατρός (DOCTORS) που έχει ειδικότητα ακτινολόγου μπορεί να εκτελέσει την εντολή ακτινολογικής εξέτασης (RAD\_REQUESTS) και κάθε εντολή ακτινολογικής εξέτασης (RAD\_REQUESTS), εκτελείται από έναν μόνο ακτινολόγο γιατρό (DOCTORS).
- 9) Ένας ασθενής (PATIENT) παραπέμπεται σε πολλές ακτινολογικές εξετάσεις (RAD\_REQUESTS) και κάθε εντολή ακτινολογικής εξέτασης (RAD\_REQUESTS) εκδίδεται για έναν ασθενή (PATIENTS).
- 10) Μία εντολή ακτινολογικής εξέτασης (RAD\_REQUESTS), μπορεί να δημιουργήσει μόνο μία ακτινολογική αναφορά (RAD\_REPORT), και αντίστοιχα κάθε ακτινολογική αναφορά (RAD\_REPORT) προέρχεται ακριβώς από μία μόνο εντολή ακτινολογικής εξέτασης (RAD\_REQUESTS). Εδώ λοιπόν το κλειδί της οντότητας RAD\_REQUEST «κατεβαίνει» στην οντότητα (RAD\_REPORTS).
- 11) Ένας ιατρικός φάκελος (PATIENT\_RECORDS) περιέχει πολλές ακτινολογικές αναφορές (RAD\_REPORTS) και κάθε ακτινολογική αναφορά (RAD\_REPORTS) ανήκει σε μια εγγραφή του ιατρικού φακέλου του ασθενούς (PATIENT\_RECORDS).

- 12) Κάθε εντολή ακτινολογικής εξέτασης (RAD\_REQUESTS) ανήκει σε – δημιουργεί μια εγγραφή του ιατρικού φακέλου του ασθενούς (PATIENT\_RECORDS) και ένας ιατρικός φάκελος (PATIENT\_RECORDS) μπορεί να περιλαμβάνει πολλές εντολές ακτινολογικής εξέτασης (RAD\_REQUESTS) που έχουν γίνει κατά καιρούς στον ασθενή.

## 2.2.2 Διάγραμμα μοντέλου βάσης δεδομένων ασφαλείας συστήματος



Εικόνα 2-2 : E-R Diagram (2)

Στο μοντέλο οντοτήτων – σχέσεων που βλέπουμε στην [εικόνα 2-2](#), περιγράφονται οι παρακάτω οντότητες και σχέσεις μεταξύ τους:

- 1) Κάθε χρήστης του συστήματος (SQ\_USER) μπορεί να ανήκει σε πολλούς ρόλους (SQ\_ROLES), ενώ αντίστοιχα κάθε ρόλος (SQ\_ROLES) ανήκει σε πολλούς χρήστες. Επομένως, η σχέση αυτή «πολλά προς πολλά», «σπάει» και δημιουργείται μια νέα οντότητα-σχέση (USER\_ROLES), όπου τα κλειδιά των δύο προηγούμενων οντοτήτων κατεβαίνουν στην νέα οντότητα.
- 2) Κάθε ρόλος (SQ\_ROLES) μπορεί να εκτελεί πολλές λειτουργίες (SQ\_TASKS), ενώ αντίστοιχα κάθε δραστηριότητα (SQ\_TASK) μπορεί να ανήκει σε πολλούς ρόλους (SQ\_ROLES). Επομένως και εδώ, δημιουργείται μια νέα οντότητα – σχέση (SQ\_ACCESS), όπου τα κλειδιά των δύο προηγούμενων οντοτήτων κατεβαίνουν στην νέα οντότητα.



- 3) Κάθε ρόλος (SQ\_ROLES) έχει τα δικά του πολλαπλά είδη προσβάσεως σε δεδομένα (SQ\_AUTHORIZATIONS) και κάθε είδος πρόσβασης (SQ\_AUTHORIZATIONS) μπορεί να δοθεί σε πολλούς ρόλους (SQ\_ROLES). Τα διαφορετικά είδη πρόσβασης είναι insert, update, delete, select, access(οθόνη). Λόγω όμως της νέας οντότητας που δημιουργήσαμε (SQ\_ACCESS), συνδέουμε σε κάθε συνδυασμό ρόλου-δραστηριότητας και ένα είδος πρόσβασης. Επομένως ο ρόλος που έχει συγκεκριμένες δραστηριότητες, έχει και συγκεκριμένο είδος πρόσβασης για κάθε δραστηριότητα.

## 2.3 Σχεσιακό μοντέλο δεδομένων

Στην ενότητα αυτή θα παρουσιαστεί το σχεσιακό μοντέλο των δύο υποσυστημάτων που περιγράφηκαν στην προηγούμενη ενότητα.

Με την βοήθεια εργαλείων μετατροπής του μοντέλου οντοτήτων – σχέσεων σε σχεσιακό μοντέλο, δημιουργούνται οι ακόλουθοι σχεσιακοί πίνακες και τα κλειδιά σύνδεσής τους και θα αποτελέσουν τους σχεσιακούς πίνακες της κάθε βάσης δεδομένων.

Η βάση που χρησιμοποιήθηκε στην συγκεκριμένη υλοποίηση είναι η Oracle 10g Release2.

Το schema της βάσης έχει όνομα : HOSPITAL

### 2.3.1 Σχεσιακό μοντέλο συστήματος υποβολής και εκτέλεσης ακτινολογικής εντολής

Οι πίνακες που δημιουργούνται στο σχήμα HOSPITAL είναι οι εξής:

#### 2.3.1.1 IDRIMA: ΙΔΡΥΜΑ

ΟΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
IDRIMA_KOD	Κωδικός Ιδρύματος (Νοσοκομείου, Κέντρου Υγείας, Π.Γ.Ν.)	VARCHAR2 (10)	ΝΑΙ	ΟΧΙ
IDRIMA_TYPE	Τύπος Ιδρύματος ('1' Νοσοκομείο, '2' Π.Γ.Ν., '3' Κέντρο Υγείας)	VARCHAR2 (5)	ΟΧΙ	ΟΧΙ
IDRIMA_TITLE	Τίτλος Ιδρύματος	VARCHAR2 (50)	ΟΧΙ	ΟΧΙ
IDRIMA_DESCR	Περιγραφή Ιδρύματος	VARCHAR2 (150)	ΟΧΙ	ΝΑΙ
ADDRESS	Ταχυδρομική διεύθυνση	VARCHAR2 (150)	ΟΧΙ	ΝΑΙ
TAX_KOD	Ταχ. Κώδικας	VARCHAR2 (10)	ΟΧΙ	ΝΑΙ
PHONE1	Τηλέφωνο 1	NUMBER	ΟΧΙ	ΝΑΙ
PHONE2	Τηλέφωνο 2	NUMBER	ΟΧΙ	ΝΑΙ

Πίνακας 2-1: Περιγραφή πεδίων του πίνακα ΙΔΡΥΜΑ

### 2.3.1.2 DEPARTMENTS: ΤΜΗΜΑΤΑ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
DEPT_KOD	Κωδικός Τμήματος	VARCHAR2 (10)	ΝΑΙ	ΟΧΙ
IDRIMA_KOD	Κωδικός Ιδρύματος	VARCHAR2 (10)	ΟΧΙ	ΝΑΙ
DEPT_TYPE	Τύπος Τμήματος ('ΚΤ' = Κλινικό Τμήμα, 'ΑΤ' = Ακτινολογικό τμήμα, 'ΕΞ' = Εξωτερικά ιατρεία, 'ΕΠ' = Επείγοντα	VARCHAR2 (5)	ΟΧΙ	ΟΧΙ
DEPT_TITLE	Περιγραφή Τμήματος	VARCHAR2 (150)	ΟΧΙ	ΝΑΙ

Πίνακας 2-2: Περιγραφή πεδίων του πίνακα ΤΜΗΜΑΤΑ

### 2.3.1.3 DOCTORS: ΓΙΑΤΡΟΣ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
DOCT_KOD	Κωδικός Γιατρού	VARCHAR2 (10)	ΝΑΙ	ΟΧΙ
EIDIK_KOD	Κωδικός Ειδικότητας Γιατρού	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
DEPT_KOD	Κωδικός Τμήματος που εργάζεται	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
DOCT_NAME	Όνομα γιατρού	VARCHAR2 (20)	ΟΧΙ	ΟΧΙ
DOCT_SURNAME	Επίθετο γιατρού	VARCHAR2 (50)	ΟΧΙ	ΟΧΙ
DOCT_FATHERNAME	Όνομα πατρός	VARCHAR2 (20)	ΟΧΙ	ΝΑΙ
DOCT_EMAIL	e-mail	VARCHAR2 (150)	ΟΧΙ	ΝΑΙ

Πίνακας 2-3: Περιγραφή πεδίων του πίνακα ΓΙΑΤΡΟΣ

### 2.3.1.4 ΕΙΔΙΚΟΤΗΤΕΣ: ΕΙΔΙΚΟΤΗΤΑ ΓΙΑΤΡΟΥ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
EIDIK_KOD	Κωδικός ειδικότητας	VARCHAR2 (10)	ΝΑΙ	ΟΧΙ
EIDIK_DESCR	Περιγραφή Ειδικότητας	VARCHAR2 (100)	ΟΧΙ	ΟΧΙ

Πίνακας 2-4: Περιγραφή πεδίων του πίνακα ΕΙΔΙΚΟΤΗΤΑ\_ΓΙΑΤΡΟΥ

### 2.3.1.5 PATIENTS: ΑΣΘΕΝΗΣ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
PAT_KOD	Κωδικός Ασθενούς	VARCHAR2 (10)	ΝΑΙ	ΟΧΙ
PAT_NAME	Όνομα	VARCHAR2 (20)	ΟΧΙ	ΟΧΙ
PAT_SURNAME	Επίθετο	VARCHAR2 (50)	ΟΧΙ	ΟΧΙ
PAT_FATHER	Όνομα Πατρός	VARCHAR2 (20)	ΟΧΙ	ΝΑΙ
PAT_MOTHER	Όνομα Μητρός	VARCHAR2 (20)	ΟΧΙ	ΝΑΙ
PAT_BIRTHDATE	Ημερομηνία Γέννησης	DATE	ΟΧΙ	ΝΑΙ
PAT_ADDRESS	Διεύθυνση	VARCHAR2 (150)	ΟΧΙ	ΝΑΙ
PAT_TAXKOD	Ταχ. Κώδικας	VARCHAR2 (10)	ΟΧΙ	ΝΑΙ
PAT_PHONE1	Τηλέφωνο 1	NUMBER	ΟΧΙ	ΝΑΙ
PAT_PHONE2	Τηλέφωνο 2	NUMBER	ΟΧΙ	ΝΑΙ
PAT_TAMEIO	Ταμείο Ασθενούς	VARCHAR2 (50)	ΟΧΙ	ΝΑΙ

PAT_AFM	ΑΦΜ	VARCHAR2 (9)	ΟΧΙ	ΝΑΙ
PAT_ARMHTROOY	Αρ. Μητρώου Ασφαλισμένου	VARCHAR2 (15)	ΟΧΙ	ΝΑΙ
PAT_EMAIL	e-mail	VARCHAR2 (100)	ΟΧΙ	ΝΑΙ
PAT_AMKA	Αριθμός μητρώου κοινων. ασφάλισης	VARCHAR2 (20)	ΟΧΙ	ΝΑΙ

**Πίνακας 2-5:** Περιγραφή πεδίων του πίνακα ΑΣΘΕΝΗΣ

### 2.3.1.6 DOCTORS\_PATIENTS: ΑΣΘΕΝΕΙΣ - ΓΙΑΤΡΟΥ

ΟΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
DOCT_KOD	Κωδικός Γιατρού	VARCHAR2 (10)	ΝΑΙ	ΟΧΙ
PAT_KOD	Κωδικός Ασθενούς	VARCHAR2 (20)	ΝΑΙ	ΟΧΙ
VISIT_DATE	Ημερομηνία Επίσκεψης	DATE	ΟΧΙ	ΟΧΙ

**Πίνακας 2-6:** Περιγραφή πεδίων του πίνακα ΑΣΘΕΝΕΙΣ\_ΓΙΑΤΡΟΥ

### 2.3.1.7 PATIENT\_ACCOUNT: ΛΟΓΑΡΙΑΣΜΟΣ ΑΣΘΕΝΟΥΣ

ΟΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
PAT_ACCOUNT	Λογαριασμός ασθενούς	VARCHAR2 (30)	ΝΑΙ	ΟΧΙ
PAT_KOD	Κωδικός Ασθενή	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
CHARGE_DATE	Ημερομηνία Χρέωσης	DATE	ΟΧΙ	ΟΧΙ
CHARGE_POSO	Ποσό	NUMBER	ΟΧΙ	ΟΧΙ

**Πίνακας 2-7:** Περιγραφή πεδίων του πίνακα ΛΟΓΑΡΙΑΣΜΟΣ\_ΑΣΘΕΝΟΥΣ

### 2.3.1.8 RAD\_REQUESTS: ΕΝΤΟΛΕΣ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ

ΟΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
RAD_ID	Κωδικός Εξέτασης	NUMBER	ΝΑΙ	ΟΧΙ
PAT_KOD	Κωδικός Ασθενή	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
DOCT_KOD	Κωδικός Θεράπων Γιατρού	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
RAD_APPOINT_DATE	Ημερομηνία Ραντεβού	DATE	ΟΧΙ	ΟΧΙ
RAD_APPOINT_TIME	Ωρα Ραντεβού	VARCHAR2(5)	ΟΧΙ	ΟΧΙ
DEPT_KOD	Κωδικός Ακτινολογικού τμήματος	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
NOTES	Σχόλια Γιατρού	VARCHAR2 (2000)	ΟΧΙ	ΟΧΙ
RAD_DOCT_KOD	Κωδ. Ακτινολόγου	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
RAD_DATE_CLOSED	Ημ/νία εκτέλεσης εξέτασης	DATE	ΟΧΙ	ΟΧΙ

**Πίνακας 2-8:** Περιγραφή πεδίων του πίνακα ΕΝΤΟΛΕΣ\_ΑΚΤΙΝΟΛΟΓΙΚΩΝ\_ΕΞΕΤΑΣΕΩΝ

### 2.3.1.9 RAD\_REPORTS: ΑΚΤΙΝΟΛΟΓΙΚΕΣ ΑΝΑΦΟΡΕΣ

ΟΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
REP_ID	Κωδικός Αναφοράς	NUMBER	ΝΑΙ	ΟΧΙ
PAT_KOD	Κωδικός Ασθενή	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
DOCT_KOD	Κωδικός Θεράπων Γιατρού	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
RAD_ID	Κωδικός ακτινολογικής εντολής	NUMBER	ΟΧΙ	ΟΧΙ
REP_NOTES	Γνωμάτευση	VARCHAR2(2000)	ΟΧΙ	ΟΧΙ
RAD_PHOTO	Φωτογραφία ακτινογραφίας	BLOB	ΟΧΙ	ΟΧΙ

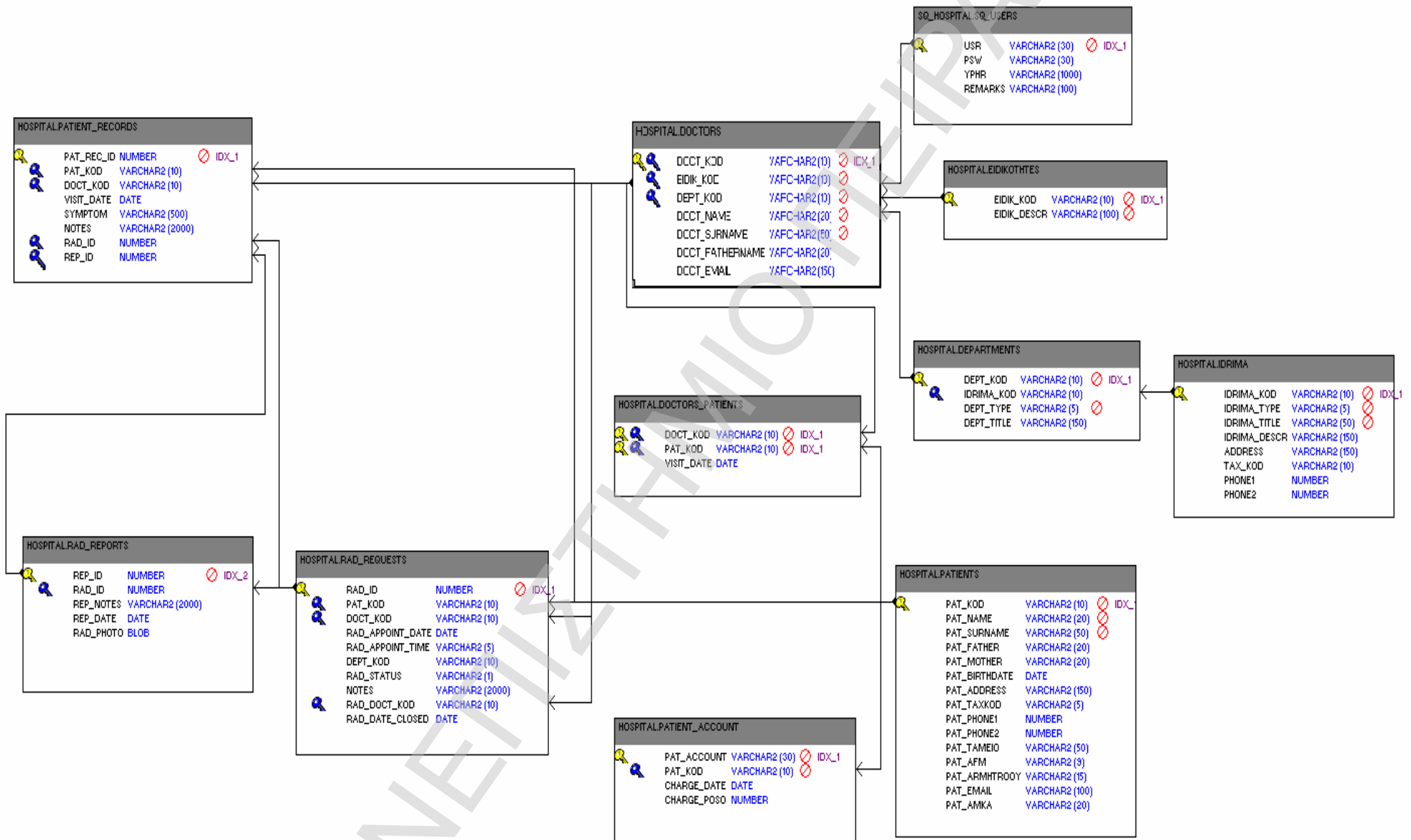
**Πίνακας 2-9:** Περιγραφή πεδίων του πίνακα ΑΚΤΙΝΟΛΟΓΙΚΕΣ\_ΑΝΑΦΟΡΕΣ

### 2.3.1.10 PATIENT\_RECORDS: ΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ ΑΣΘΕΝΟΥΣ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
PAT_REC_ID	Κωδικός Εγγραφής	NUMBER	ΝΑΙ	ΟΧΙ
PAT_KOD	Κωδικός Ασθενή	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
DOCT_KOD	Κωδικός Θεράπων Γιατρού	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
VISIT_DATE	Ημερομηνία Επίσκεψης	DATE	ΟΧΙ	ΟΧΙ
SYMPTOM	Σύμπτωμα	VARCHAR2(500)	ΟΧΙ	ΟΧΙ
NOTES	Σχόλια	VARCHAR2(2000)	ΟΧΙ	ΟΧΙ
RAD_ID	Κωδικός εντολής ακτινολογικής εξέτασης	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
REP_ID	Κωδικός ακτινολογικής αναφοράς	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ

Πίνακας 2-10: Περιγραφή πεδίων του πίνακα ΙΑΤΡΙΚΟΣ\_ΦΑΚΕΛΟΣ\_ΑΣΘΕΝΟΥΣ

## ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ



Εικόνα 2-3: Σχεσιακό Μοντέλο Υποσυστήματος

Στην [εικόνα 2-3](#), όπου παρουσιάζεται το σχεσιακό μοντέλο του συστήματος, αρχικά φαίνονται οι πίνακες και τα πεδία που χαρακτηρίζουν τις εγγραφές τους. Δίπλα από κάθε πεδίο, παρατίθεται ο τύπος του πεδίου και το μέγεθός του, όπου αυτό είναι απαραίτητο. Επιπλέον, φαίνονται τα κλειδιά των πινάκων και οι συνδέσεις τους με αυτά. Το «κίτρινο» κλειδί δείχνει το πρωτεύον κλειδί του κάθε πίνακα, ενώ το «μπλε» κλειδί υποδηλώνει ότι το πεδίο είναι εξωτερικό κλειδί από τον διασυνδεδεμένο πίνακα.

π.χ. στην σχέση IDRYMA-DEPARTMENTS όπου στο μοντέλο οντοτήτων σχέσεων η σχέση μεταξύ τους είχε χαρακτηριστεί (1-N), (ένα ίδρυμα έχει πολλά τμήματα και ένα τμήμα ανήκει σε ένα μόνο ίδρυμα), το πρωτεύον κλειδί του πίνακα IDRIMA «κατέβηκε» ως εξωτερικό κλειδί στον πίνακα DEPARTMENTS.

### 2.3.2 Σχεσιακό μοντέλο συστήματος ασφάλειας

Στο μοντέλο ασφάλειας του συστήματος, το σχήμα της βάσης δεδομένων ασφάλειας ονομάζεται SQ\_HOSPITAL.

Οι πίνακες που δημιουργούνται για το μοντέλο της βάσης δεδομένων ασφάλειας (SQ\_HOSPITAL) είναι οι εξής:

#### 2.3.2.1 SQ\_USERS: ΧΡΗΣΤΕΣ ΣΥΣΤΗΜΑΤΟΣ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
USR	Κωδικός Χρήστη	VARCHAR2 (30)	ΝΑΙ	ΟΧΙ
PSW	Συνθηματικό χρήστη	VARCHAR2 (30)	ΟΧΙ	ΟΧΙ
ΥΡΗΡ	Υπηρεσία που ανήκει	VARCHAR2(10)	ΟΧΙ	ΝΑΙ
REMARKS	Σχόλια	VARCHAR2(100)	ΟΧΙ	ΝΑΙ

Πίνακας 2-11: Περιγραφή πεδίων του πίνακα ΧΡΗΣΤΕΣ\_ΣΥΣΤΗΜΑΤΟΣ

#### 2.3.2.2 SQ\_ROLES: ΡΟΛΟΙ ΣΥΣΤΗΜΑΤΟΣ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
ROLE_KOD	Κωδικός Ρόλου	VARCHAR2 (50)	ΝΑΙ	ΟΧΙ
ROLE_DESCR	Περιγραφή Ρόλου	VARCHAR2 (500)	ΟΧΙ	ΟΧΙ

Πίνακας 2-12: Περιγραφή πεδίων του πίνακα ΡΟΛΟΙ\_ΣΥΣΤΗΜΑΤΟΣ

#### 2.3.2.3 SQ\_TASKS: ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΣΥΣΤΗΜΑΤΟΣ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
TASK_KOD	Κωδικός Δραστηριότητας	VARCHAR2 (50)	ΝΑΙ	ΟΧΙ
TASK_DESCR	Περιγραφή Δραστηριότητας	VARCHAR2 (500)	ΟΧΙ	ΟΧΙ

Πίνακας 2-13: Περιγραφή πεδίων του πίνακα ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ\_ΣΥΣΤΗΜΑΤΟΣ

#### 2.3.2.4 SQ\_AUTHORIZATIONS: ΕΙΔΗ ΠΡΟΣΒΑΣΕΩΝ ΣΥΣΤΗΜΑΤΟΣ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
ACCESS_TYPE	Τύπος Πρόσβασης 'I' = Insert, 'D' = Delete, 'U' = Update, 'S' = Select-read	CHAR(1)	ΝΑΙ	ΟΧΙ
DESCRIPTION	Περιγραφή πρόσβασης	VARCHAR2 (20)	ΟΧΙ	ΟΧΙ

Πίνακας 2-14: Περιγραφή πεδίων του πίνακα ΕΙΔΗ\_ΠΡΟΣΒΑΣΕΩΝ

### 2.3.2.5 USER\_ROLES: ΧΡΗΣΤΕΣ - ΡΟΛΟΙ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
USER_ROLES_ID	Κωδικός εγγραφής	NUMBER	ΝΑΙ	ΟΧΙ
USERNAME	Κωδικός Χρήστη	VARCHAR2 (10)	ΟΧΙ	ΟΧΙ
ROLE_KOD	Κωδικός ρόλου	VARCHAR2 (50)	ΟΧΙ	ΟΧΙ

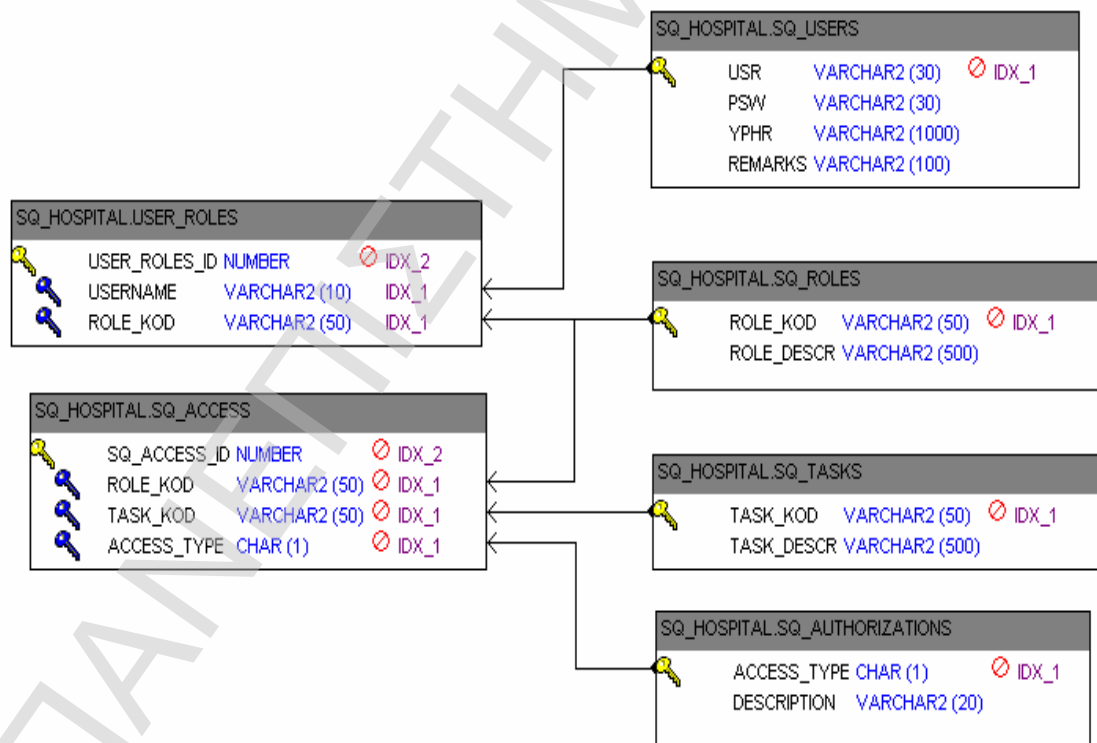
Πίνακας 2-15: Περιγραφή πεδίων του πίνακα ΧΡΗΣΤΕΣ\_ΡΟΛΟΙ

### 2.3.2.6 SQ\_ACCESS: ΡΟΛΟΙ-ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ-ΠΡΟΣΒΑΣΕΙΣ

ΌΝΟΜΑ	ΠΕΡΙΓΡΑΦΗ	ΤΥΠΟΣ	PK	NULL
SQ_ACCESS_ID	Κωδικός εγγραφής	NUMBER	ΝΑΙ	ΟΧΙ
ROLE_KOD	Κωδικός ρόλου	VARCHAR2 (50)	ΟΧΙ	ΟΧΙ
TASK_KOD	Κωδικός Δραστηριότητας	VARCHAR2 (50)	ΟΧΙ	ΟΧΙ
ACCESS_TYPE	Κωδικός τύπου πρόσβασης	CHAR(1)	ΟΧΙ	ΟΧΙ

Πίνακας 2-16: Περιγραφή πεδίων του πίνακα ΡΟΛΟΙ\_ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ\_ΠΡΟΣΒΑΣΕΙΣ

Στην [εικόνα 2-4](#), όπου παρουσιάζεται το σχεσιακό μοντέλο του συστήματος ασφάλειας, αρχικά φαίνονται οι πίνακες και τα πεδία που χαρακτηρίζουν τις εγγραφές τους. Δίπλα από κάθε πεδίο, παρατίθεται ο τύπος του πεδίου και το μέγεθός του, όπου αυτό είναι απαραίτητο. Επιπλέον, φαίνονται τα κλειδιά των πινάκων και οι συνδέσεις τους με αυτά. Το «κίτρινο» κλειδί δείχνει το πρωτεύον κλειδί του κάθε πίνακα, ενώ το «μπλε» κλειδί υποδηλώνει ότι το πεδίο είναι εξωτερικό κλειδί από τον διασυνδεδεμένο πίνακα.



Εικόνα 2-4: Σχεσιακό μοντέλο ασφάλειας συστήματος

Στο παράρτημα 1, υπάρχει ο κώδικας σε SQL που δημιουργεί όλους τους παραπάνω πίνακες στην βάση HOSPITAL και SQ\_HOSPITAL αντίστοιχα.

## 3<sup>ο</sup> ΚΕΦΑΛΑΙΟ

ΕΓΚΑΤΑΣΤΑΣΗ - ΔΙΑΜΟΡΦΩΣΗ  
ΣΥΣΤΗΜΑΤΟΣ ΡΟΗΣ ΕΡΓΑΣΙΑΣ

Installation - Configuration of  
Workflow System



---

## 3 ΕΓΚΑΤΑΣΤΑΣΗ – ΔΙΑΜΟΡΦΩΣΗ ΣΥΣΤΗΜΑΤΟΣ ΡΟΗΣ ΕΡΓΑΣΙΑΣ

### 3.1 Εισαγωγή

Το εργαλείο MQSeries Workflow, παρέχει ένα σύστημα αυτοματοποίησης της διαδικασίας διαχείρισης ατόμων, δεδομένων, εφαρμογών και επιχειρησιακών διαδικασιών μέσα σε έναν οργανισμό, συμπεριλαμβανομένων εξωτερικών συνεργατών μέσω του internet ή του intranet και extranet.

Το MQSeries Workflow, διαχειρίζεται τις ροές διαδικασιών μεταξύ πολλαπλών δραστηριοτήτων, κάνοντας τις επιχειρησιακές διαδικασίες σαφείς, ορατές και μετατρέψιμες. Μπορεί να αυτοματοποιήσει όλα τα βήματα μέσα σε μια επιχειρησιακή διαδικασία από την αρχή ως το τέλος, υποστηρίζοντας την εργασία που εκτελούν άτομα ή/και εφαρμογές ώστε να εξυπηρετείται κάθε αίτημα του πελάτη (ή της εφαρμογής).

Δίχως το Workflow, μπορούμε να αυτοματοποιήσουμε μια διαδικασία γράφοντας νέο επιπλέον κώδικα , αλλά:

Τότε δεν θα γνωρίζαμε ακριβώς την διαδικασία, αφού η λογική της θα είναι κρυμμένη μέσα στον κώδικα της εφαρμογής.

Μπορούμε να αλλάξουμε μια διαδικασία όπως προστάζει μια απαίτηση της επιχείρησης; Μόνο αλλάζοντας τον κώδικα της εφαρμογής.

Μπορείς να δεις ανά πάσα στιγμή την κατάσταση μιας διαδικασίας ενώ ακόμα τρέχει και εξελίσσεται; Θα έπρεπε να αναμείξουμε και να ελέγξουμε όλες τις εμπλεκόμενες εφαρμογές ώστε να αποφανθούμε για κάτι τέτοιο.

Το MQSeries Workflow, επιτρέπει στα τμήματα της επιχείρησης να σχεδιάζουν και να αλλάζουν τις επιχειρησιακές διαδικασίες της επιχείρησης, χωρίς την βοήθεια των προγραμματιστών. Η ανάπτυξη των συστημάτων ροής εργασιών μπορεί να αποκεντρωθεί, που σημαίνει ότι τα τμήματα μπορούν μόνο τους να σχεδιάσουν, μοντελοποιήσουν και αναπτύξουν τις περισσότερες διαδικασίες τους, ή με περιορισμένες αλλαγές των εφαρμογών.

Το Workflow επιτρέπει να διαχωριστεί η επιχειρησιακή ροή από την λογική της εφαρμογής.

## 3.2 Διαμόρφωση και αρχικοποίηση του συστήματος ροής εργασίας

Η υλοποίηση του συστήματος ροής εργασίας που θα υποστηρίζει την διαδικασία που περιγράφηκε στο κεφάλαιο 1, θα υλοποιηθεί μέσω των εργαλείων ανάπτυξης της IBM και πιο συγκεκριμένα:

A) IBM WebSphere MQ WorkFlow Buildtime Version 3.6.0 (ή MQ Series Workflow Buildtime). Με το εργαλείο αυτό, επιτυγχάνουμε εύκολα την γραφική απεικόνιση της ροής της διαδικασίας, αλλά ταυτόχρονα δηλώνουμε και όλη την πληροφορία που χρειάζεται το σύστημα, ώστε να ολοκληρωθεί με επιτυχία η διαδικασία.

B) IBM DB2 UNIVERSAL DATABASE PERSONAL EDITION. Είναι η βάση δεδομένων, που χρησιμοποιείται για την αποθήκευση των δεδομένων του συστήματος ροής εργασιών, όπως αυτά δηλώθηκαν κατά το buildtime.

Γ) IBM WebSphere MQ 6.0. Ο Server επικοινωνίας του Workflow με την βάση της DB2.

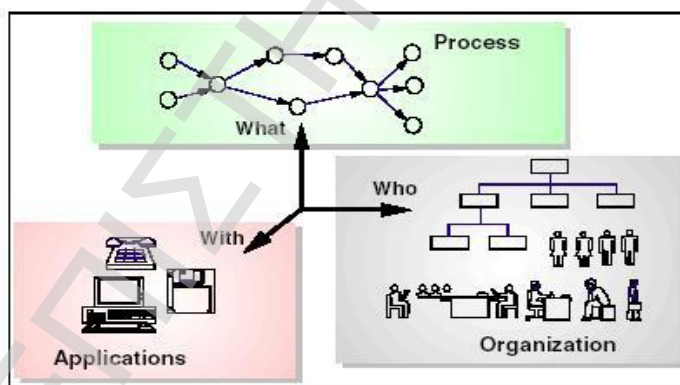
Δ) IBM WebSphere MQ WorkFlow Client (ή MQ Series Workflow Client). Εργαλείο χρήστη όπου μπορεί να εκτελέσει τις δραστηριότητες της διαδικασίας.

Ε) Oracle JDeveloper 10.1.3.2.0. Εργαλείο ανάπτυξης εφαρμογών java.

ΣΤ) Oracle database 10g Release 2. Η βάση δεδομένων που χρησιμοποιήθηκε στο κεφάλαιο 4, για τα δεδομένα της εφαρμογής.

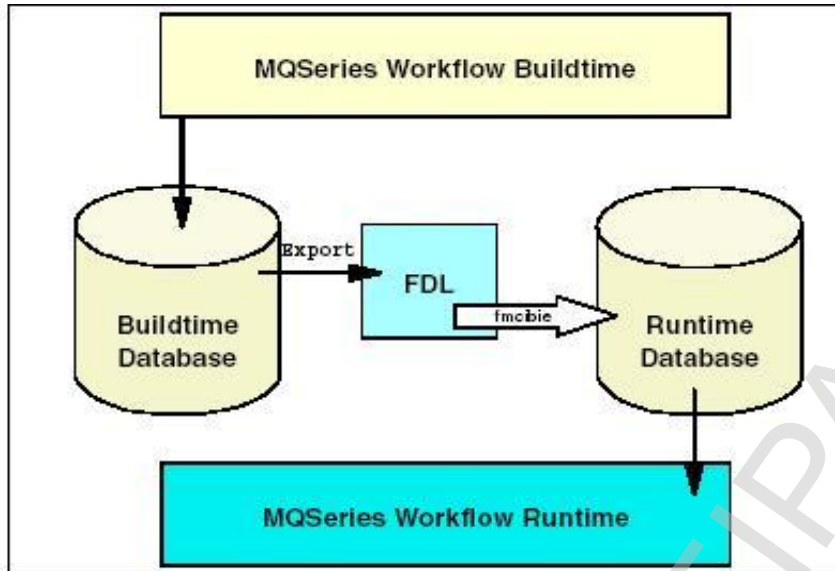
Z) J2SE 1.5.0\_18. Η έκδοση της java που χρησιμοποιήθηκε προκειμένου να υλοποιηθούν τα προγράμματα της εφαρμογής που καλούνται μέσα από το σύστημα ροής εργασιών.

Αρχικά μέσω του εργαλείου MQSeries Workflow Buildtime, δημιουργείται το μοντέλο δραστηριοτήτων και ρυθμίζονται, οι δραστηριότητες που θα εκτελούνται, η σειρά που αυτές θα εκτελούνται, τα άτομα που θα τις εκτελούν, οι ρόλοι τους, τα προγράμματα που θα καλούνται προς εκτέλεση από κάθε δραστηριότητα και οι δομές δεδομένων που αυτές θα ανταλλάσσουν.



Εικόνα 3-1: Εμπλεκόμενοι ενός συστήματος ροής εργασίας

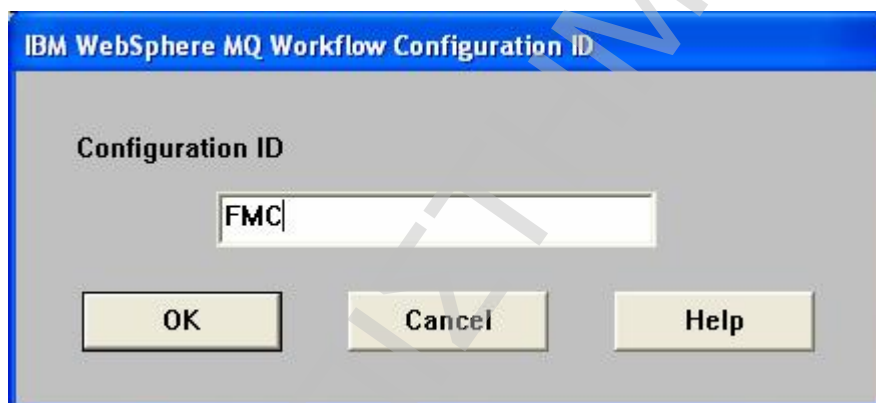
Όλες οι παραπάνω ρυθμίσεις αποθηκεύονται στην ανάλογη – καθορισμένη βάση του Buildtime, όπου στην προκειμένη υλοποίηση είναι η DB2. Αφού ολοκληρωθεί ο σχεδιασμός της διαδικασίας, στη συνέχεια από το Buildtime κάνοντας export, δημιουργείται το fdl αρχείο που περιέχει όλες τις ρυθμίσεις του μοντέλου. Το αρχείο αυτό παρατίθεται στο παράρτημα 3. Το συγκεκριμένο αρχείο γίνεται import στην runtime βάση δεδομένων με την εντολή fmcibie με αποτέλεσμα την δυνατότητα της έναρξης του MQSeries Workflow Runtime. Από το σημείο αυτό και μετά, ο χρήστης μπορεί να ανοίξει τον MQSeries Workflow Client για να εκτελέσει τις δραστηριότητες που του ανήκουν. Η παραπάνω διαδικασία περιγράφεται στην [εικόνα 3-2](#).



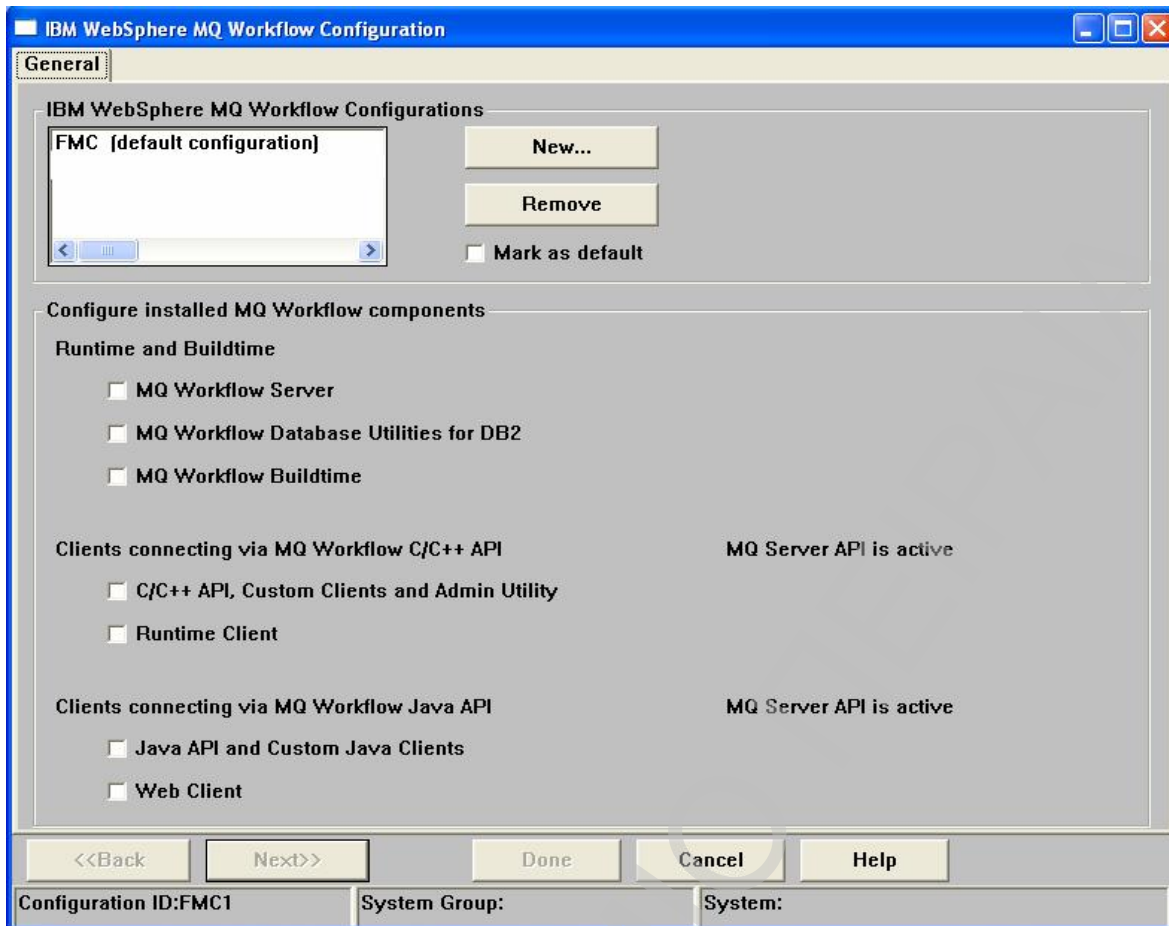
Εικόνα 3-2: Βασικά μέρη του MQ Workflow

Μετά την εγκατάσταση των παραπάνω εργαλείων, ο χρήστης- administrator είναι έτοιμος να ρυθμίσει το Configuration Tool ακολουθώντας την διαδρομή : Start – IBM – IBM WebSphere MQ Workflow – IBM WebSphere MQ Workflow Configuration Utility. Από την στιγμή που θα ανοίξει, ακολουθούν τα παρακάτω βήματα :

**Βήμα 1ο** : Πατώντας το κουμπί New δημιουργούμε το νέο Configuration ID = FMC, [Εικόνα 3-3](#) και επιλέγουμε όλα τα components τα οποία είναι ενεργοποιημένα, [Εικόνα 3-4](#).

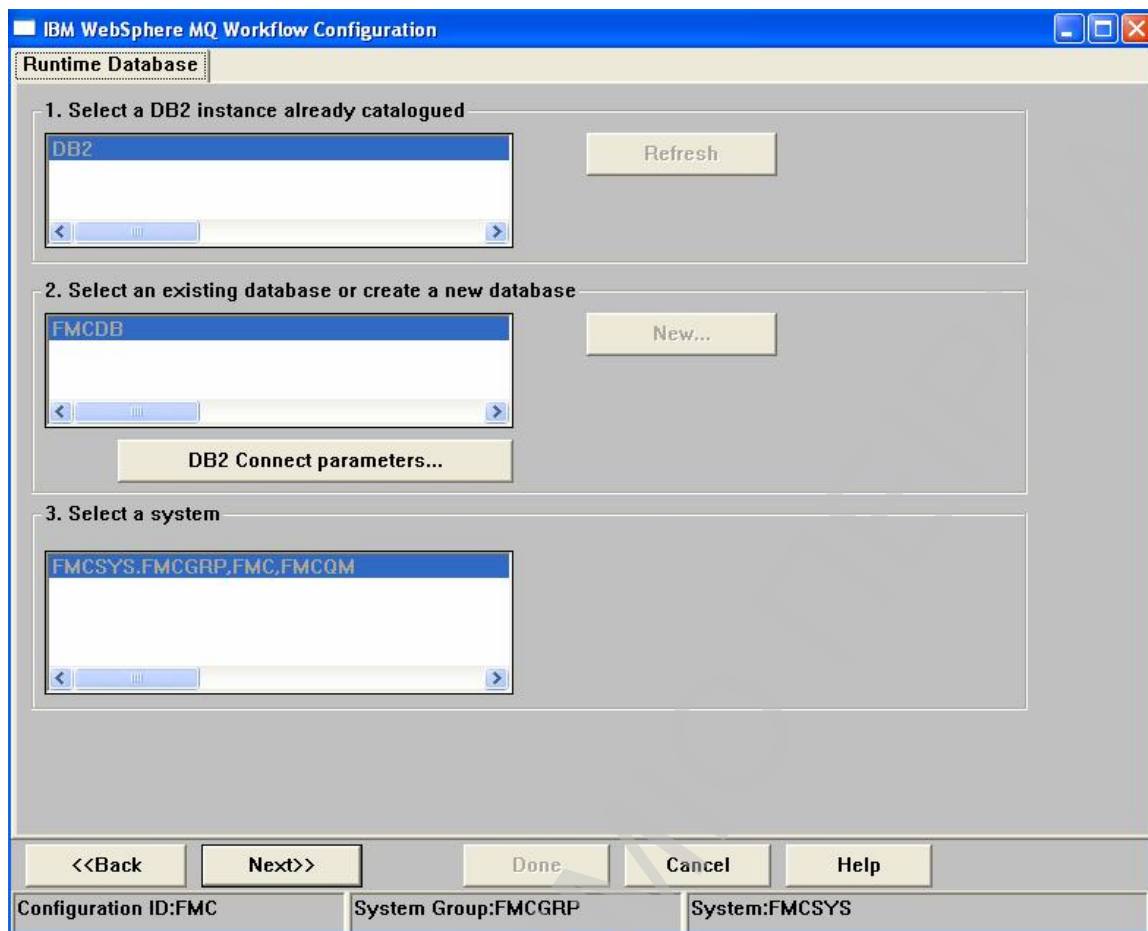


Εικόνα 3-3: Νέο Configuration ID



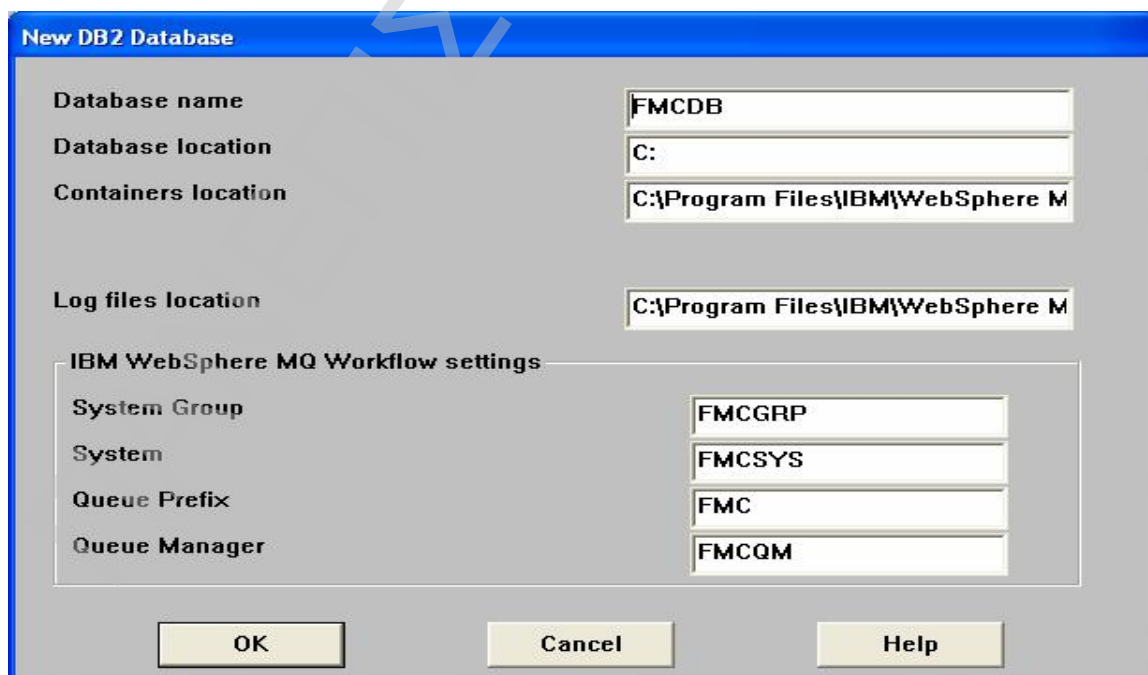
Εικόνα 3-4: Επιλογή όλων των components

**Βήμα 2<sup>ο</sup>:** Ξεκινάμε το configuration της Runtime database

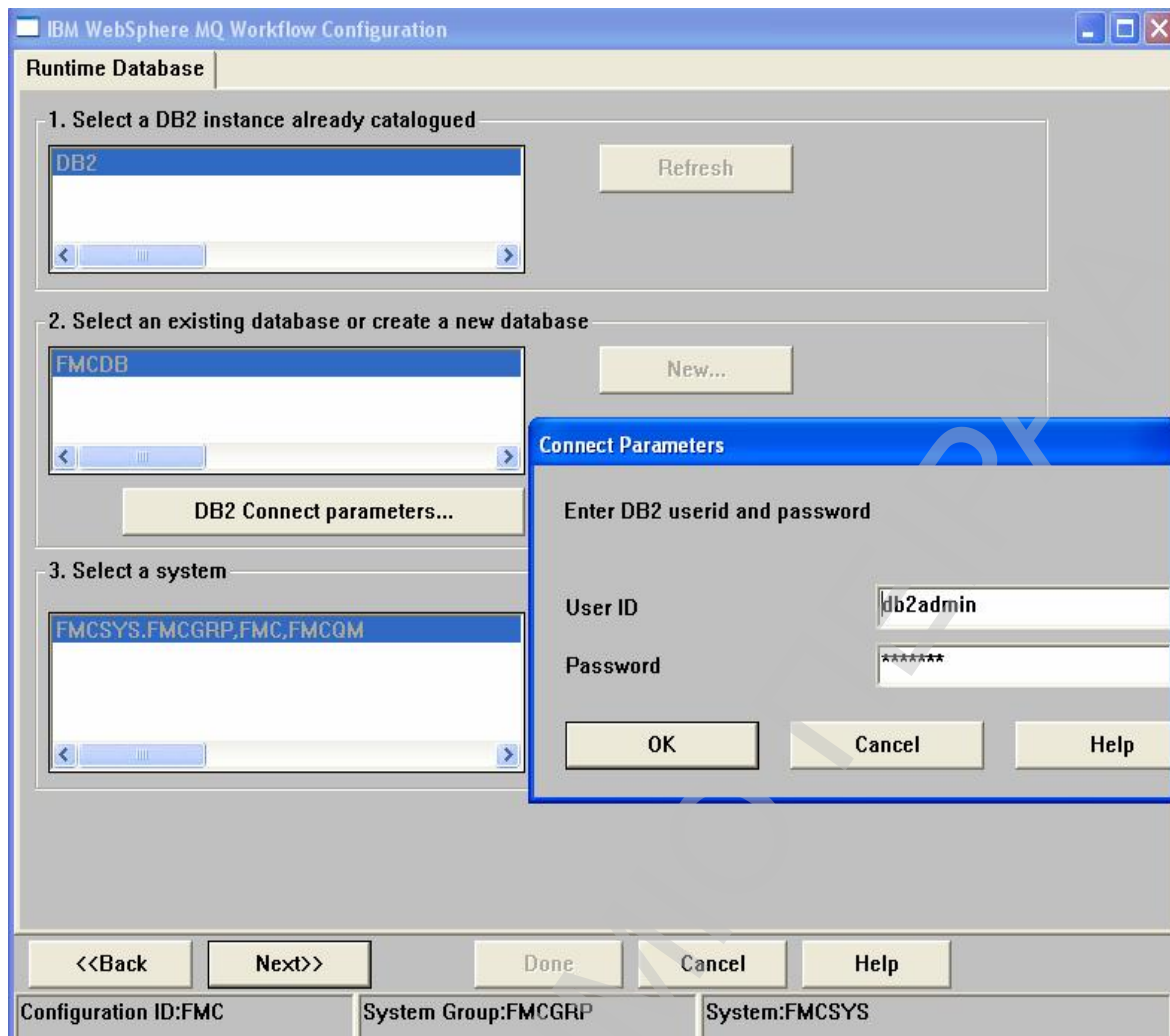


**Εικόνα 3-5:** Configuration of Runtime Database

Επιλέγουμε την DB2 instance και πατώντας το πλήκτρο New, δημιουργούμε μια νέα σύνδεση με την βάση δεδομένων, όπως φαίνεται στην [εικόνα 3-6](#).



**Εικόνα 3-6:** Νέα σύνδεση με την βάση δεδομένων του workflow

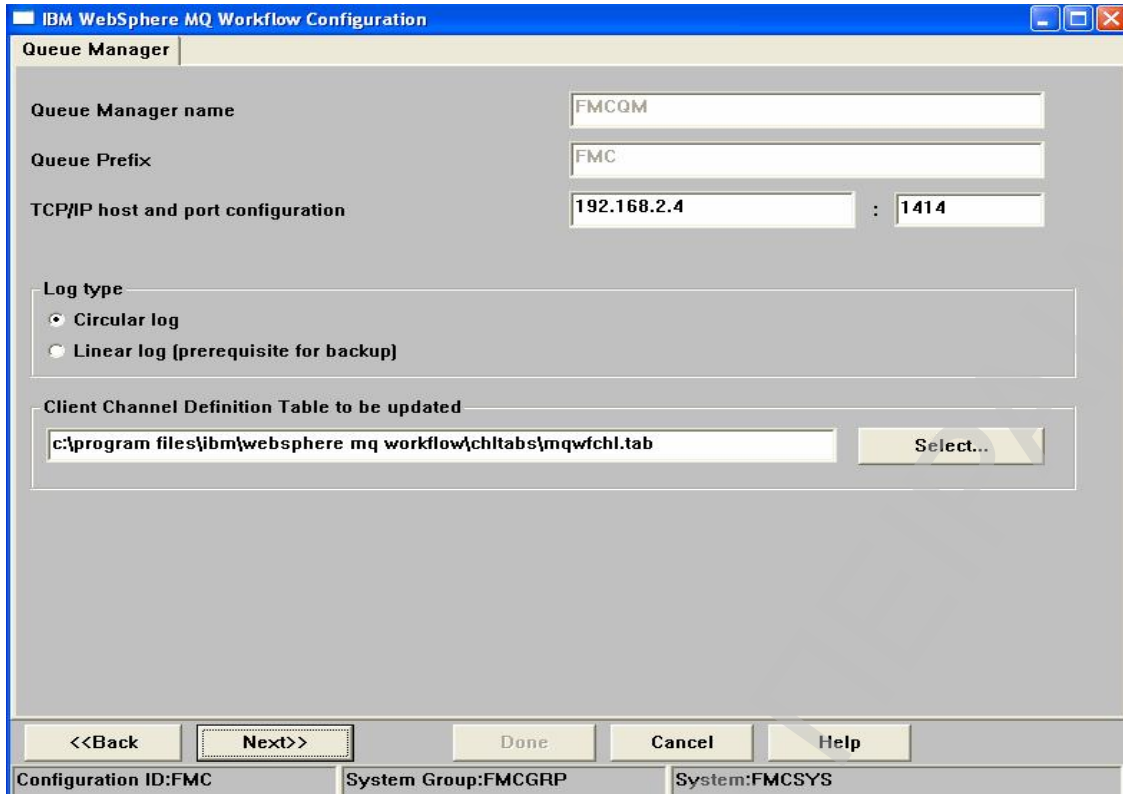


**Εικόνα 3-7:** Connect parameters for runtime database

Πατώντας το πλήκτρο «DB2 Connect parameters», δηλώνουμε το userID και password για την σύνδεση με την βάση. USER ID= db2admin, password=despina (Εικόνα 3-7)

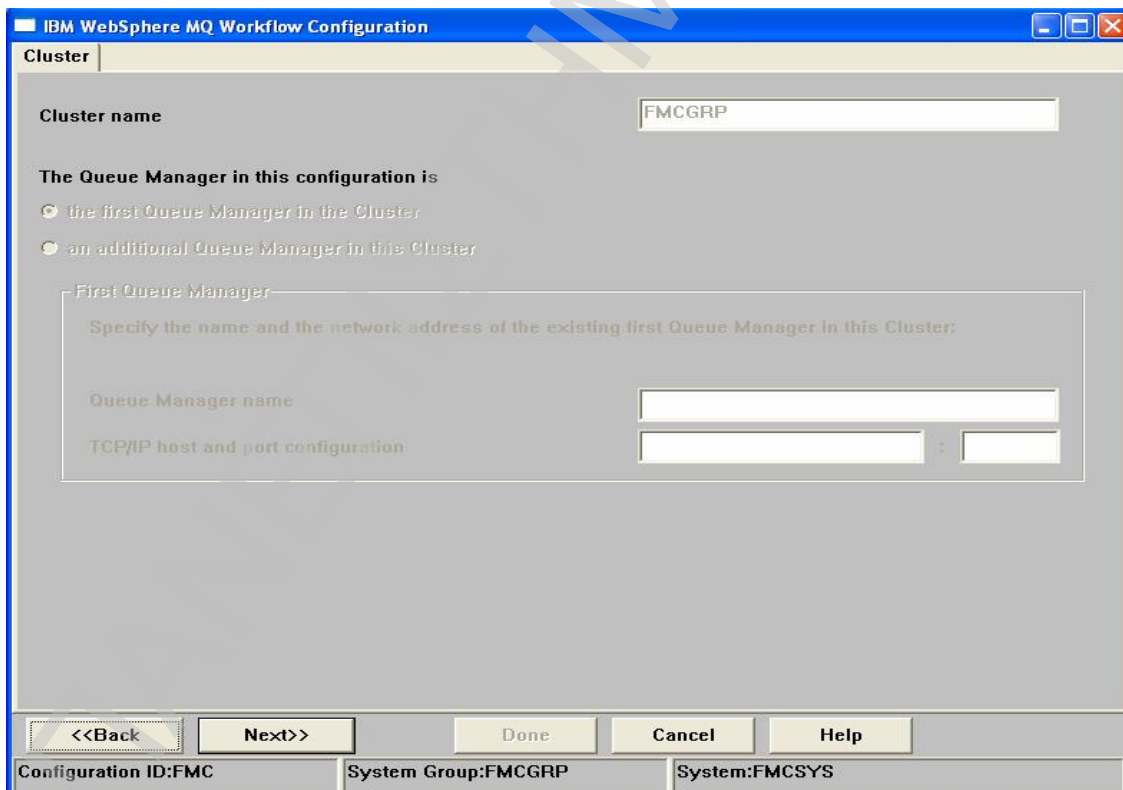
**Βήμα 3<sup>ο</sup>:** Ρύθμιση του Queue Manager

Όπως φαίνεται και στην εικόνα 3-8, συμπληρώνουμε την TCP/IP host address του server (192.168.2.4) και το port (1414) στο οποίο θα «ακούει».



**Εικόνα 3-8:** Configuration του Queue Manager

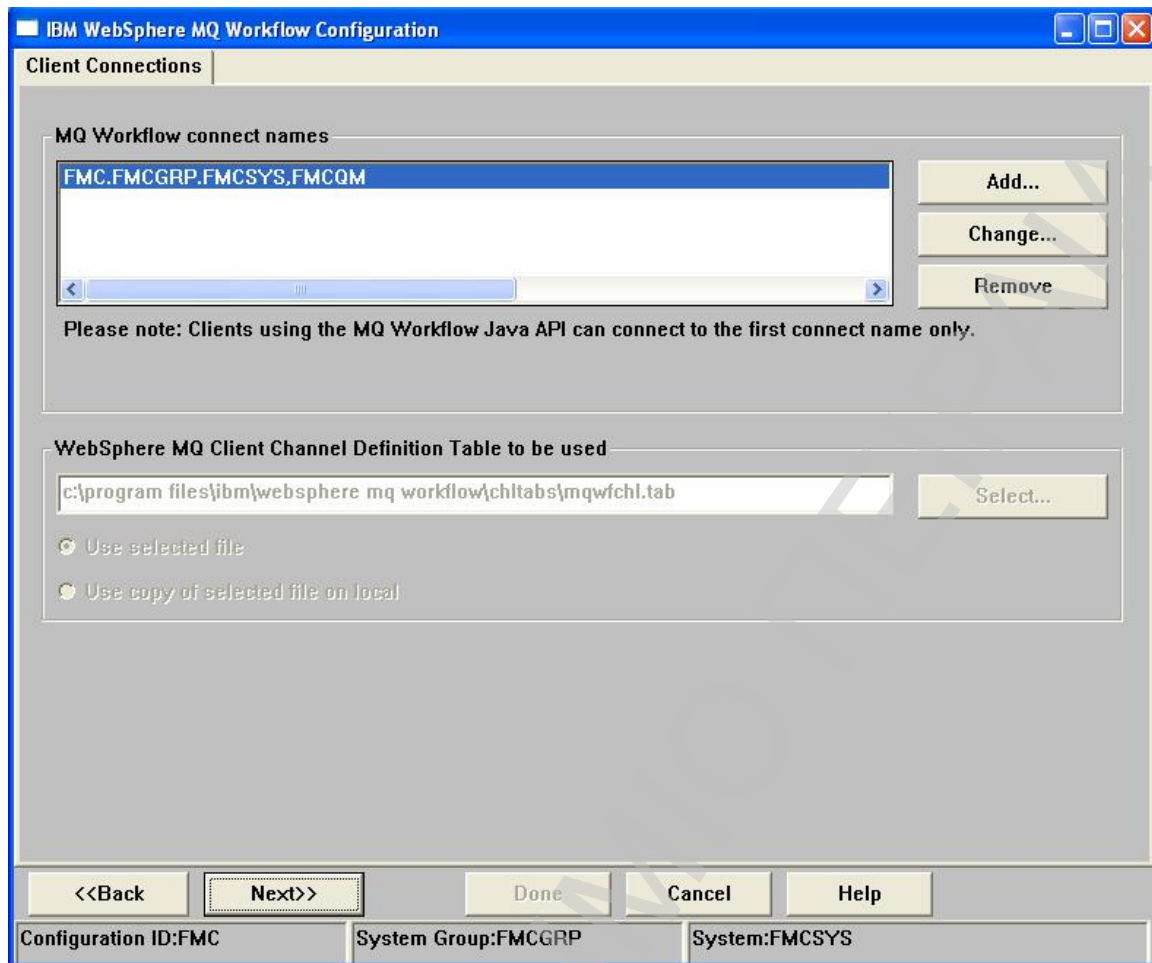
**Βήμα 4<sup>ο</sup>:** Ρύθμιση του Cluster όπως ακριβώς φαίνεται στην [εικόνα 3-9](#).



**Εικόνα 3-9:** Configuration of Cluster



**Βήμα 5<sup>ο</sup>:** Ρύθμιση των Client Connections, όπως φαίνεται στην [εικόνα 3-10](#).

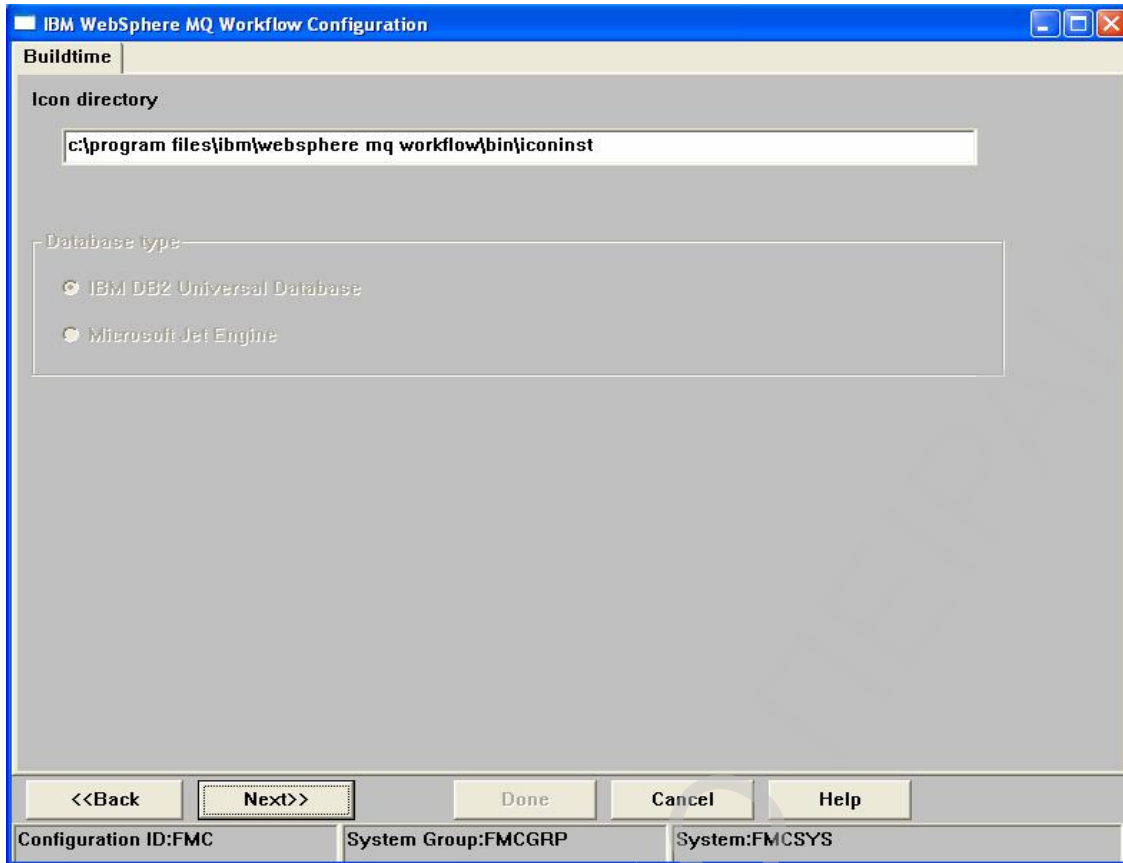


**Εικόνα 3-10:** Ρύθμιση των Client Connections

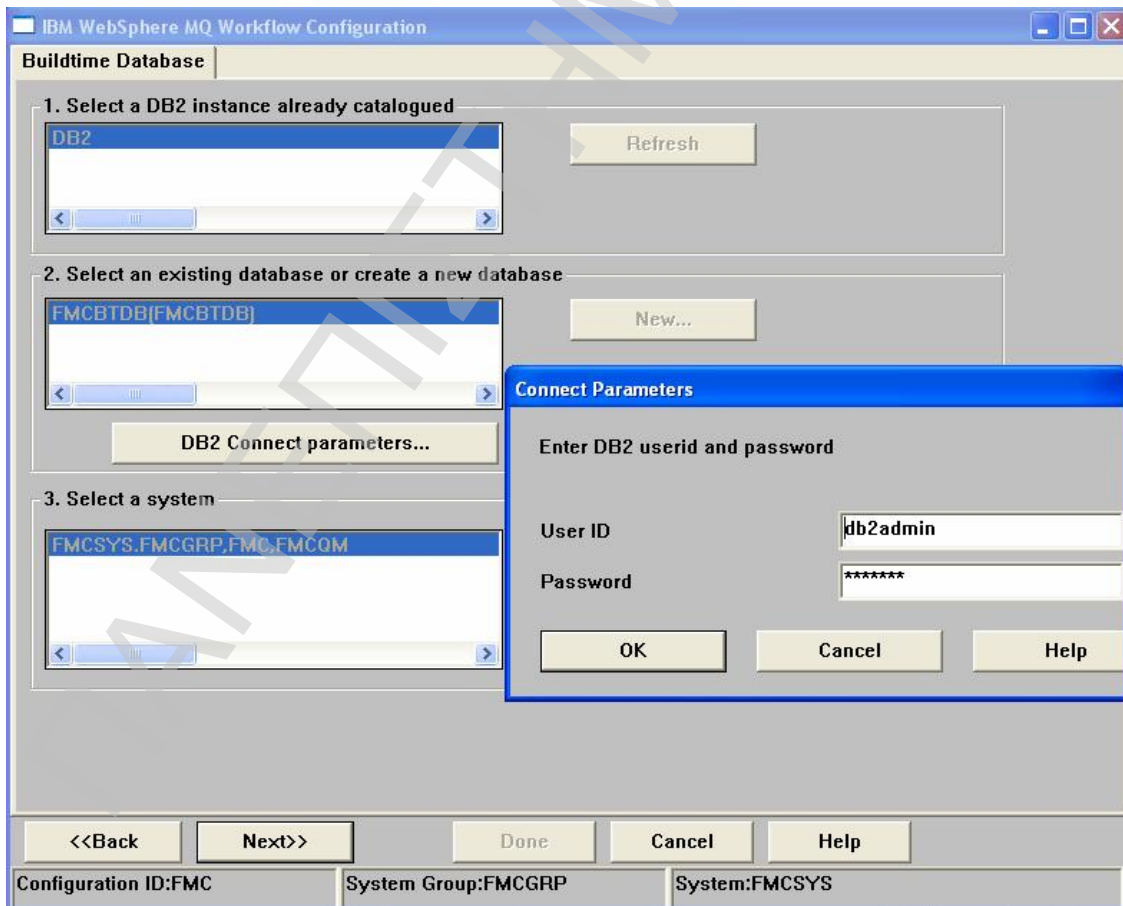
**Βήμα 6<sup>ο</sup>:** Στην ετικέτα Buildtime επιλέγω η Βάση να είναι DB2, ενώ στην ετικέτα Buildtime Database φτιάχνω τις ίδιες ρυθμίσεις με το Βήμα 2ο .

Οι δύο αυτές ρυθμίσεις, φαίνονται στις επόμενες εικόνες που ακολουθούν ([3-11](#), [3-12](#))



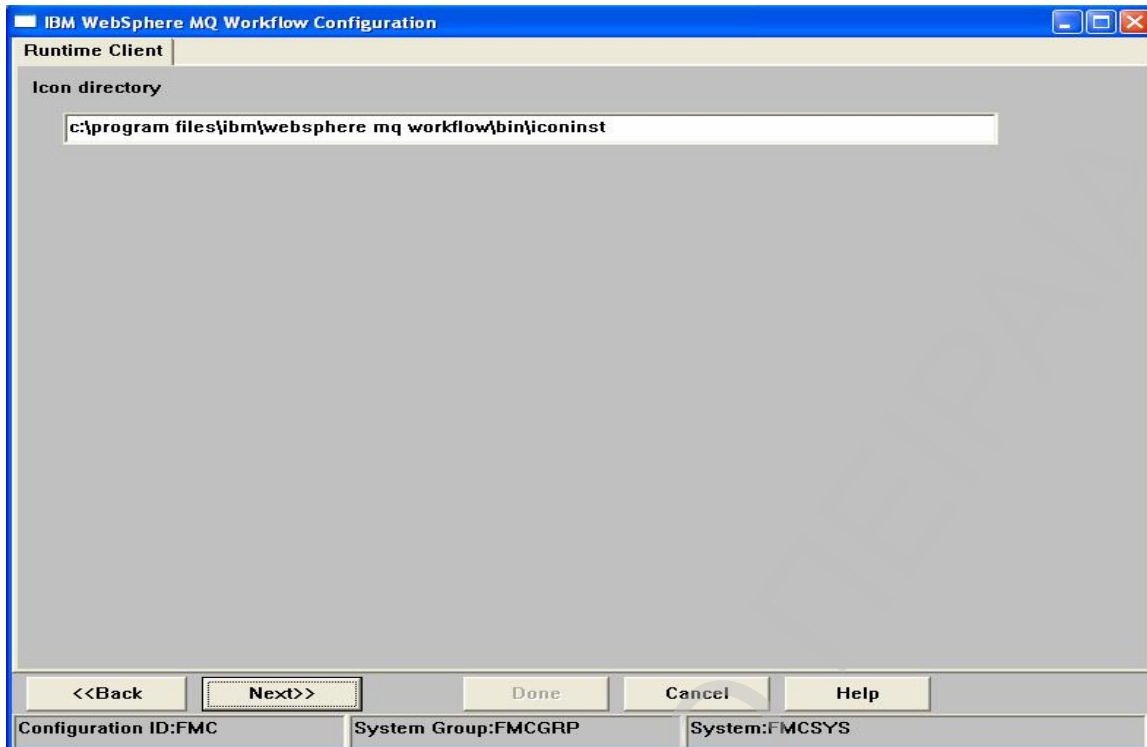


Εικόνα 3-11: Buildtime configuration



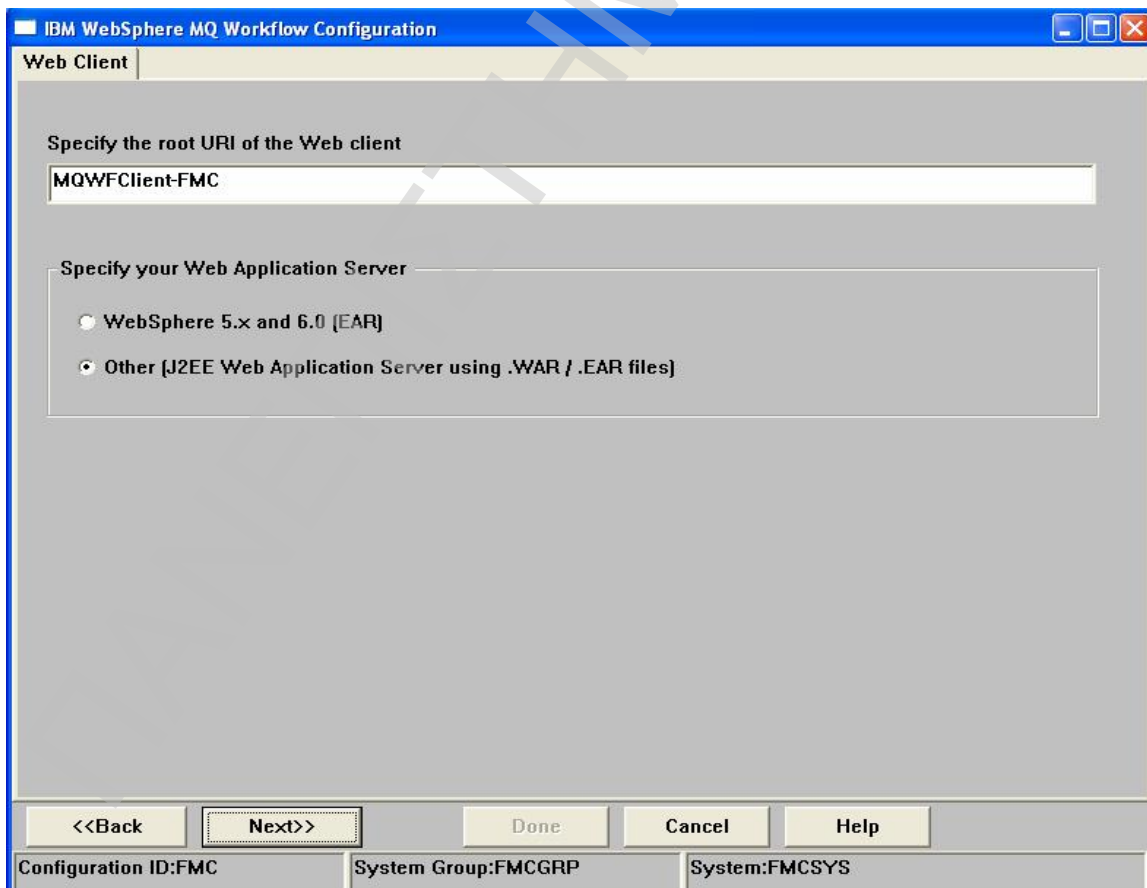
Εικόνα 3-12: Buildtime Database Configuration

**Βήμα 7<sup>ο</sup>:** Ρύθμιση του Runtime Client (Εικόνα 3-13)



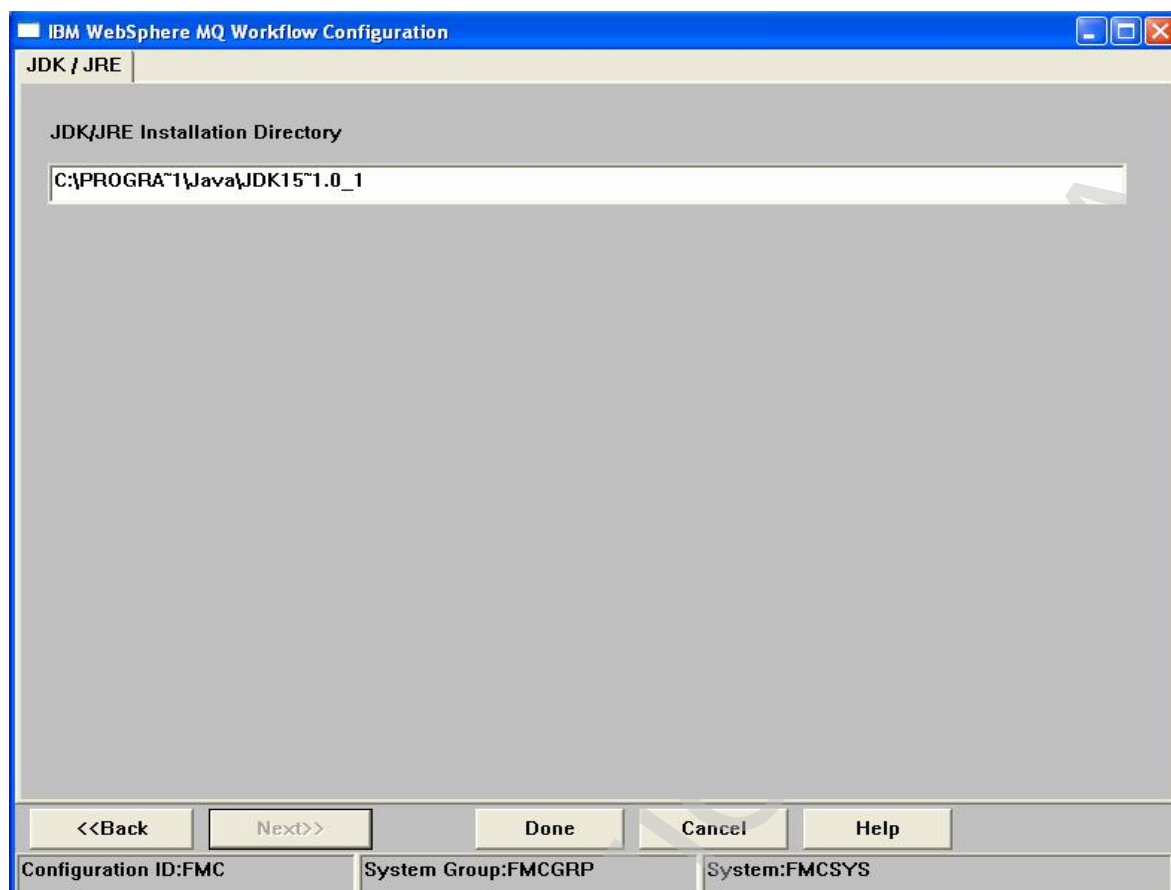
**Εικόνα 3-13:** Configuration of Runtime Client

**Βήμα 8<sup>ο</sup>:** Ρύθμιση του Web Client (Εικόνα 3-14)



**Εικόνα 3-14:** Configuration of Web Client

**Βήμα 9<sup>ο</sup>:** Δήλωση του JDK/JRE (Εικόνα 3-15)



**Εικόνα 3-15:** Configuration of JDK/JRE

### 3.3 Καθορισμός των δραστηριοτήτων στην buildtime βάση δεδομένων

Φτάνοντας στο στάδιο του buildtime και αφού προηγουμένως έχουν ολοκληρωθεί όλα τα προηγούμενα βήματα του configuration που περιγράφηκαν στην προηγούμενη ενότητα, έχουμε την δυνατότητα να ορίσουμε το μοντέλο του συστήματός μας, μέσα από το εργαλείο του IBM WebSphere MQ Workflow Buildtime.

Με την εισαγωγή μας στο εργαλείο του buildtime , μας ζητάει username και password ,όπου είναι τα ακόλουθα:

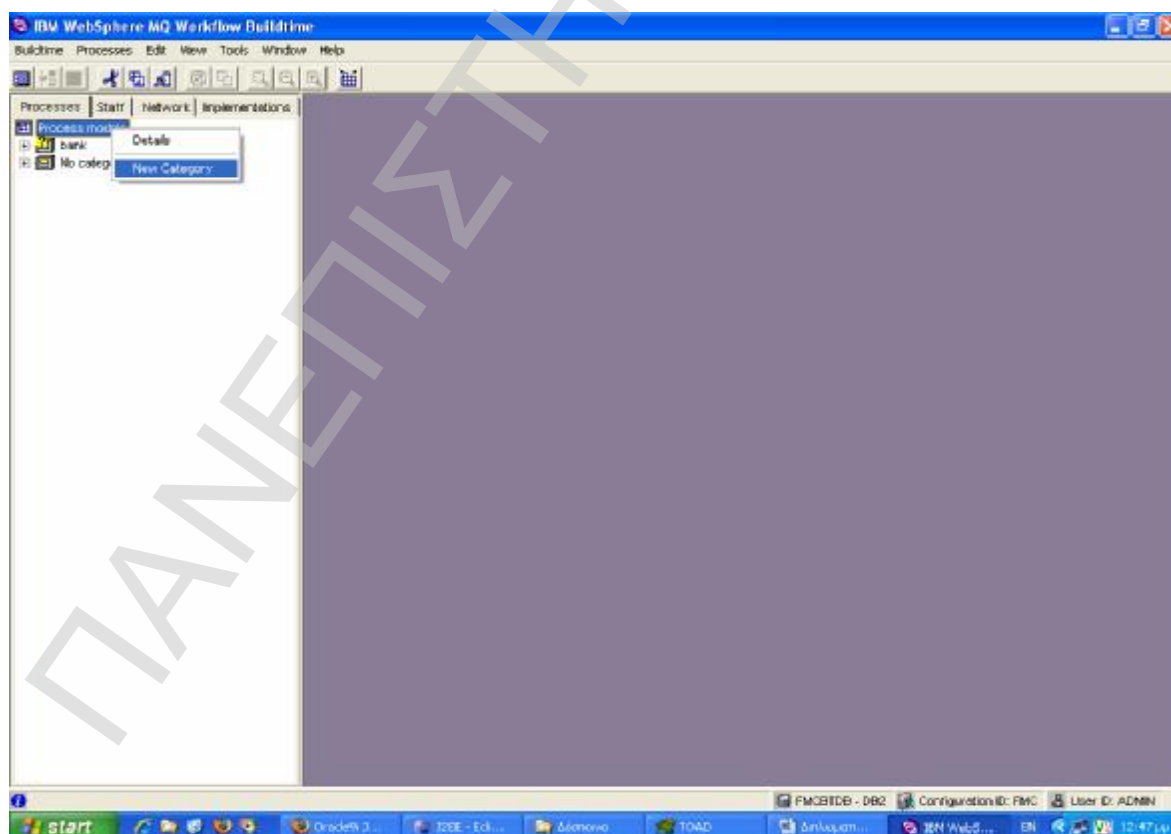
**Username= ADMIN και password =password.** Μας προτρέπει το εργαλείο να τα αλλάξουμε για λόγους ασφαλείας, αλλά για την συγκεκριμένη εργασία θα παραμείνουν ως έχουν.

Οι κύριες κατηγορίες αντικειμένων που θα πρέπει να οριστούν, όπως φαίνεται και στην [εικόνα 3-16](#), είναι τα εξής:

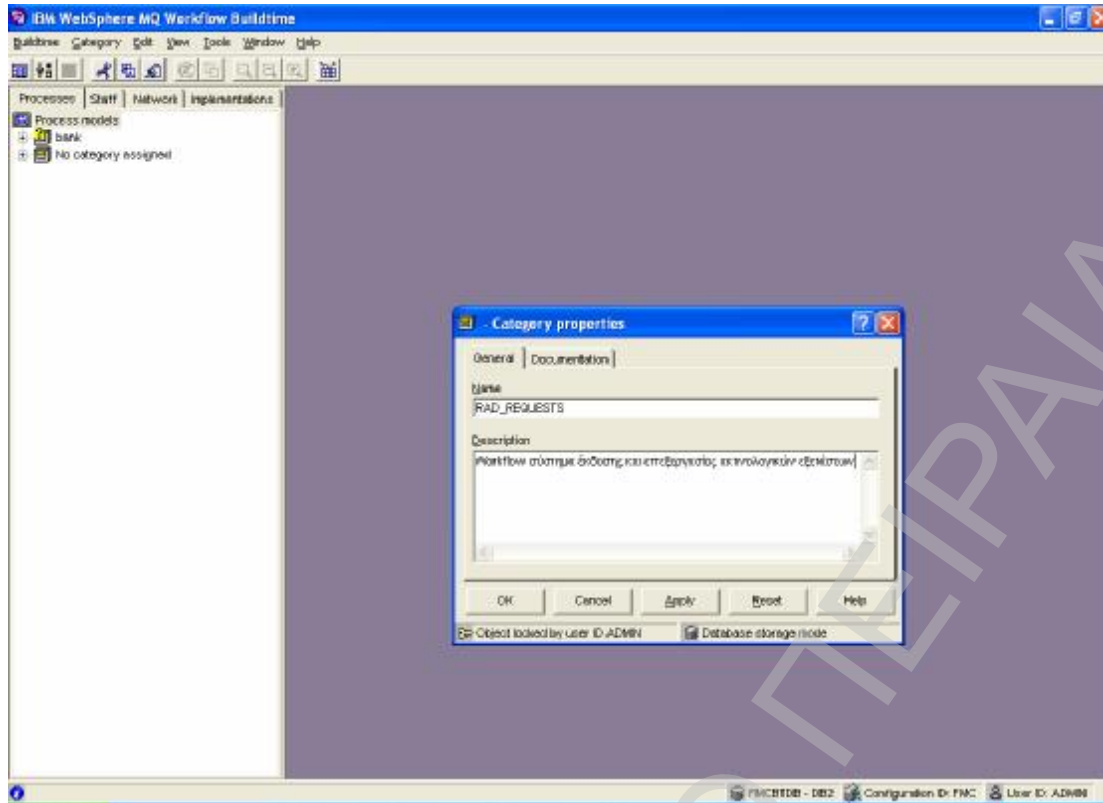
- 1) PROCESSES
- 2) STAFF
- 3) NETWORK
- 4) IMPLEMENTATIONS

#### 3.3.1 Διαδικασίες ( PROCESSES)

Στο σημείο αυτό ξεκινάει η δημιουργία του μοντέλου της διαδικασίας που θέλουμε να υλοποιήσουμε. Με απλό και γραφικό τρόπο, έχουμε την δυνατότητα να απεικονίσουμε τις δραστηριότητες της διαδικασίας και να τις διατάξουμε νοητά σε διδιάστατο χώρο με την σειρά που επιθυμούμε την εκτέλεσή τους. Αρχικά, δημιουργούμε μια νέα κατηγορία διαδικασιών που θα ονομαστεί «RAD\_REQUESTS», όπως φαίνεται και στις εικόνες [3-16](#), [3-17](#).

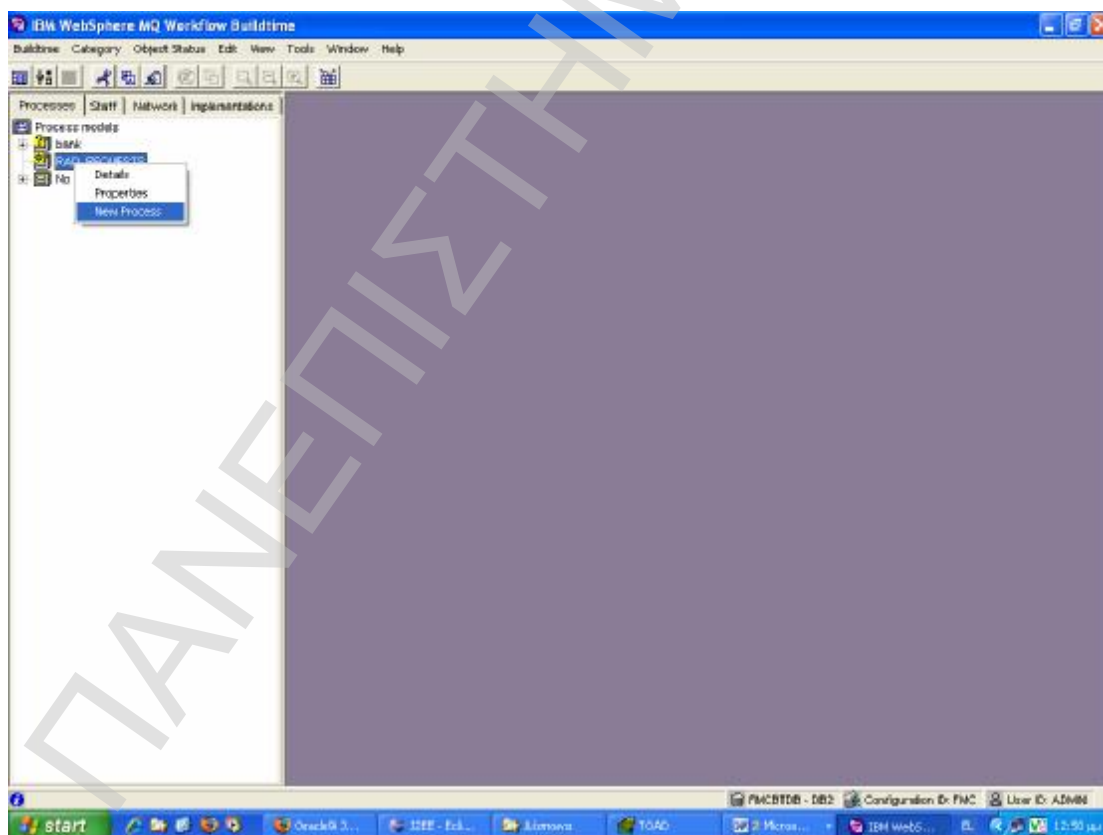


Εικόνα 3-16: New process category

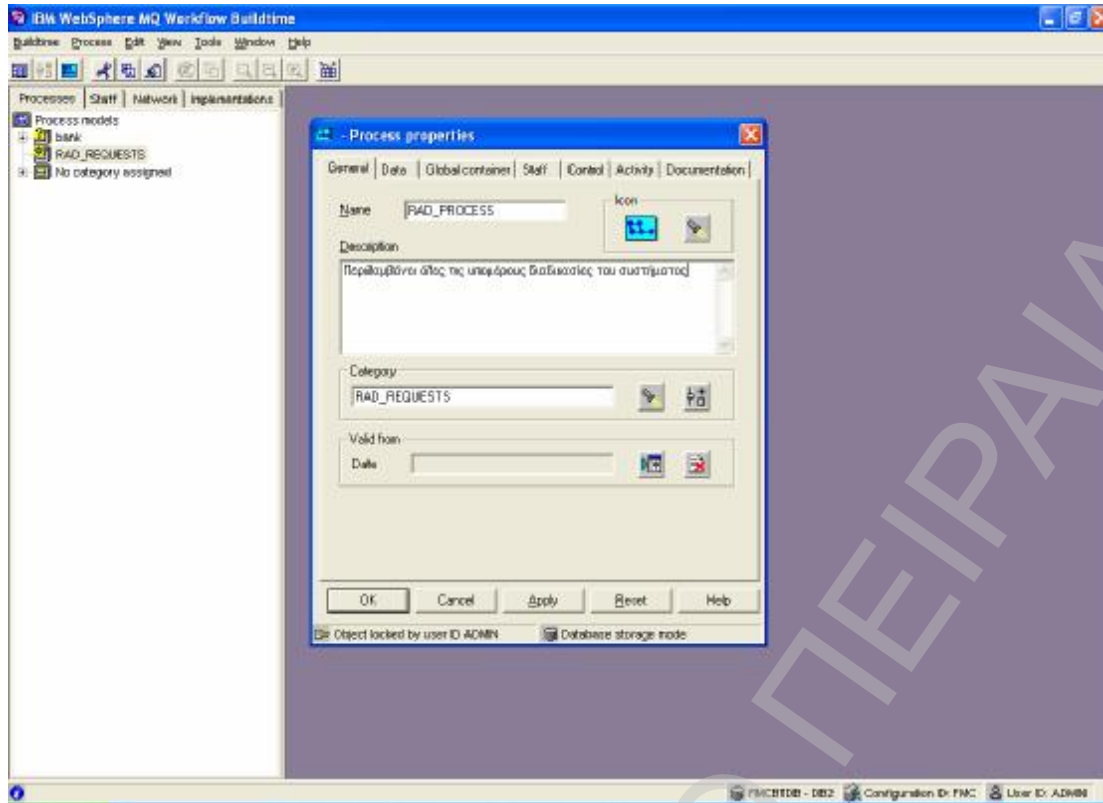


**Εικόνα 3-17:** Category properties

Εν συνεχεία, δημιουργείται μια νέα διαδικασία (process) μέσα στην κατηγορία rad\_requests, η οποία ονομάζεται «RAD\_PROCESS» (Εικόνα 3-18, Εικόνα 3-19)



**Εικόνα 3-18:** New Process



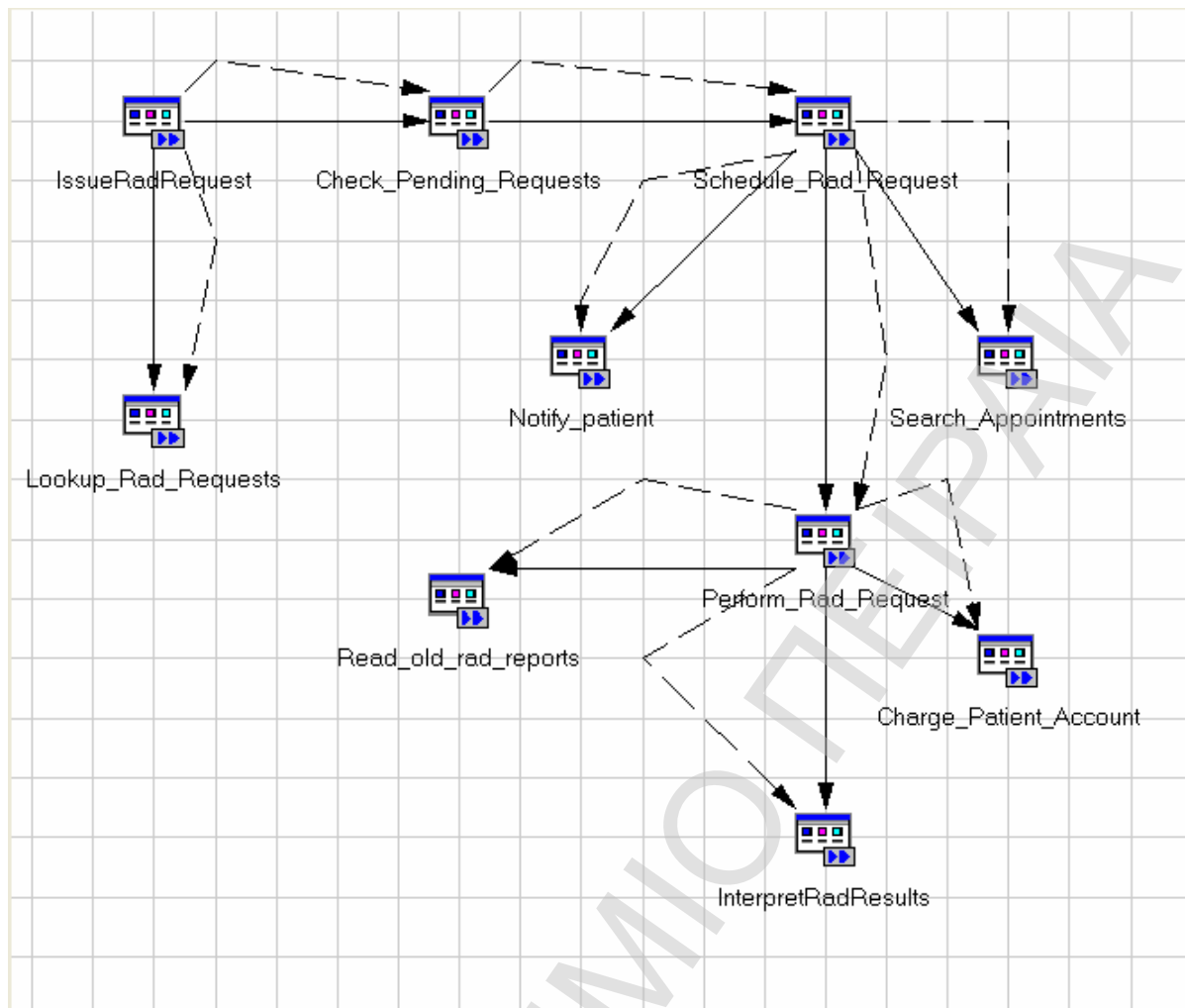
Εικόνα 3-19: New process properties

### 3.3.1.1 Μοντέλο διαδικασίας

Στην ενότητα αυτή, θα δημιουργηθεί το μοντέλο της διαδικασίας που θα υλοποιηθεί και ονομάζεται «RAD\_PROCESS».

Οι έννοιες των δραστηριοτήτων και η ροή τους, περιγράφηκε αναλυτικά στο κεφάλαιο 1.3. Σε αυτή την ενότητα, θα περιγραφεί, ο τρόπος δημιουργίας τους στο buildtime.

Αρχικά, δημιουργείται το διάγραμμα των δραστηριοτήτων, το οποίο αποτελείται από program activities, όπως φαίνεται και στο σχήμα που ακολουθεί. [Εικόνα 3-20](#).



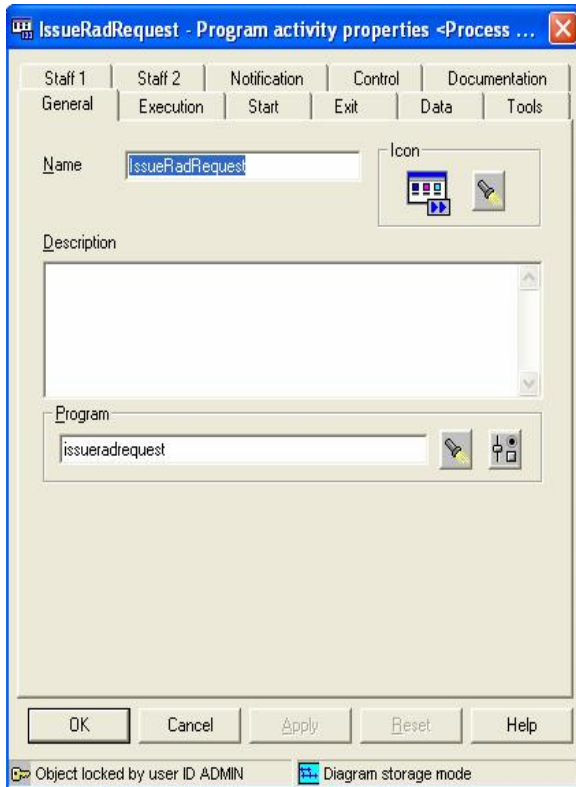
**Εικόνα 3-20:** Σύστημα ροής εργασίας έκδοσης και εκτέλεσης ακτινολογικής εξέτασης

Κάθε εικονίδιο  είναι ένα Program Activity.

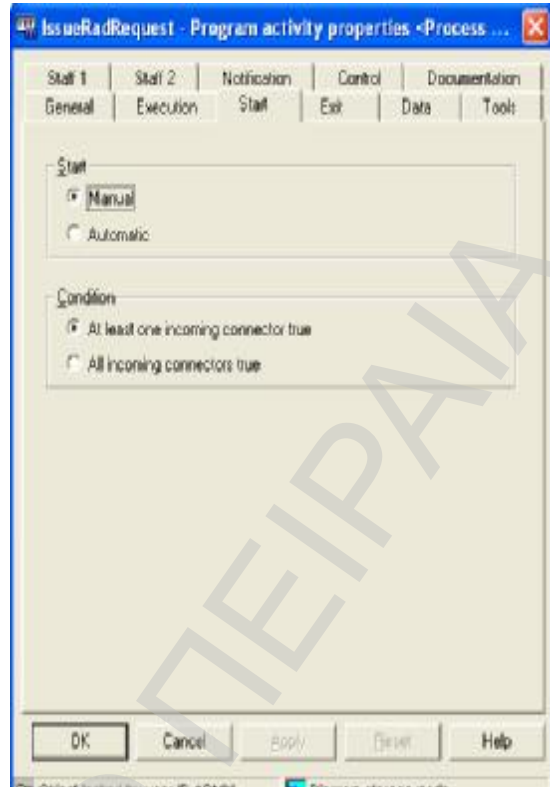
Τοποθετούμε με την μέθοδο drag and drop κάθε activity στον καμβά των δραστηριοτήτων στην δεξιά μεριά της οθόνης και σε κάθε activity ορίζεται στα properties, το όνομά του και το πρόγραμμα το οποίο θα καλείται κατά την κλήση του από τον client. βλ. [Εικόνα 3-21](#)

Ενώνουμε τις διαδικασίες με τους λεγόμενους data connectors (διακεκομμένα βελάκια), ώστε να γνωρίζει το σύστημα αν θα μεταδίδονται δεδομένα και προς τα πού. Τα μη διακεκομμένα βελάκια μεταξύ των δραστηριοτήτων υποδεικνύουν την σειρά εκτέλεσης των δραστηριοτήτων.

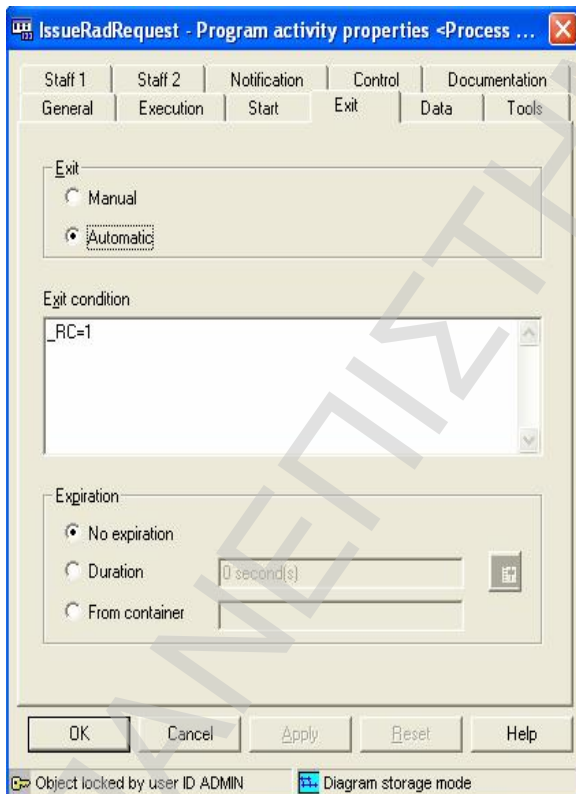
Στην [εικόνα 3-22](#), ορίζεται ο τρόπος με τον οποίο θα ξεκινάει η κλήση του προγράμματος, αυτόματα ή όχι και στην [εικόνα 3-23](#), ορίζεται ο τρόπος με τον οποίο θα τερματίζεται η λειτουργία του. Βλέπουμε ότι έχει οριστεί ειδική συνθήκη ( $\_RC=1$ ), όπου όταν θα ικανοποιηθεί η συνθήκη αυτή μέσα στο πρόγραμμα, η διεργασία θα τερματίσει.



Εικόνα 3-21: Program Activity Properties-General



Εικόνα 3-22: Program Activity Properties Start

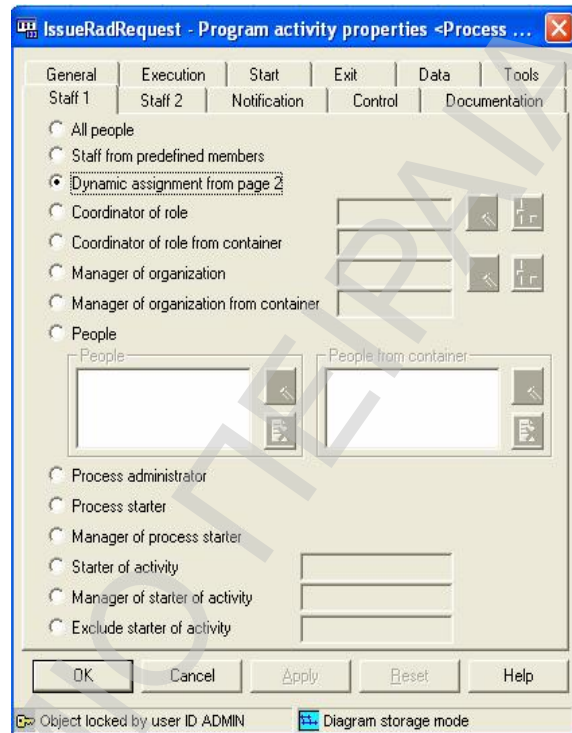
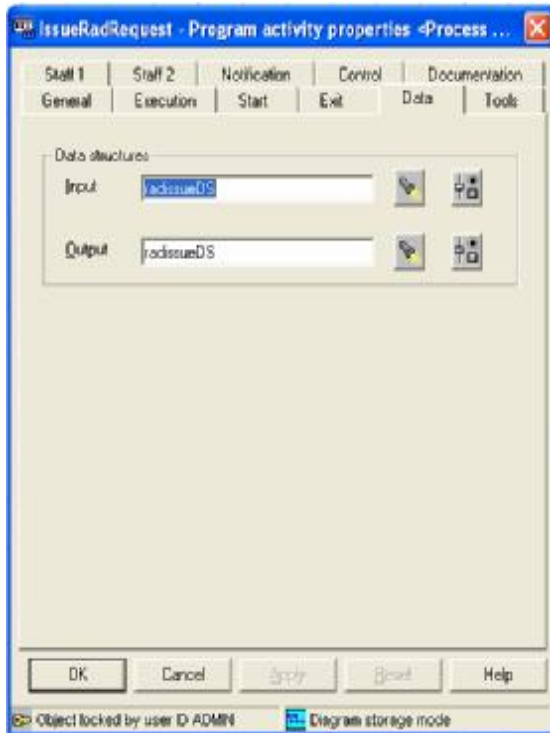


Εικόνα 3-23: Program Activity Properties-Exit



Εν συνεχεία, ορίζονται οι δομές των δεδομένων τα οποία θα λαμβάνει και αντίστοιχα θα δίνει κατά την έναρξη και λήξη του προγράμματος.

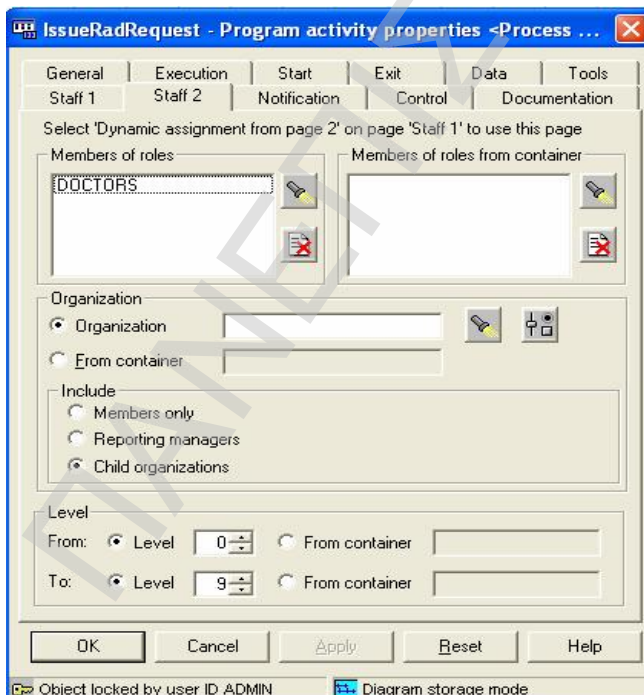
Θα περιγράψουμε παρακάτω τα λεγόμενα data structures (ενότητα 3.3.3.1) και πως δημιουργούνται, τα οποία σε αυτή την ενότητα μπορούμε να επιλέξουμε ως input και output δεδομένα του προγράμματος. Εικόνα 3-24.



Εικόνα 3-24: Program Activity Properties-Data Εικόνα 3-25: Program Activity Properties-Staff1

Τέλος, ορίζουμε τα άτομα που θα έχουν πρόσβαση στην εκτέλεση του προγράμματος, ή τους ρόλους χρηστών που θα έχουν πρόσβαση. Στην εικόνα 3-25, φαίνεται ότι επιλέγεται η επιλογή του Staff2 όπως φαίνεται στην Εικόνα 3-26. π.χ. για την διεργασία IssueRadRequest καθορίζεται ως αρμόδιος εκτελεστής της ο ρόλος DOCTORS.

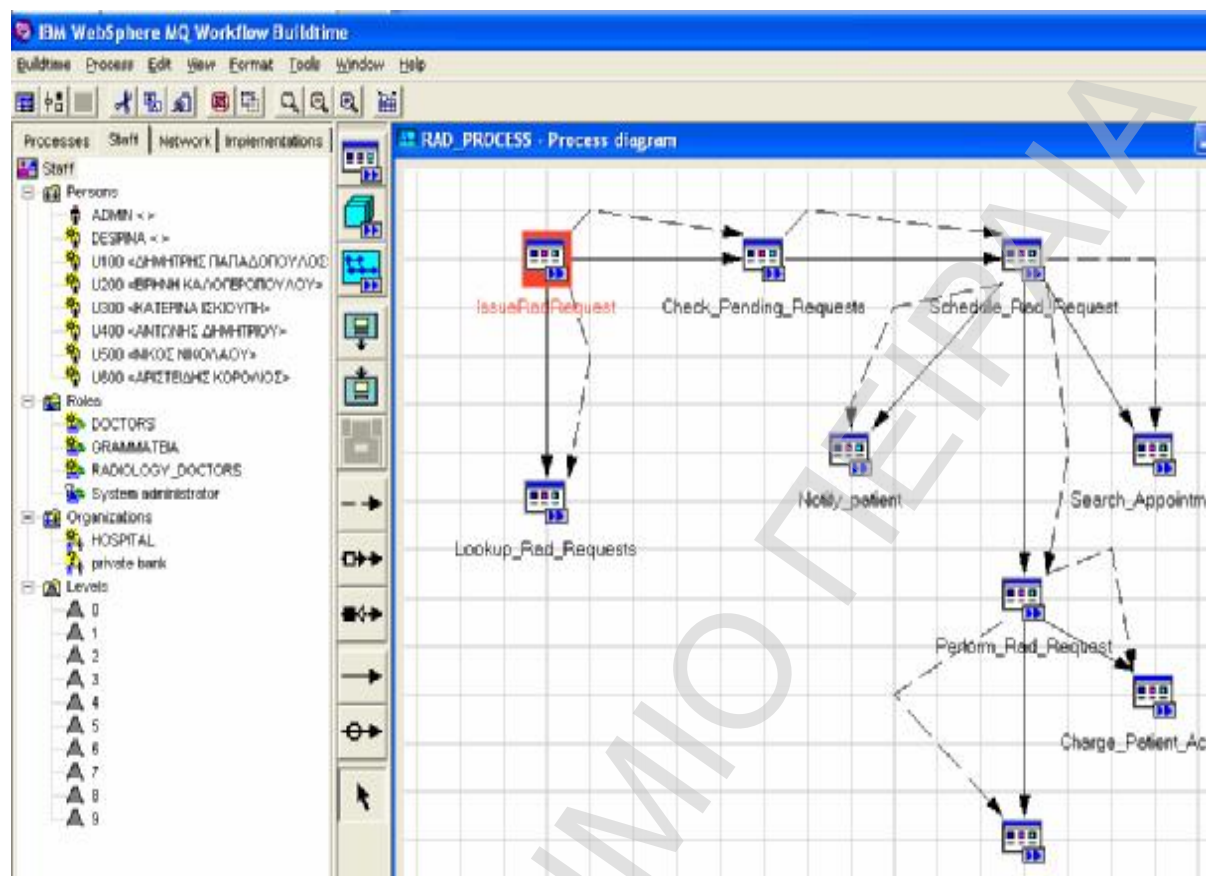
Ανάλυση του τρόπου δημιουργίας χρηστών και ρόλων θα γίνει στην ενότητα 3.3.2.



Εικόνα 3-26: Program Activity Properties-Staff2

### 3.3.2 Χρήστες και ρόλοι συστήματος (STAFF)

Στην ενότητα αυτή, θα περιγραφούν οι διαδικασίες δημιουργίας των χρηστών – ρόλων του συστήματος.



Εικόνα 3-27: STAFF elements

Στην εικόνα 3-27, διακρίνουμε 4 υπό -κατηγορίες για την κατηγορία staff.

- 1) Persons
- 2) Roles
- 3) Organizations
- 4) Levels

Αρχικά ξεκινάμε να δημιουργήσουμε έναν οργανισμό (Organization), τον οποίο τον ονομάζουμε HOSPITAL.

Σε αυτόν τον οργανισμό θα δημιουργήσουμε 3 ρόλους:

- DOCTORS
- GRAMMATEIA
- RADIOLOGY\_DOCTORS

Στην πρώτη κατηγορία, Persons, δημιουργούμε τα άτομα- χρήστες που θα μπορούν κατά το runtime να εκτελούν τις διάφορες διαδικασίες που τους έχουν ανατεθεί, είτε απευθείας είτε μέσω κάποιου ρόλου που τους έχει δοθεί.

Επομένως κάθε άτομο-person, τοποθετείτε σε έναν οργανισμό- τμήμα και του δίνεται ένας ρόλος. Εικόνα 3-28.

Για παράδειγμα, ο χρήστης U100 – Δημήτρης Παπαδόπουλος, εργάζεται στον οργανισμό HOSPITAL και έχει τον ρόλο της γραμματείας GRAMMATEIA.

Έχουμε την δυνατότητα να ορίσουμε πολλά organizations και να τα ιεραρχήσουμε με την βοήθεια των levels. Στην παρούσα εργασία, δεν θα ακολουθηθεί αυτή η τακτική, μιας και η διάρθρωση των τμημάτων της υγειονομικής περιφέρειας αναλύεται στην βάση δεδομένων του συστήματος και θα ελέγχονται από εκεί.

Επίσης, επειδή στο σύστημα αναπτύσσεται ξεχωριστή βάση δεδομένων ασφαλείας του συστήματος, αυτό που έχουμε να κάνουμε είναι να ορίσουμε μόνο τους χρήστες του συστήματος και τους ρόλους που έχουμε και στην βάση δεδομένων της ασφάλειας και ο έλεγχος θα εξασφαλίζεται από το πρόγραμμα και τους κανόνες ασφαλείας αυτής της βάσης.

The image shows a software dialog box titled "U100 - Person properties" with a blue header bar. It has four tabs: "General", "Staff" (which is selected), "Information", and "Authorizations". The "Staff" tab contains three main sections: "Member of organization", "Substitute", and "Member of roles".  
1. "Member of organization": A text field labeled "Organization name" contains the text "HOSPITAL". To the right of the field are two small icons: a magnifying glass and a person icon.  
2. "Substitute": A text field labeled "Substitute User ID" is empty. To its right are the same two icons as above. Below this field is the label "Last name:".  
3. "Member of roles": A list box contains the text "GRAMMATEIA". To the right of the list box are three icons: a magnifying glass, a person icon, and a red 'X' icon.  
At the bottom of the dialog are five buttons: "OK", "Cancel", "Apply", "Reset", and "Help".  
The status bar at the very bottom of the window contains two items: "Object locked by user ID ADMIN" and "Database storage mode".

Εικόνα 3-28: User U100 – Person Properties - Staff

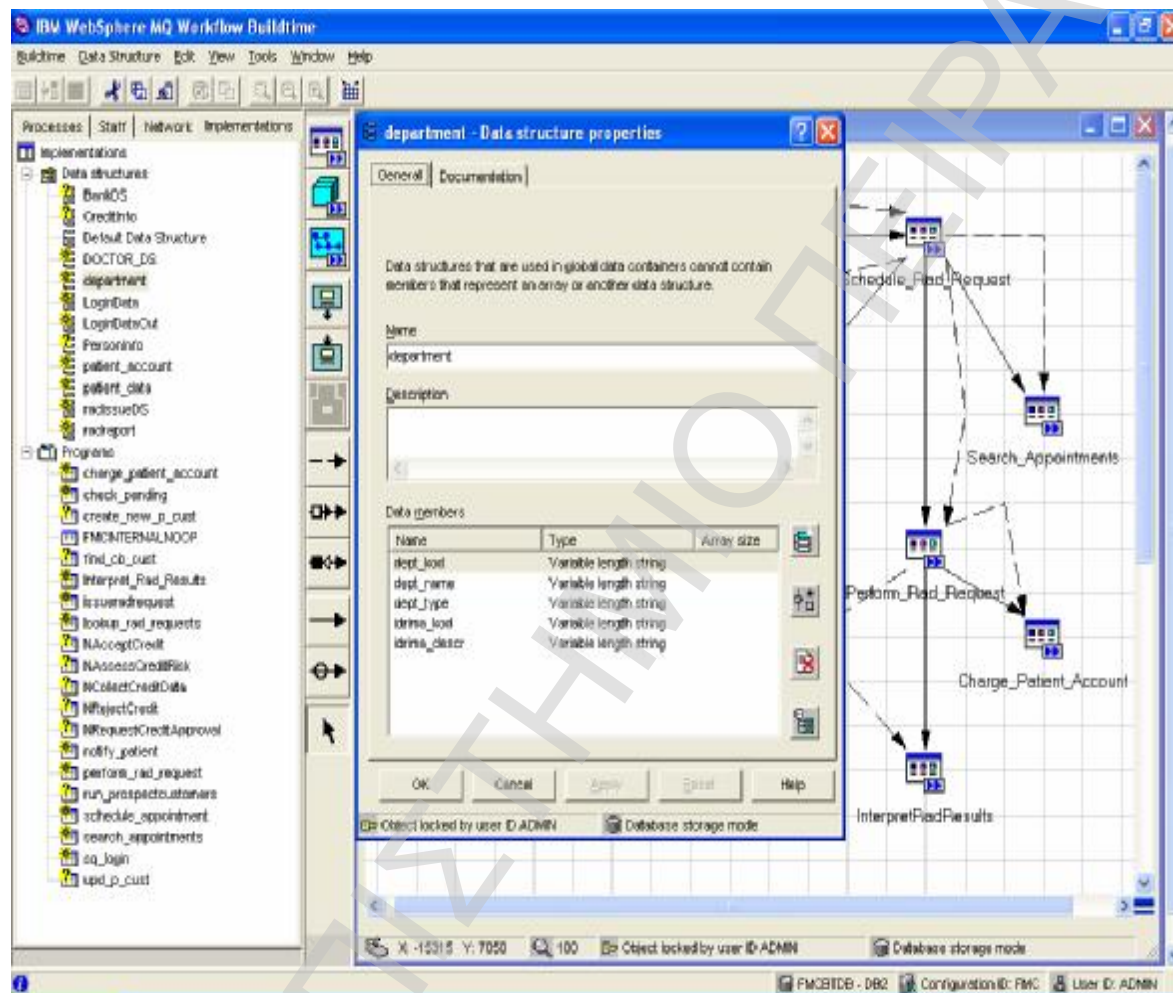
### 3.3.3 Υλοποιήσεις (Implementations)

Στην ενότητα των υλοποιήσεων, θα δούμε δύο κύριες υπό-κατηγορίες.

- 1) Data Structures
- 2) Programs

#### 3.3.3.1 Data Structures

Είναι οι δομές δεδομένων που ανταλλάσσονται μεταξύ των activity programs. Χρησιμοποιούνται ως input και output data structures των activities.



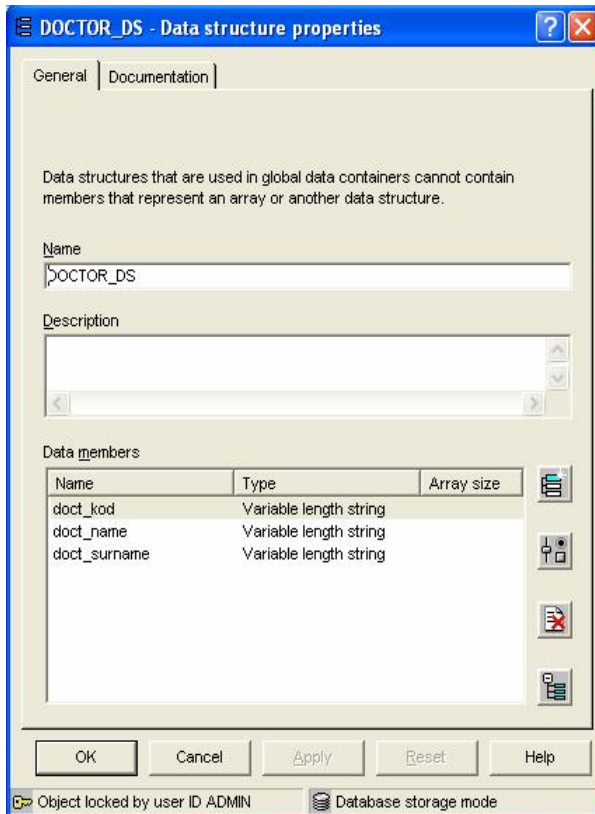
Εικόνα 3-29: Department data structure

Στην εικόνα 3-29, η δομή department, αποτελείται από 5 μεταβλητές τύπου string, οι οποίες χαρακτηρίζουν ένα τμήμα νοσοκομείου.

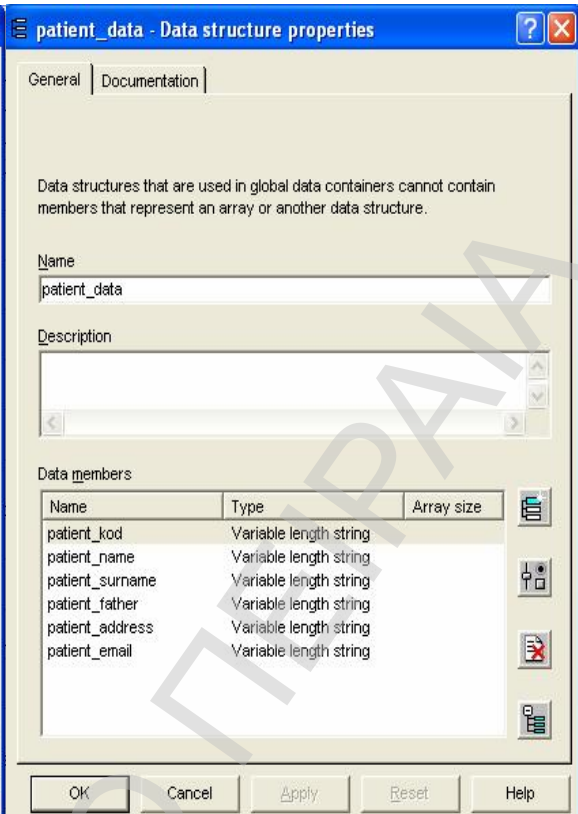
Στην [εικόνα 3-30](#), η δομή doctor\_ds, χαρακτηρίζει έναν γιατρό. (DOCT\_KOD, DOCT\_NAME, DOCT\_SURNAME)

Στην [εικόνα 3-31](#), η δομή patient\_data χαρακτηρίζει έναν ασθενή. (Patient\_kod, patient\_name, patient\_surname, patient\_father, patient\_address, patient\_email)

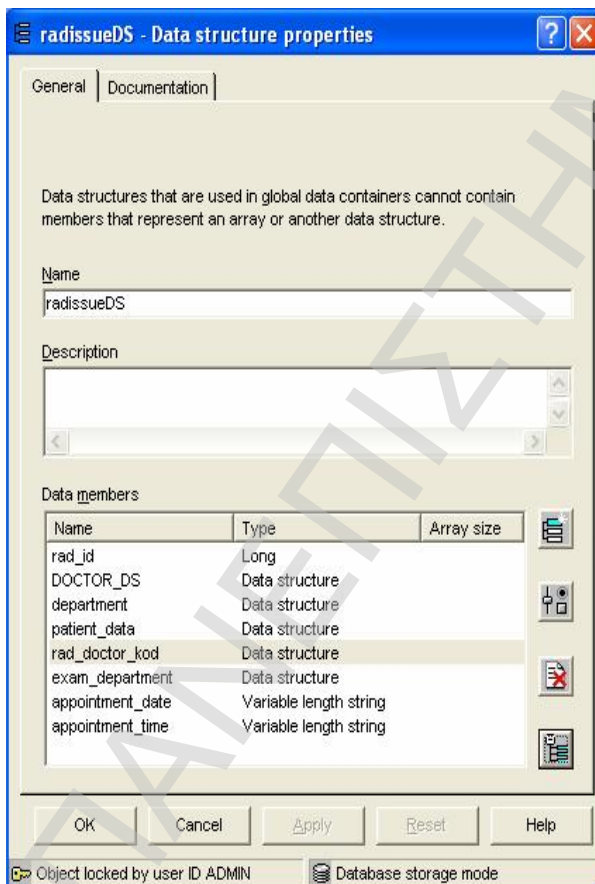




Εικόνα 3-30: DOCTOR\_DS Data Structure



Εικόνα 3-31: patient\_data Data structure



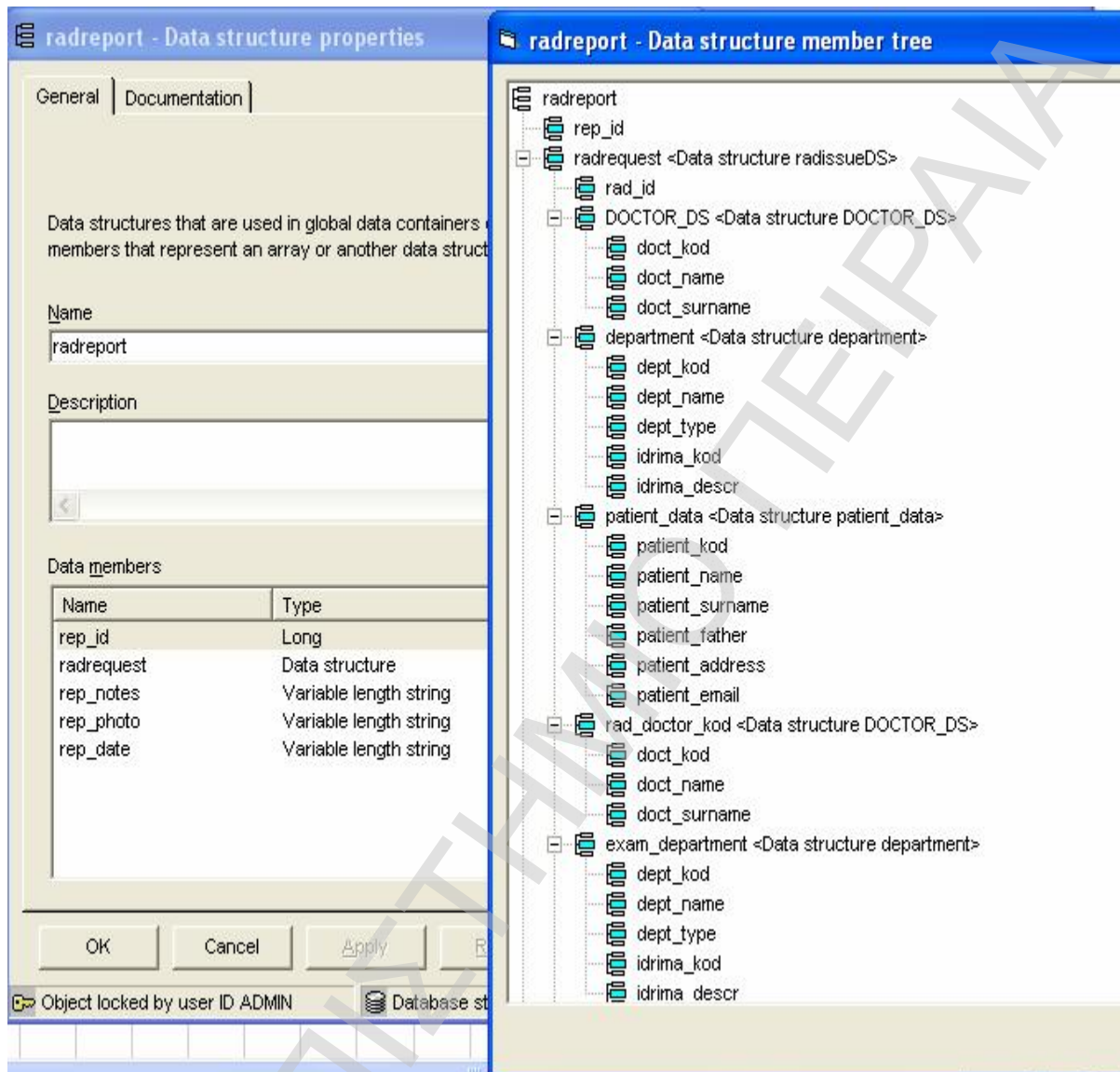
Εικόνα 3-32: radissueDS Data Structure



Εικόνα 3-33: radissueDS member tree

Τέλος, έχουμε την δυνατότητα, να αναμειξουμε διαφορετικά data structures και να δημιουργήσουμε ένα νέο data structure όπως φαίνεται στις εικόνες 3-32, 3-33.

Το συγκεκριμένο data structure (radissueDS) χρησιμοποιείται ως input και output data στην διαδικασία IssueRadRequest. Εικόνα 3-34.



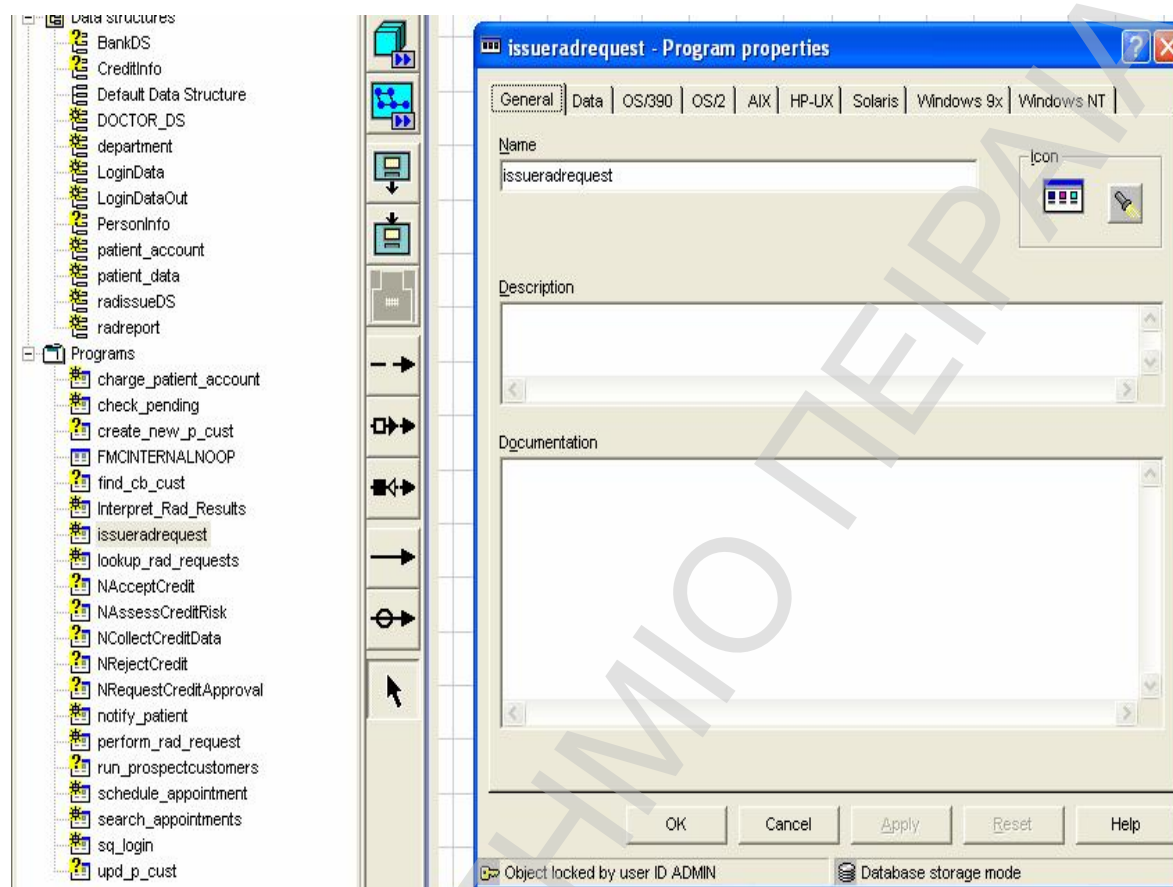
Εικόνα 3-34: Data Structure - repreport

Σημείωση :

Το εργαλείο του Workflow, υποστηρίζει του τύπους : String, Binary, Long, Float, Data Structure.

### 3.3.3.2 Προγράμματα - Programs

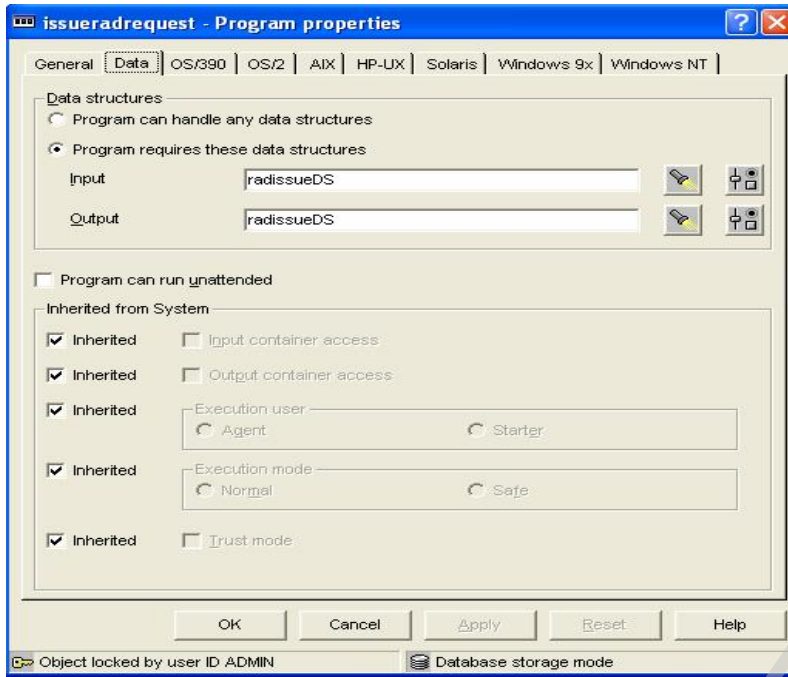
Στην ενότητα αυτή, δημιουργούμε τα προγράμματα που θα εκτελούνται από το σύστημα ροής εργασίας, ένα για κάθε program activity του process model. Όπως διαφαίνεται και στην εικόνα 3-35 , αριστερά φαίνονται όλα τα διαφορετικά προγράμματα τα οποία μπορεί να εκτελέσει το workflow engine.



Εικόνα 3-35: issueradrequest – Program properties

Αρχικά, ορίζουμε στο general tab το όνομα του προγράμματος.(issueradrequest)

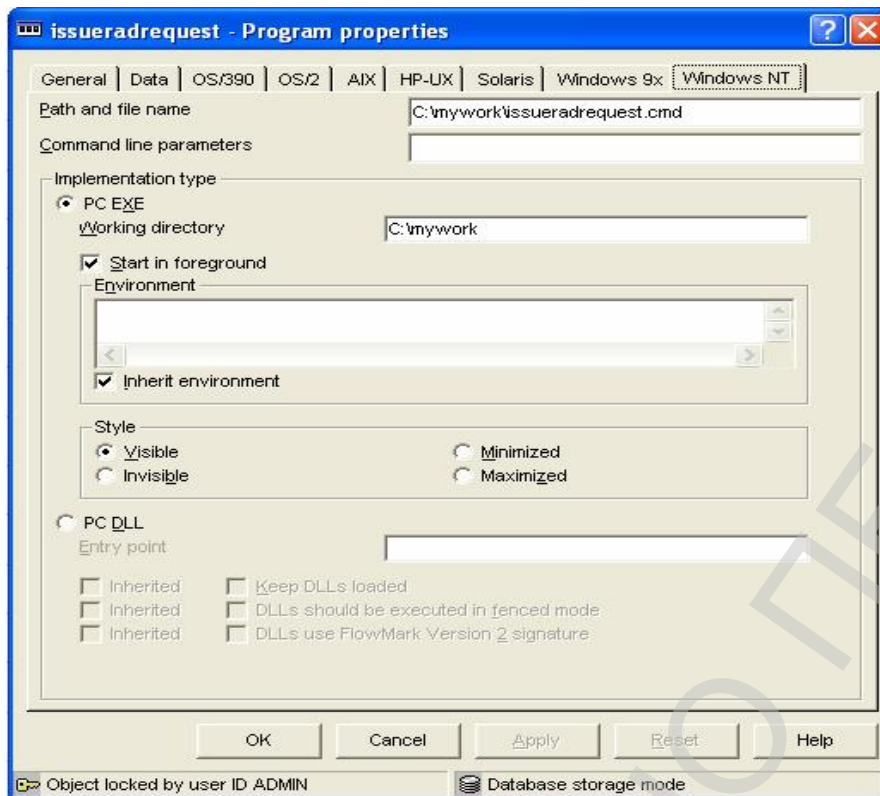
Εν συνέχεια, στο Data tab, ορίζουμε τις εισερχόμενες και εξερχόμενες δομές δεδομένων, όπως τις ορίσαμε στην προηγούμενη ενότητα. [Εικόνα 3-36](#).



Εικόνα 3-36: issueradrequest- program properties - data



Τέλος, ορίζουμε την διαδρομή στο fileserver του συστήματος ροής εργασίας, απ' όπου θα αντληθεί το εκτελέσιμο αρχείο που θα πρέπει να εκτελεστεί κατά το runtime. Εικόνα 3-37.



Εικόνα 3-37: issueradrequest-Program properties- WindowsNT

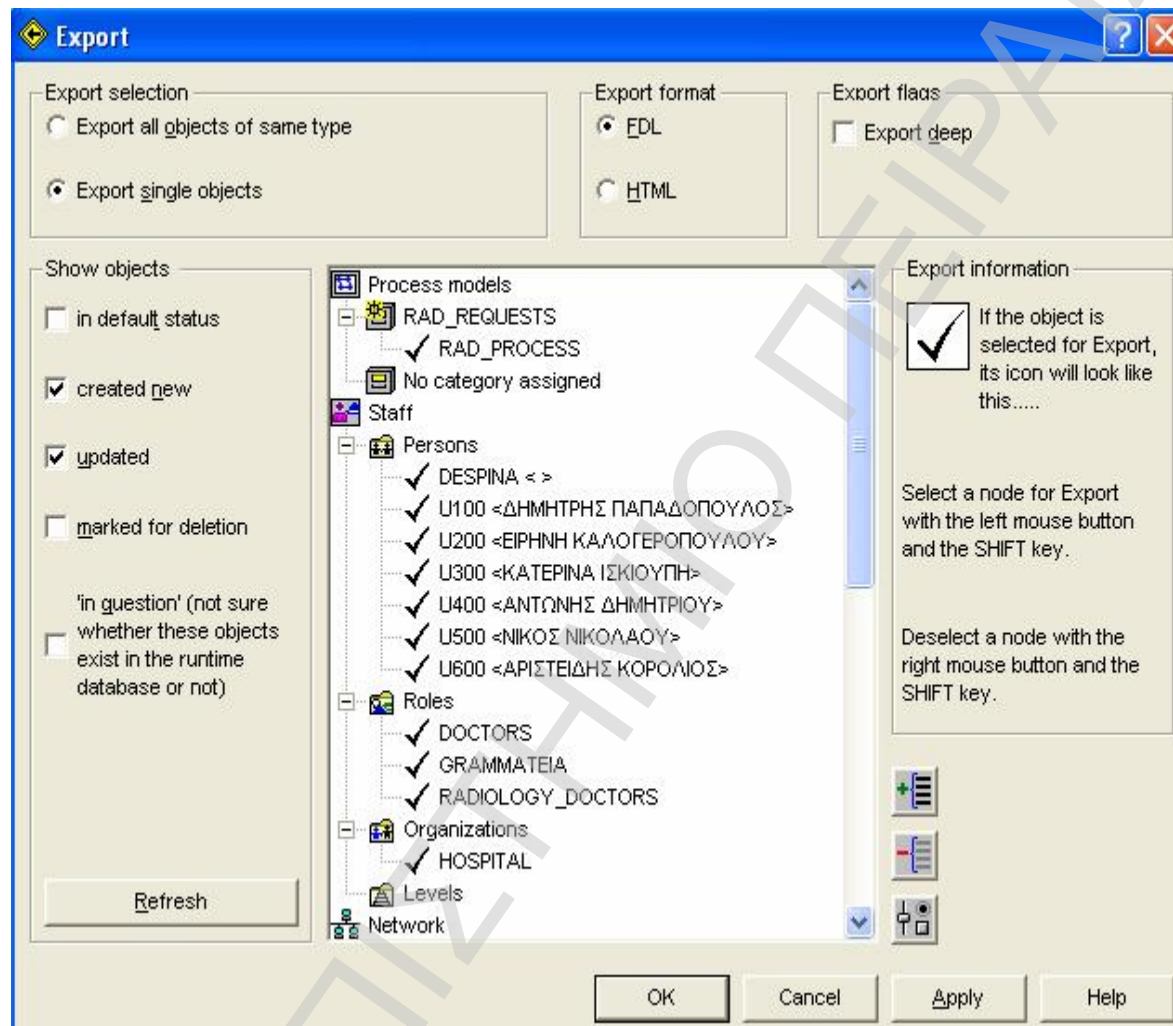
Το cmd πρόγραμμα που αναφέρεται στην παραπάνω εικόνα 3-37, περιέχει τον εξής κώδικα εκτέλεσης:

```
@ECHO OFF
C:
CD \mywork
"C:\Program Files\Java\jdk1.5.0_18\bin\java.exe" -classpath
lib\fmcojagt.jar;lib\fmcojapi.jar;lib\jdev-rt.jar;lib\classes12.jar;lib\nls-charset11.jar;.
project1.IssueRadRequest
PAUSE
```

Εδώ διαφαίνεται ότι εκτελείται ένα java class IssueRadRequest.

### 3.4 EXPORT

Μετά την ολοκλήρωση των παραπάνω ρυθμίσεων και τον σχεδιασμό του μοντέλου, πρέπει να δημιουργηθεί ένα xml αρχείο με κατάληξη fdl, το οποίο θα περιέχει όλες τις ρυθμίσεις του μοντέλου και με βάση αυτό θα γίνει η σύνδεση με τον Client του Workflow. Η διαδικασία αυτή ονομάζεται Export και βρίσκεται στο Menu Bar στην επιλογή Buildtime ( Buildtime – Export ). Στην οθόνη που εμφανίζεται θα πρέπει να γίνει η επιλογή των ρυθμίσεων για κάθε μοντέλο ξεχωριστά ενεργοποιώντας την επιλογή Export Single Objects

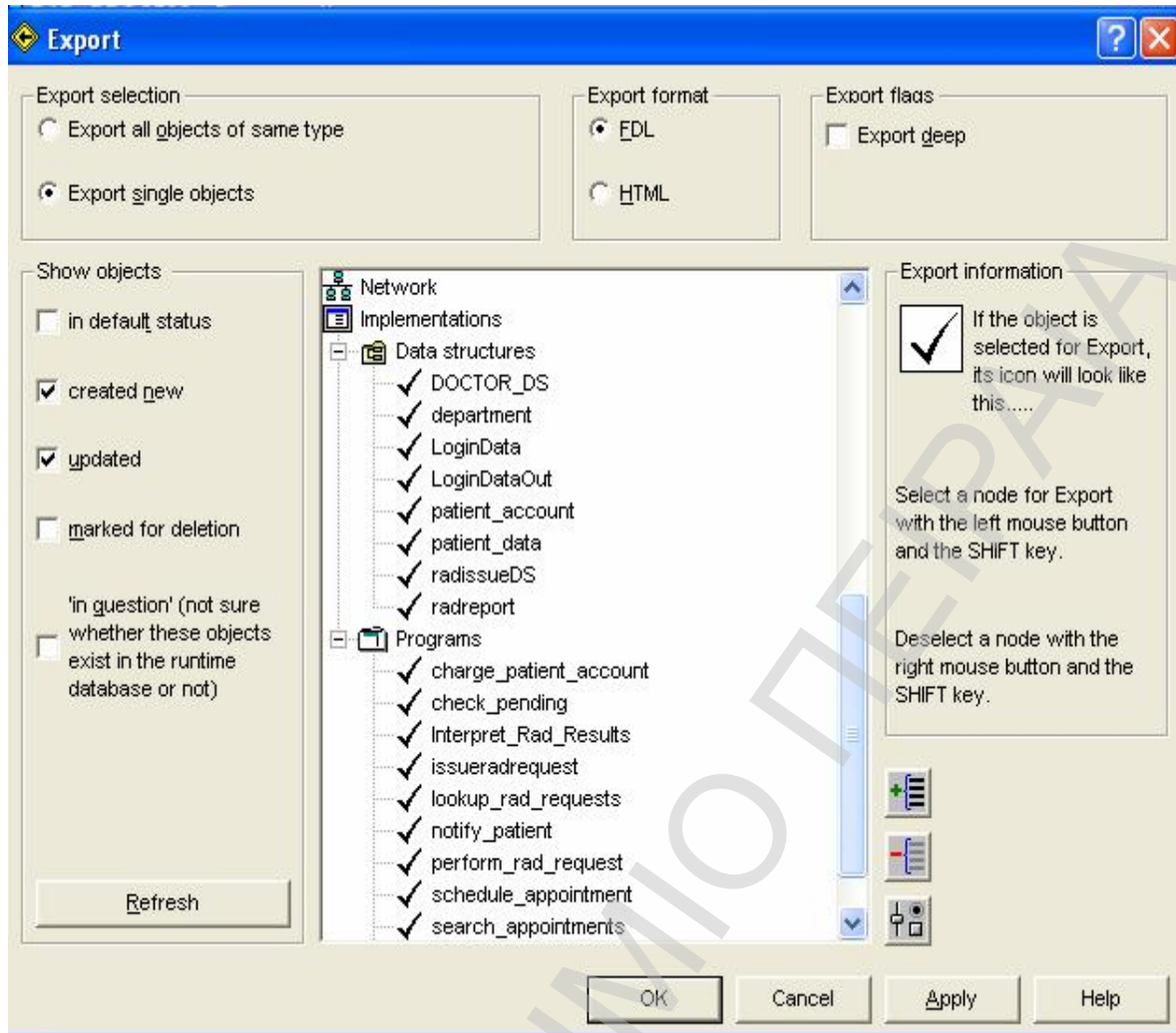


Εικόνα 3-38: Export (1)

Επιλέγουμε όλα τα components του buildtime model όπως αυτά διαρθρώνονται στην εικόνα 3-38, και 3-39.

Η επιλογή των στοιχείων, γίνεται με το Shift (κουμπί) και με αριστερό κλικ από το ποντίκι. Όταν έχουν επιλεγεί όλα τα στοιχεία, με το κουμπί Ok δημιουργείται το fdl αρχείο. Το συγκεκριμένο αρχείο, απαιτείται να βρίσκεται στον φάκελο FDL του Workflow, γιατί μέσα από αυτόν θα γίνει το Import και η είσοδος στον Client.

Το αρχείο fdl που έχει δημιουργηθεί βρίσκεται στο παράρτημα 3.



Εικόνα 3-39: Export (2)

### 3.5 Websphere MQ Workflow Client

Ο Client είναι το μέσο με το οποίο ο απλός χρήστης θα μπορεί να τρέχει την εφαρμογή. Για να μπορέσει ο Client να λειτουργήσει θα πρέπει το fdl αρχείο που έγινε Export από το Buildtime, να γίνει Import στην runtime database.

Τα βήματα είναι τα εξής :

Βήμα 1ο: Άνοιγμα του Ms-Dos Prompt

Βήμα 2ο: Μετάβαση στον φάκελο FDL του Workflow => CD C:\Program Files\IBM\WebSphere MQ\Workflow\FDL

Βήμα 3ο: Εκτέλεση της εντολής : fmcibie -u ADMIN -p password -i rad.fdl -o -t

Με την ολοκλήρωση της παραπάνω εργασίας ο Client μπορεί να τεθεί σε λειτουργία. Στον Client, πρέπει να ορισθούν τα Worklists, Templates, Instances, για κάθε άτομο που έχει ορισθεί στο μοντέλο. Κάθε άτομο του μοντέλου μπορεί να δημιουργεί τα δικά του Worklists, Templates και Instances. ο προγραμματιστής μπορεί να κάνει εισαγωγή στο σύστημα με την ιδιότητα του administrator, να δημιουργήσει μια δικιά του Worklist με την οποία θα μπορεί να παρακολουθεί ποιες δραστηριότητες έχουν πραγματοποιηθεί από ποιον χρήστη.

## 4<sup>ο</sup> ΚΕΦΑΛΑΙΟ

### ΔΗΜΙΟΥΡΓΙΑ ΠΡΟΓΡΑΜΜΑΤΩΝ ΣΕ ΓΛΩΣΣΑ JAVA

#### Java Programs

## 4 ΠΑΡΟΥΣΙΑΣΗ – ΑΝΑΛΥΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ

### 4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα επισημανθούν και αναλυθούν τα κύρια σημεία των προγραμμάτων που έχουν δημιουργηθεί σε java και που καλούνται σε κάθε δραστηριότητα της διαδικασίας έκδοσης και εκτέλεσης μιας ακτινολογικής εξέτασης σε μια υγειονομική περιφέρεια, όπως αυτή αναλύθηκε σε προηγούμενο κεφάλαιο. Η έκδοση της java που έχει χρησιμοποιηθεί είναι η 1.5.0\_18, ενώ εργαλείο ανάπτυξης των προγραμμάτων επιλέχθηκε ο Oracle JDeveloper 10.1.3.2.0.

### 4.2 Δομή προγραμμάτων – κρίσιμα σημεία ενδιαφέροντος

Όλα τα προγράμματα java που έχουν δημιουργηθεί για την παρούσα εφαρμογή, ακολουθούν συγκεκριμένη δομή. Λόγω της κλήσης τους μέσα από τον workflow engine, θα πρέπει να επιτυγχάνουν σύνδεση τόσο με την βάση δεδομένων του workflow όπου καταχωρούνται οι απαραίτητες πληροφορίες που δημιουργήθηκαν για κάθε δραστηριότητα της διαδικασίας στο προηγούμενο κεφάλαιο, όσο και με την βάση δεδομένων του περιφερειακού νοσοκομείου για τα δεδομένα πλέον των εμπλεκόμενων φορέων της εφαρμογής. Παράλληλα, κάθε πρόγραμμα θα πρέπει να ακολουθεί τους κανόνες ασφαλείας και πρόσβασης που έχουν καθοριστεί από την εφαρμογή. Τέλος, θα πρέπει να εξασφαλίζεται η ορθή μεταφορά δεδομένων μεταξύ των δραστηριοτήτων του συστήματος, ανάλογα με τις δομές δεδομένων που έχουν καθοριστεί στην βάση του workflow. Θα αναλυθούν στην συνέχεια αυτά τα σημεία ενδιαφέροντος, ώστε να γίνει κατανοητή η υλοποίηση του κάθε προγράμματος.

#### 4.2.1 Σύνδεση με την βάση δεδομένων

Προκειμένου να αντλήσουμε ή να καταχωρήσουμε δεδομένα στην βάση δεδομένων του περιφερειακού νοσοκομείου, θα πρέπει να αποκτήσουμε πρόσβαση στην βάση δεδομένων της Oracle database (HOSPITAL). Η σύνδεση με την βάση δεδομένων επιτυγχάνεται μέσω jdbc driver της Oracle, όπως φαίνεται στην παρακάτω μέθοδο, που υπάρχει σε κάθε κλάση.

```
public static void createConnection()
{
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        conn =
        DriverManager.getConnection("jdbc:oracle:thin:@despina:1521:ORCL","hospital","
        hospital");
    }
    catch (ClassNotFoundException cnfe) {
        System.out.println("class not found "+cnfe);}
    catch (SQLException sqle) {System.out.println("sql exception "+sqle);}
}
```

Αφού λοιπόν γίνει κλήση της παραπάνω μεθόδου και επιτευχτεί η σύνδεση με την βάση HOSPITAL, με χρήση της κλάσης PreparedStatement δημιουργούμε sql statements, είτε για εύρεση δεδομένων (select) , είτε για εισαγωγή δεδομένων (insert). Π.χ.

```

try{
    PreparedStatement pStmt = conn.prepareStatement("insert into RAD_REQUESTS
(RAD_ID,PAt_KOD,DOCT_KOD,DEPT_KOD) values (?,?=?,find_aktinologiko("+
DOCT_DEPT_KOD.toString()+"))");
    pStmt.setInt(1,a);
    pStmt.setString(2, jComboBox1.getSelectedItem().toString());
    pStmt.setString(3, userid.toString());
    pStmt.executeUpdate();
    pStmt.close();
    JOptionPane.showMessageDialog(this, "Η εγγραφή σας καταχωρήθηκε
επιτυχώς.");
    conn.close();
}
catch(SQLException sqle)
{System.out.println("Error in insert statement "+sqle.getLocalizedMessage());
System.out.println("Error in insert statement "+sqle.getErrorCode());}

```

Η για εύρεση δεδομένων:

```

    PreparedStatement pStmt1 = conn.prepareStatement("select rad_seq.nextval as
rad_id from dual");
    ResultSet rset3 = pStmt1.executeQuery();
    while(rset3.next()){
        a= rset3.getInt("rad_id");
    }
    pStmt1.close();
}
catch(SQLException sqle)
{System.out.println("Error in select sequence statement "+sqle);}

```

Με την μέθοδο του select από την βάση δεδομένων, στην Oracle, έχουμε την δυνατότητα να δημιουργήσουμε μια μέθοδο (function), στην βάση δεδομένων με plsql κώδικα και να γίνει κλήση της μέσα από τον java κώδικα. Αυτό φαίνεται στο παρακάτω σημείο κώδικα:

```

try{
    PreparedStatement pst = conn.prepareStatement("select
find_aktinologiko("+ DOCT_DEPT_KOD.toString()+") as aktinologiko from
dual");
    ResultSet rset = pst.executeQuery();
    while(rset.next()){
        EXAM_DEPT_KOD =rset.getString("aktinologiko");
    }
    System.out.println(EXAM_DEPT_KOD);
}
catch(SQLException sqle)
{System.out.println("Error in select aktinologiko statement "+sqle);}
}

```

Αυτό που φαίνεται, είναι ότι έχει δημιουργηθεί μια function με όνομα `find_aktinologiko` στην βάση δεδομένων HOSPITAL με plsql (παράρτημα 2) δίνοντας ως παράμετρο έναν κωδικό τμήματος νοσοκομείου, η οποία επιστρέφει έναν κωδικό τμήματος το οποίο θα είναι το ακτινολογικό όπου θα γίνει η ακτινολογική εξέταση. Για να γίνει η κλήση της μέσα από την java, γίνεται select από τον dual ψευδό πίνακα της oracle και το αποτέλεσμα του select επιστρέφεται στην παράμετρο EXAM\_DEPT\_KOD.

#### 4.2.2 Κλήση του workflow engine

Προκειμένου να γίνει σύνδεση με το workflow engine και κατ' επέκταση με την βάση της DB2 του workflow, χρησιμοποιείται το `api com.ibm.workflow.api.*` όπου

υπάρχουν όλες οι απαραίτητες java κλάσεις που θα διαχειριστούν την σύνδεση με το workflow engine. Χρησιμοποιείται η κλάση ExecutionAgent και οι έννοιες των containers για τα δεδομένα που ανταλλάσσονται.

Σε όλα τα προγράμματα χρησιμοποιούνται και ορίζονται οι μεταβλητές:

```
ExecutionAgent PEA;  
ReadOnlyContainer input;  
ReadWriteContainer output;
```

```
// Ορισμός του input container  
input = PEA.inContainer();  
// Ορισμός του output container  
output = PEA.outContainer();  
// Καθορισμός της predefined μεταβλητής "_RC" του output container  
output.setLong("_RC", 0);
```

Η τελευταία μεταβλητή output.setLong("\_RC", 0) ορίζεται στο ξεκίνημα της δραστηριότητας με τιμή 0 και θα αλλάξει σε όποιο σημείο κρίνεται απαραίτητο με 1, ώστε το πρόγραμμα να τερματίσει στο workflow αφού έτσι έχει καθοριστεί κατά τον σχεδιασμό των δραστηριοτήτων ως exit μεταβλητή βλ. [εικόνα 3-23](#).

### 4.2.3 Έλεγχος ορθής και επιτρεπτής πρόσβασης

Σε όλα τα προγράμματα java, κατά την εκκίνησή τους, ελέγχεται η επιτρεπτή πρόσβαση σε αυτά αναλόγως τον χρήστη που το καλεί κάθε φορά.

Ο χρήστης που το καλεί βρίσκεται μέσω της κλάσης PEA.userID(); Όπου PEA είναι το instantiation της κλάσης ExecutionAgent. Έτσι εκτελείται το εξής στην αρχή κάθε κλάσης:

```
try {  
    if (this.check_access())  
    {  
        jblnit();  
    }  
    else  
    {  
        try {  
            output.setLong("_RC", -1);  
            PEA.setOutContainer(output);  
        }  
        catch(FmcException fmc){}  
    }  
}  
catch (Exception e) {  
    e.printStackTrace();  
}
```

Βλέπουμε λοιπόν ότι, εκτελείται μια μέθοδος check\_access(), προκειμένου να αποφανθεί αν θα συνεχίσει στο «άνοιγμα» της φόρμας, ή αλλιώς να στείλει τιμή -1 στην μεταβλητή \_RC ώστε να τερματίσει η δραστηριότητα στο workflow.

Η μέθοδος check\_access είναι η εξής:

```
public boolean check_access()  
{  
    //Σύνδεση με την βάση δεδομένων HOSPITAL  
    this.createConnection();  
  
    try  
    {  
        Statement stmt = conn.createStatement();
```



```

ResultSet rset1 = stmt.executeQuery("select sq_hospital.sql.check_access
("+userid.toString()+", 'ISSUE_RAD_REQUEST', 'I') as iparxei from dual");
if(rset1 != null)
{
    while(rset1.next()){
        if (rset1.getInt("iparxei")==0)
        {
            JFrame j=new warning(this);
            j.setLocation(100, 100);
            this.setEnabled(false);
            j.setVisible(true);
            return false;
        }
        else{
            return true;
        }
    }
    return false;
}
return false;
}
catch(SQLException e1)
{
    System.out.println("SQLException : " + e1.getMessage());
    System.out.println("SQLException : " + e1.getErrorCode());
    System.out.println("SQLException : " + e1.getSQLState());
    System.out.println("SQLException1 : " + e1.getLocalizedMessage());
    return false;
}
}

```

Διακρίνουμε, ότι εκτελείται function της βάσης δεδομένων SQ\_HOSPITAL του πακέτου SQ με όνομα check\_access (παράρτημα 2), η οποία παίρνει ως ορίσματα τον χρήστη και το όνομα του προγράμματος που ελέγχεται και αναλόγως αν επιστραφεί τιμή, δηλαδή έχει δηλωθεί ο χρήστης να βλέπει αυτό το πρόγραμμα τότε επιστρέφει true αλλιώς επιστρέφεται false.

#### 4.2.4 Μεταφορά των δεδομένων μεταξύ των δραστηριοτήτων του συστήματος ροής εργασίας

Προκειμένου να έχουμε την ορθή μεταφορά των δεδομένων μεταξύ των δραστηριοτήτων της διαδικασίας, μέσω των data structures που ορίστηκαν στο κεφάλαιο 3, § 3.3.3.1, υπάρχουν τρόποι άντλησης των δεδομένων από μια δραστηριότητα, αλλά και αντίστοιχα αποθήκευσης δεδομένων σε μια δραστηριότητα, προκειμένου να μεταφερθούν στην επόμενη.

Σε κάθε πρόγραμμα ορίζουμε τα παρακάτω αντικείμενα:

```

ExecutionAgent PEA;
Agent agent = new Agent();
agent.setLocator(Agent.LOC_LOCATOR);
agent.setName("LOCAL");
PEA=agent.getExecutionAgent();

```

Και εν συνεχεία εκτελείται το επιθυμητό πρόγραμμα χρησιμοποιώντας αυτόν τον executionagent PEA.

```

// Αποθήκευση του UserID του χρήστη
userid = PEA.userID();
// Ορισμός του input container
input = PEA.inContainer();
// Ορισμός του output container
radissueDS = PEA.outContainer();

```



Διαβάζουμε τις input τιμές ως εξής:

```
π.χ. DOCT_KOD = input.getString(input.structMembers()[1].leaves()[0].fullName());  
π.χ. DOCT_DEPT_KOD= input.getString(input.structMembers()[2].leaves()[0].fullName());  
π.χ. EXAM_DEPT_KOD = input.getString(input.structMembers()[5].leaves()[0].fullName());
```

Αποθηκεύουμε τις νέες τιμές στο νέο output container ως εξής:

```
radissueDS.setLong(input.leaves()[0].fullName(),input.getLong(input.leaves()[0].fullName()  
));  
radissueDS.setString(input.structMembers()[1].leaves()[0].fullName(),  
input.getString(input.structMembers()[1].leaves()[0].fullName()));  
  
radissueDS.setString(input.structMembers()[1].leaves()[1].fullName(),  
input.getString(input.structMembers()[1].leaves()[1].fullName()));  
  
radissueDS.setString(input.structMembers()[1].leaves()[2].fullName(),  
input.getString(input.structMembers()[1].leaves()[2].fullName()));  
  
radissueDS.setString(input.structMembers()[2].leaves()[0].fullName(),  
input.getString(input.structMembers()[2].leaves()[0].fullName()));  
  
radissueDS.setString(input.structMembers()[3].leaves()[0].fullName(),  
input.getString(input.structMembers()[3].leaves()[0].fullName()));  
  
radissueDS.setString(input.structMembers()[3].leaves()[1].fullName(),input.getString(input.s  
tructMembers()[3].leaves()[1].fullName()));  
  
radissueDS.setString(input.structMembers()[3].leaves()[2].fullName(),input.getString(input.s  
tructMembers()[3].leaves()[2].fullName()));  
  
radissueDS.setString(input.structMembers()[3].leaves()[3].fullName(),input.getString(input.s  
tructMembers()[3].leaves()[3].fullName()));  
  
radissueDS.setString(input.structMembers()[3].leaves()[4].fullName(),input.getString(input.s  
tructMembers()[3].leaves()[4].fullName()));  
  
radissueDS.setString(input.structMembers()[3].leaves()[5].fullName(),input.getString(input.s  
tructMembers()[3].leaves()[5].fullName()));  
  
radissueDS.setString(input.structMembers()[5].leaves()[0].fullName(),input.getString(input.s  
tructMembers()[5].leaves()[0].fullName()));  
  
PEA.setOutContainer(radissueDS);
```

#### 4.2.5 Έξοδος προγράμματος

```
radissueDS.setLong("_RC", 1);  
PEA.setOutContainer(radissueDS);  
  
System.exit(0);
```

## 5<sup>ο</sup> ΚΕΦΑΛΑΙΟ

ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ ΕΦΑΡΜΟΓΗΣ

User Manual

## 5 ΑΝΑΛΥΣΗ ΦΟΡΜΩΝ

### 5.1 Εισαγωγή εντολής ακτινολογικής εξέτασης

Πρόκειται για το πρώτο πρόγραμμα που εκτελείται σε όλη την διαδικασία του μοντέλου μας. Αρμόδιος να εκτελέσει το πρόγραμμα είναι ο γιατρός (Θεράπων Ιατρός του ασθενούς). Η φόρμα, «φορτώνει» σε λίστα όλους τους ασθενείς του συγκεκριμένου γιατρού που έχει κάνει login στο σύστημα. Αριστερά της οθόνης φαίνονται τα στοιχεία του εισαχθέντος χρήστη στην κάθε εφαρμογή. Παράλληλα, αποτυπώνεται και το τμήμα στο οποίο εργάζεται ο συγκεκριμένος γιατρός και βρίσκεται κατά την ώρα που ανοίγει η φόρμα και γνωρίζει πλέον τον κωδικό του χρήστη και που εργάζεται αυτός. Δεξιά, έχει την δυνατότητα να επιλέξει τον κατάλληλο ασθενή για τον οποίο θέλει εισάγει την εντολή για ακτινολογική εξέταση. Αφού επιλεγεί ο κωδικός του ασθενή, αντλούνται από την βάση δεδομένων του συστήματος, και τα υπόλοιπα στοιχεία του ασθενούς (επώνυμο, όνομα κ.τ.λ.).

ΙΔΡΥΜΑ: ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΘΗΝΑΣ ΤΜΗΜΑ: ΚΑΡΔΙΟΛΟΓΙΚΟ ΤΜΗΜΑ	Επιλογή Ασθενούς Κωδικός: A100 Επώνυμο: ΠΑΠΑΔΟΠΟΥΛΟΣ Όνομα: ΑΛΕΞΑΝΔΡΟΣ Όνομα Πατρός: ΜΙΧΑΗΛ Διεύθυνση: ΑΓ. ΠΑΝΤΩΝ 18-20 E-mail: dpartsakoulaki@gmail.com
Στοιχεία Γιατρού ΟΝΟΜΑ: ΜΙΧΑΗΛΗΣ ΕΠΙΘΕΤΟ: ΠΑΡΤΣΑΚΟΥΛΑΚΗΣ	

29/06/2009 18:14:48

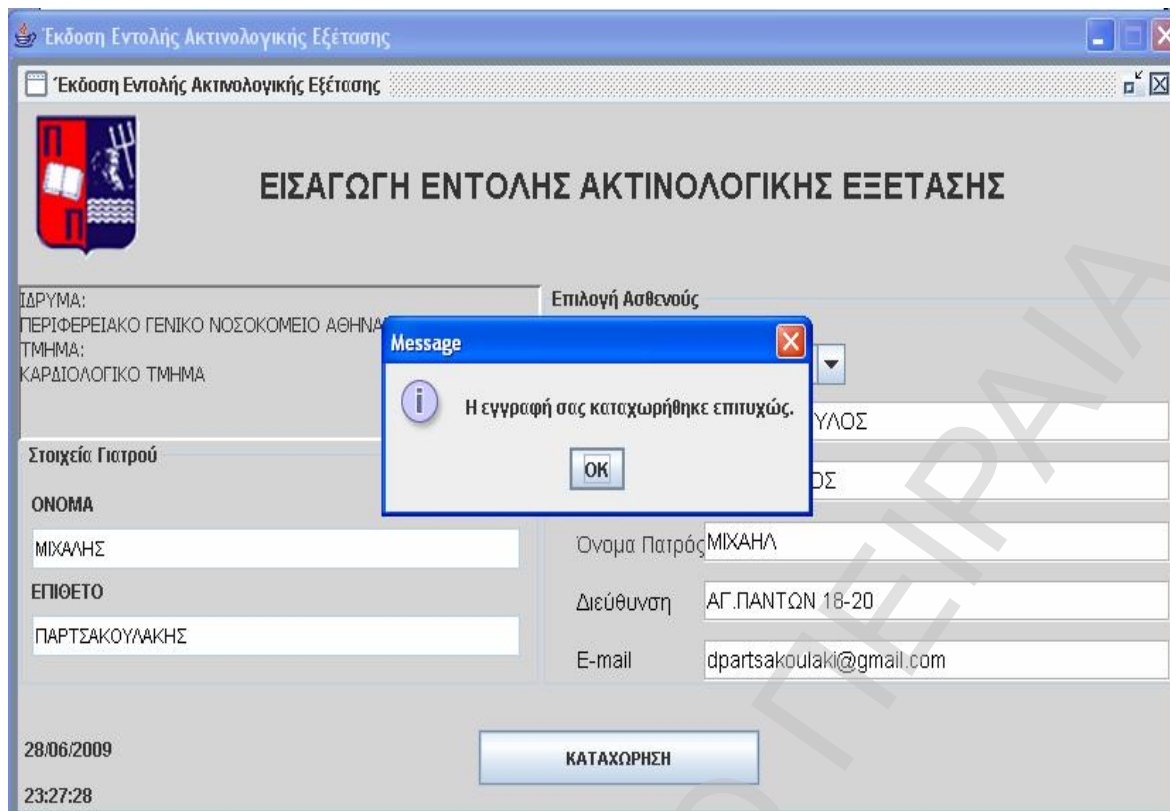
ΚΑΤΑΧΩΡΗΣΗ

**Εικόνα 5-1:** ΕΙΣΑΓΩΓΗ ΕΝΤΟΛΗΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ

Κάτω αριστερά η οθόνη πάντα δείχνει την τρέχουσα ώρα και ημέρα του συστήματος.

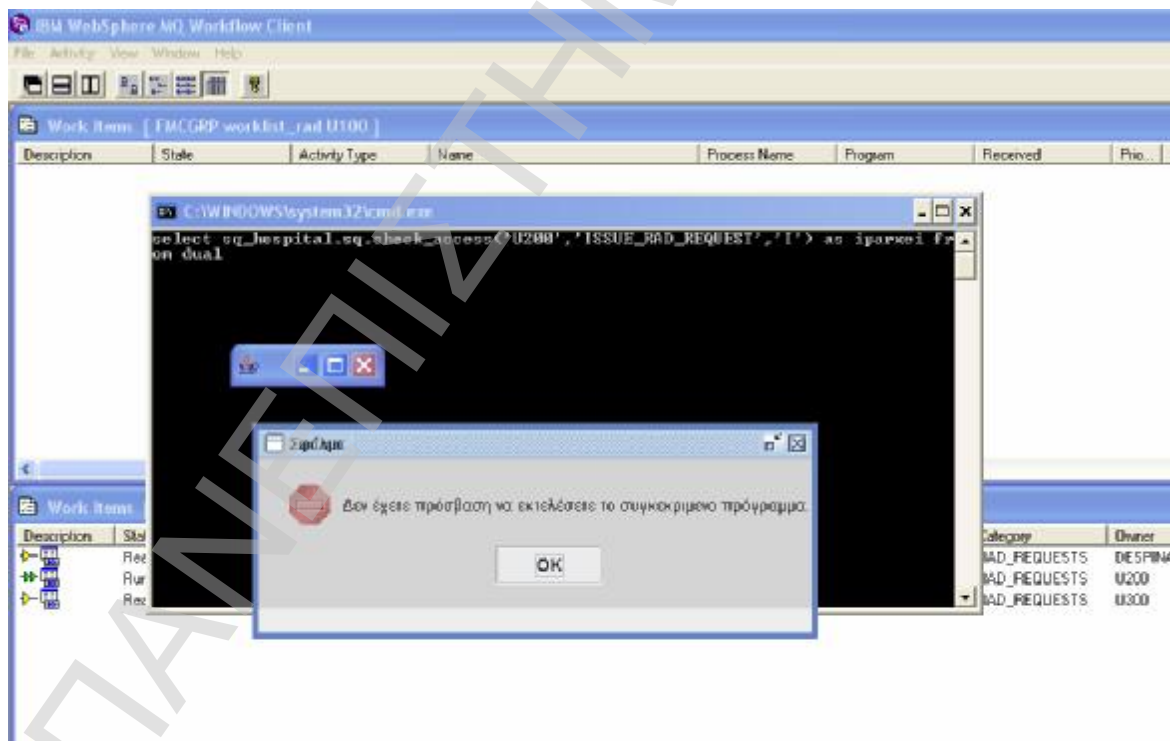
Ο θεράπων ιατρός, όταν πατήσει το κουμπί «ΚΑΤΑΧΩΡΗΣΗ», το πρόγραμμα θα καταχωρήσει την εντολή του ιατρού και η εντολή θα παραπεμφθεί στο κατάλληλο ακτινολογικό τμήμα προς διεκπεραίωση. Εάν ο γιατρός ανήκει σε νοσοκομείο που διαθέτει ακτινολογικό τμήμα, τότε επιλέγεται αυτό ως το ακτινολογικό τμήμα που θα εκτελεστεί η ακτινογραφία. Εάν όχι, τότε επιλέγεται το ακτινολογικό τμήμα του Περιφερειακού Γενικού Νοσοκομείου.

Εάν η εγγραφή καταχωρηθεί στο σύστημα χωρίς κανένα πρόβλημα, τότε ο χρήστης λαμβάνει μήνυμα ορθής καταχώρησης της εντολής του, όπως φαίνεται και στο σχήμα που ακολουθεί. [Εικόνα 5-2.](#)



Εικόνα 5-2: ΕΙΣΑΓΩΓΗ ΕΝΤΟΛΗΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ- Μήνυμα OK

Σε περίπτωση που ο θεράπων ιατρός δεν έχει πρόσβαση στο παραπάνω πρόγραμμα, κατά την κλήση του προγράμματος, ελέγχεται το δικαίωμα πρόσβασης του χρήστη και λαμβάνει μήνυμα λάθους πρόσβασης στο πρόγραμμα.



Εικόνα 5-3: Μήνυμα μη επιτρεπτής πρόσβασης

## 5.2 Λίστα Εντολών Ακτινολογικών Εξετάσεων

Ο κάθε θεράπων ιατρός, έχει την δυνατότητα ανά πάσα στιγμή, να μπορεί να βλέπει λίστα με όλες τις εντολές ακτινολογικών εξετάσεων που έχει εκδώσει ο ίδιος. Στην λίστα αυτή,

βλέπει σε ποιους ασθενείς τις έχει εκδώσει, εάν και πότε έχει καθοριστεί ραντεβού για αυτές και σε ποιόν ακτινολόγο γιατρό έχουν δοθεί προς εκτέλεση (βλ. [Εικόνα 5-4](#))

Επίσης, υπάρχει η δυνατότητα της ανανέωσης δεδομένων της φόρμας, ώστε να μπορεί να δει τις τρέχουσες αλλαγές.

Εικόνα 5-4: ΛΙΣΤΑ ΕΝΤΟΛΩΝ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ

### 5.3 Εκκρεμείς Εντολές Ακτινολογικών Εξετάσεων

Στο πρόγραμμα αυτό, έχει πρόσβαση μόνο χρήστης της ΓΡΑΜΜΑΤΕΙΑΣ του ακτινολογικού τμήματος που έχει δοθεί η εντολή.

Αριστερά της οθόνης φαίνονται τα στοιχεία του χρήστη από την γραμματεία που εκτελεί το πρόγραμμα και σε ποιο ακριβώς τμήμα εργάζεται.

Στην δεξιά μεριά, φαίνονται τα στοιχεία του ασθενούς που αφορά η εντολή που μόλις ήρθε. (βλ. [Εικόνα 5-5](#)).

Πατώντας το κουμπί «ΡΑΝΤΕΒΟΥ», ο χρήστης παραπέμπεται σε μια δεύτερη οθόνη όπου θα πρέπει να καθορίσει το ραντεβού του ασθενούς.

Ελεγχος Εκκρεμών Ακτινολογικών Εντολών

Ελεγχος Εκκρεμών Ακτινολογικών Εντολών

**ΕΚΚΡΕΜΕΙΣ ΕΝΤΟΛΕΣ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ**

**ΙΔΡΥΜΑ:**  
 ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΘΗΝΑΣ  
**ΤΜΗΜΑ:**  
 ΑΚΤΙΝΟΛΟΓΙΚΟ ΤΜΗΜΑ

**Στοιχεία Εξέτασης Ασθενούς**  
 Κωδικός Εξέτασης: 88  
 Επώνυμο: ΠΑΠΑΔΟΠΟΥΛΟΣ  
 Όνομα: ΑΛΕΞΑΝΔΡΟΣ  
 Όνομα Πατρός: ΜΙΧΑΗΛ  
 Διεύθυνση: ΑΓ. ΠΑΝΤΩΝ 18-20  
 E-mail: dpartsakoulaki@gmail.com

**Στοιχεία Γραμματείας**  
**ΟΝΟΜΑ**  
 ΔΕΣΠΟΙΝΑ  
**ΕΠΙΘΕΤΟ**  
 ΠΑΡΤΣΑΚΟΥΛΑΚΗ

29/06/2009  
00:34:38

ΡΑΝΤΕΒΟΥ

Εικόνα 5-5: ΕΚΚΡΕΜΕΙΣ ΕΝΤΟΛΕΣ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ

#### 5.4 Καθορισμός Ραντεβού Εξέτασης

Στην οθόνη αυτή, η γραμματεία έχει την ευχέρεια να καθορίσει το ραντεβού του ασθενούς. Υπάρχουν δύο δυνατότητες για αυτό. Είτε πατώντας το κουμπί «Εύρεση Προτεινόμενου Ραντεβού», το σύστημα αυτόματα να του προτείνει ημέρα, ώρα και γιατρό για το ραντεβού αυτό, εκτελώντας κάποιες διαδικασίες από την βάση δεδομένων εκείνη την στιγμή, είτε μόνη της να επιλέξει – αλλάξει την ημέρα και ώρα του ραντεβού, αλλά και να επιλέξει ακτινολόγο γιατρό για την εξέταση. Εικόνες 5-6, 5-7.



Καθορισμός Ραντεβού Ακτινολογικής Εξέτασης

## ΚΑΘΟΡΙΣΜΟΣ ΡΑΝΤΕΒΟΥ ΕΞΕΤΑΣΗΣ

**ΙΔΡΥΜΑ:**  
ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΘΗΝΑΣ  
**ΤΜΗΜΑ:**  
ΑΚΤΙΝΟΛΟΓΙΚΟ ΤΜΗΜΑ

**Στοιχεία Γραμματείας**  
ΟΝΟΜΑ  
ΔΕΣΠΟΙΝΑ  
ΕΠΙΘΕΤΟ  
ΠΑΡΤΣΑΚΟΥΛΑΚΗ

**Στοιχεία Εξέτασης Ασθενούς**  
Κωδικός Εξέτασης 00  
Επώνυμο ΠΑΠΑΔΟΠΟΥΛΟΣ  
Όνομα ΑΛΕΞΑΝΔΡΟΣ  
Όνομα Πατρός ΜΙΧΑΗΛ  
Διεύθυνση ΑΓ. ΠΑΝΤΩΝ 18-20  
E-mail dpartsakoulaki@gmail.com

**Εύρεση Προτεινόμενου Ραντεβού**

Ακτινολόγος Γιατρός  
U400 ▾ ΑΝΑΣΤΑΣΙΑ ΠΑΠΑΔΟΠΟΥΛΟΥ

Ημερομηνία Ραντεβού 29 Ιούλιος 2009 ▾ Ώρα Ραντεβού 08:00 ▾

29/06/2009 00:51:33 ΚΑΤΑΧΩΡΗΣΗ

+ Ιούλιος 2009 +

Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ	Κυρ
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

29/6/2009 jivacmapicker.com

**Εικόνα 5-6:** ΚΑΘΟΡΙΣΜΟΣ ΡΑΝΤΕΒΟΥ ΕΞΕΤΑΣΗΣ - Ημερομηνία

Καθορισμός Ραντεβού Ακτινολογικής Εξέτασης

## ΚΑΘΟΡΙΣΜΟΣ ΡΑΝΤΕΒΟΥ ΕΞΕΤΑΣΗΣ

**ΙΔΡΥΜΑ:**  
ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΘΗΝΑΣ  
**ΤΜΗΜΑ:**  
ΑΚΤΙΝΟΛΟΓΙΚΟ ΤΜΗΜΑ

**Στοιχεία Γραμματείας**  
ΟΝΟΜΑ  
ΔΕΣΠΟΙΝΑ  
ΕΠΙΘΕΤΟ  
ΠΑΡΤΣΑΚΟΥΛΑΚΗ

**Στοιχεία Εξέτασης Ασθενούς**  
Κωδικός Εξέτασης 00  
Επώνυμο ΠΑΠΑΔΟΠΟΥΛΟΣ  
Όνομα ΑΛΕΞΑΝΔΡΟΣ  
Όνομα Πατρός ΜΙΧΑΗΛ  
Διεύθυνση ΑΓ. ΠΑΝΤΩΝ 18-20  
E-mail dpartsakoulaki@gmail.com

**Εύρεση Προτεινόμενου Ραντεβού**

Ακτινολόγος Γιατρός  
U400 ▾ ΑΝΑΣΤΑΣΙΑ ΠΑΠΑΔΟΠΟΥΛΟΥ

Ημερομηνία Ραντεβού 29 Ιούλιος 2009 ▾ Ώρα Ραντεβού 08:00 ▾

29/06/2009 00:51:33 ΚΑΤΑΧΩΡΗΣΗ

08:00  
08:30  
09:00  
09:30  
10:00  
10:30  
11:00  
11:30

**Εικόνα 5-7:** ΚΑΘΟΡΙΣΜΟΣ ΡΑΝΤΕΒΟΥ ΕΞΕΤΑΣΗΣ - Ώρα

Εφόσον λοιπόν ο χρήστης είναι έτοιμος να καταχωρήσει το ραντεβού, πατώντας το πλήκτρο «ΚΑΤΑΧΩΡΗΣΗ», η εγγραφή αποθηκεύεται στη βάση δεδομένων

(βλ. Εικόνα 5-8)

Καθορισμός Ραντεβού Ακτινολογικής Εξέτασης

ΚΑΘΟΡΙΣΜΟΣ ΡΑΝΤΕΒΟΥ ΕΞΕΤΑΣΗΣ

ΤΑΞΙΝΟΜΗΣΗ: ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΘΗΝΑΣ  
ΤΜΗΜΑ: ΑΚΤΙΝΟΛΟΓΙΚΟ ΤΜΗΜΑ

Στοιχεία Εξέτασης Ασθενούς  
Κωδικός Εξέτασης: 88

Message  
Το ραντεβού καταχωρήθηκε επιτυχώς.  
OK

Εύρεση Προτεινόμενου Ραντεβού

Ακτινολόγος Γιατρός  
U400 | ΑΝΑΣΤΑΣΙΑ ΠΑΠΑΔΟΠΟΥΛΟΥ

Ημερομηνία Ραντεβού: 29 Ιουνίου 2009  
Ώρα Ραντεβού: 08:00

ΚΑΤΑΧΩΡΗΣΗ

29/06/2009  
01:02:38

Εικόνα 5-8: ΚΑΘΟΡΙΣΜΟΣ ΡΑΝΤΕΒΟΥ ΕΞΕΤΑΣΗΣ - Καταχώρηση

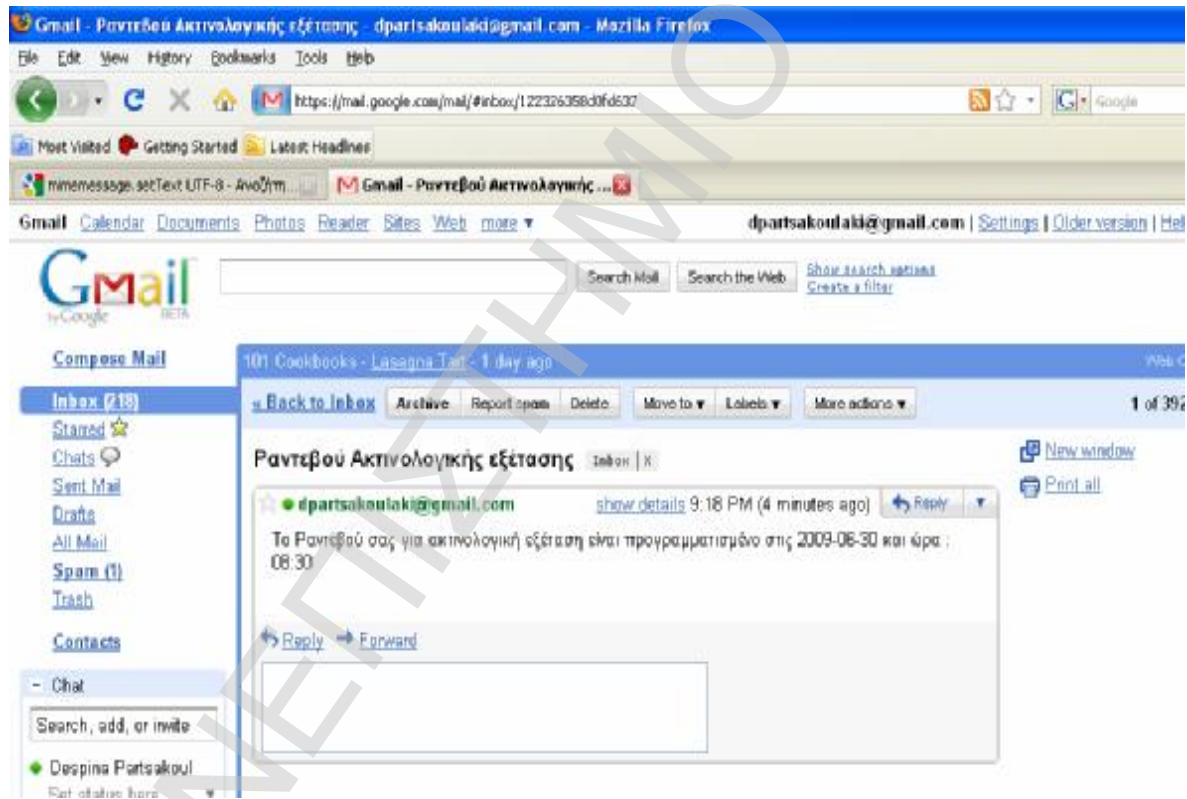
## 5.5 Ειδοποίηση ασθενούς – θεράποντος ιατρού

Η επόμενη αυτόματη διαδικασία του συστήματος είναι να δημιουργηθεί μήνυμα ειδοποίησης, τόσο στον ασθενή, όσο και στον θεράπων ιατρό του ασθενούς. Εφόσον το σύστημα αντλήσει τα e-mails των δυο εμπλεκόμενων, αποστέλλει e-mail και στους δύο, ενημερώνοντάς τους για την ημέρα και ώρα του ραντεβού.  
Εικόνες 5-9, 5-10, 5-11.

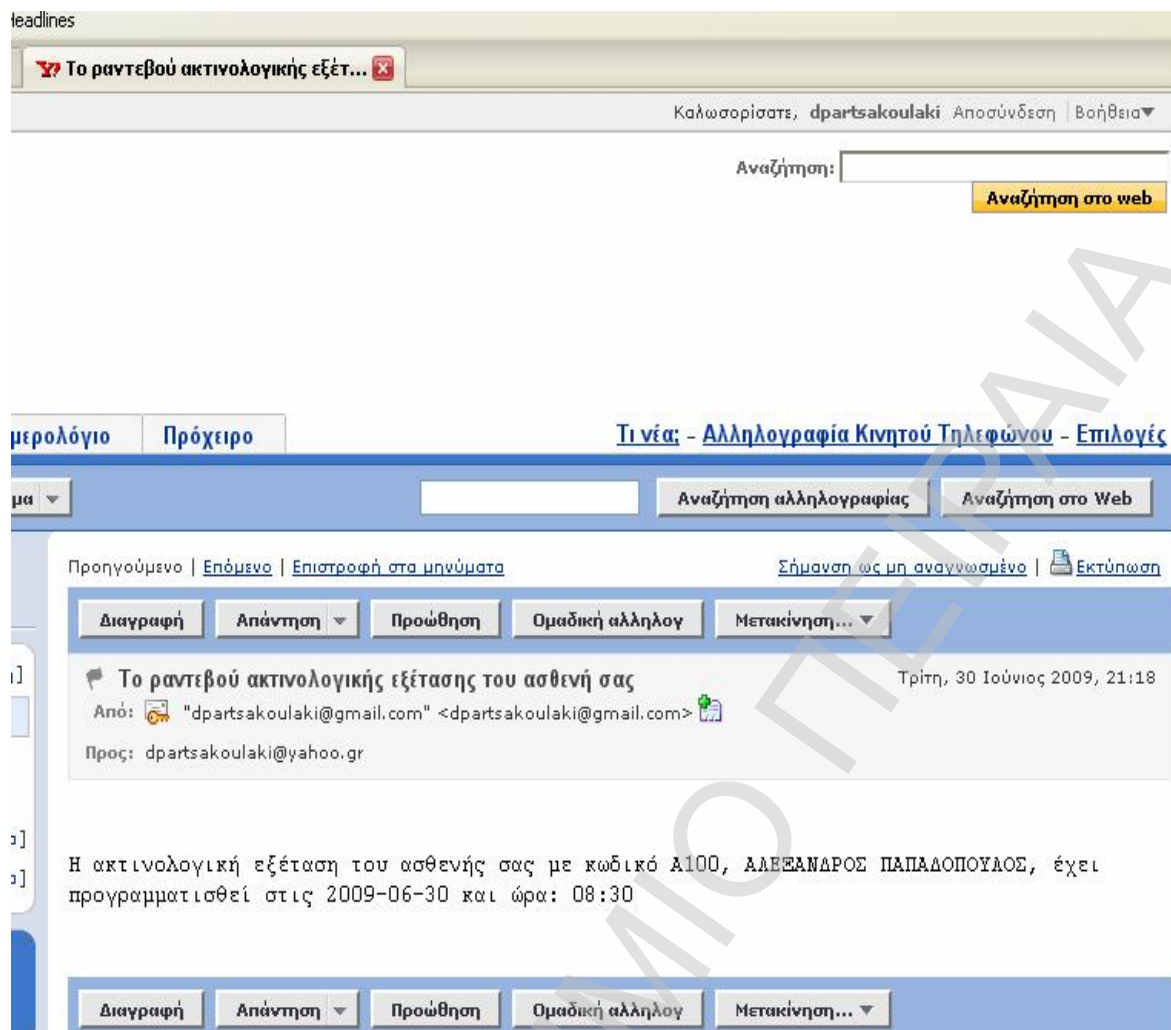


```
C:\WINDOWS\system32\cmd.exe
patient mail dpartsakoulaki@gmail.com
DOCT_EMAIL= dpartsakoulaki@yahoo.gr
DEBUG: JavaMail version 1.3.1
DEBUG: java.io.FileNotFoundException: C:\Program Files\Java\jdk1.5.0_18\jre\lib\
javamail.providers (The system cannot find the file specified)
DEBUG: !anyLoaded
DEBUG: not loading resource: /META-INF/javamail.providers
DEBUG: successfully loaded resource: /META-INF/javamail.default.providers
DEBUG: Tables of loaded providers
DEBUG: Providers Listed By Class Name: {com.sun.mail.smtp.SMTPTransport=javax.ma
il.Provider[TRANSPORT,smtp,com.sun.mail.smtp.SMTPTransport,Sun Microsystems, Inc
], com.sun.mail.imap.IMAPStore=javax.mail.Provider[STORE,imap,com.sun.mail.imap.
IMAPStore,Sun Microsystems, Inc], com.sun.mail.pop3.POP3Store=javax.mail.Provide
r[STORE,pop3,com.sun.mail.pop3.POP3Store,Sun Microsystems, Inc]}
DEBUG: Providers Listed By Protocol: {imap=javax.mail.Provider[STORE,imap,com.su
n.mail.imap.IMAPStore,Sun Microsystems, Inc], pop3=javax.mail.Provider[STORE,pop
3,com.sun.mail.pop3.POP3Store,Sun Microsystems, Inc], smtp=javax.mail.Provider[
TRANSPORT,smtp,com.sun.mail.smtp.SMTPTransport,Sun Microsystems, Inc]}
DEBUG: successfully loaded resource: /META-INF/javamail.default.address.map
DEBUG: !anyLoaded
DEBUG: not loading resource: /META-INF/javamail.address.map
DEBUG: java.io.FileNotFoundException: C:\Program Files\Java\jdk1.5.0_18\jre\lib\
javamail.address.map (The system cannot find the file specified)
HOSPITAL
dpartsakoulaki@gmail.com
```

Εικόνα 5-9: Αποστολή e-mail



Εικόνα 5-10: Παραλαβή e-mail από τον ασθενή.



Εικόνα 5-11: Παραλαβή e-mail από τον θεράποντα ιατρό

## 5.6 Λίστα Ραντεβού Ακτινολογικών Εξετάσεων

Στο επόμενο πρόγραμμα που ξεκινάει, έχει πρόσβαση ο κάθε ακτινολόγος γιατρός, δίνοντάς του την δυνατότητα να βλέπει όλα τα ραντεβού που τον αφορούν. Υπάρχει η δυνατότητα ανανέωσης των δεδομένων, όπου και ξαναεκτελείται το select statement από την βάση με όλα τα ραντεβού του συγκεκριμένου ακτινολόγου. Εικόνα 5-12.

**ΛΙΣΤΑ ΡΑΝΤΕΒΟΥ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ**

ΣΔΡΥΜΑ:  
ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΘΗΝΑΣ  
ΤΜΗΜΑ:  
ΑΚΤΙΝΟΛΟΓΙΚΟ ΤΜΗΜΑ

Ανανέωση Δεδομένων

Κωδ Ασθενή	Επώνυμο Α.	Όνομα Ασθε.	Ον. Πατρός	Κωδ Γιατρού	Επώνυμο Γ.	Όνομα Γ.	Ημέρια Ραντ.	Ώρα Ραντεβ.	Κωδ Γιατρού	Επώνυμο Α.	Όνομα Α.
B100	ΑΝΑΣΤΑΣΙΑ	ΕΥΑΓΓΕΛΟ...	ΝΙΚΟΛΑΟΣ	U200	ΜΙΧΑΗΛΗΣ	ΠΑΡΤΣΑΚΟ...	2009-05-24	08:30	U400	ΑΝΑΣΤΑΣΙΑ	ΠΑΠΑΔΟΣ
A100	ΑΛΕΞΑΝΔΡ...	ΠΑΠΑΔΟΠ...	ΜΙΧΑΗΛ	U200	ΜΙΧΑΗΛΗΣ	ΠΑΡΤΣΑΚΟ...	2009-05-14	08:30	U400	ΑΝΑΣΤΑΣΙΑ	ΠΑΠΑΔΟΣ
A100	ΑΛΕΞΑΝΔΡ...	ΠΑΠΑΔΟΠ...	ΜΙΧΑΗΛ	U200	ΜΙΧΑΗΛΗΣ	ΠΑΡΤΣΑΚΟ...	2009-05-19	08:00	U400	ΑΝΑΣΤΑΣΙΑ	ΠΑΠΑΔΟΣ
A100	ΑΛΕΞΑΝΔΡ...	ΠΑΠΑΔΟΠ...	ΜΙΧΑΗΛ	U200	ΜΙΧΑΗΛΗΣ	ΠΑΡΤΣΑΚΟ...	2009-05-20	08:00	U400	ΑΝΑΣΤΑΣΙΑ	ΠΑΠΑΔΟΣ
B100	ΑΝΑΣΤΑΣΙΑ	ΕΥΑΓΓΕΛΟ...	ΝΙΚΟΛΑΟΣ	U200	ΜΙΧΑΗΛΗΣ	ΠΑΡΤΣΑΚΟ...	2009-05-23	08:00	U400	ΑΝΑΣΤΑΣΙΑ	ΠΑΠΑΔΟΣ
B100	ΑΝΑΣΤΑΣΙΑ	ΕΥΑΓΓΕΛΟ...	ΝΙΚΟΛΑΟΣ	U200	ΜΙΧΑΗΛΗΣ	ΠΑΡΤΣΑΚΟ...	2009-05-24	08:00	U400	ΑΝΑΣΤΑΣΙΑ	ΠΑΠΑΔΟΣ
B100	ΑΝΑΣΤΑΣΙΑ	ΕΥΑΓΓΕΛΟ...	ΝΙΚΟΛΑΟΣ	U200	ΜΙΧΑΗΛΗΣ	ΠΑΡΤΣΑΚΟ...	2009-05-25	08:00	U400	ΑΝΑΣΤΑΣΙΑ	ΠΑΠΑΔΟΣ
A100	ΑΛΕΞΑΝΔΡ...	ΠΑΠΑΔΟΠ...	ΜΙΧΑΗΛ	U200	ΜΙΧΑΗΛΗΣ	ΠΑΡΤΣΑΚΟ...	2009-06-29	08:00	U400	ΑΝΑΣΤΑΣΙΑ	ΠΑΠΑΔΟΣ
B100	ΑΝΑΣΤΑΣΙΑ	ΕΥΑΓΓΕΛΟ...	ΝΙΚΟΛΑΟΣ	U200	ΜΙΧΑΗΛΗΣ	ΠΑΡΤΣΑΚΟ...	2009-05-13	08:00	U400	ΑΝΑΣΤΑΣΙΑ	ΠΑΠΑΔΟΣ
A100	ΑΛΕΞΑΝΔΡ...	ΠΑΠΑΔΟΠ...	ΜΙΧΑΗΛ	U200	ΜΙΧΑΗΛΗΣ	ΠΑΡΤΣΑΚΟ...	2009-05-14	08:30	U400	ΑΝΑΣΤΑΣΙΑ	ΠΑΠΑΔΟΣ

Ημερομηνία: 29/5/2009 Ώρα: 1:31

OK

Εικόνα 5-12: ΛΙΣΤΑ ΡΑΝΤΕΒΟΥ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ

## 5.7 Πληρωμή Ακτινολογικής Εξέτασης

Πληρωμή Ακτινολογικής Εξέτασης

ΠΛΗΡΩΜΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ

ΙΔΡΥΜΑ:  
ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΘΗΝΑΣ  
ΤΜΗΜΑ:  
ΑΚΤΙΝΟΛΟΓΙΚΟ ΤΜΗΜΑ

Στοιχεία Ασθενούς

Κωδικός Εξέτασης 93

Επώνυμο ΠΑΠΑΔΟΠΟΥΛΟΣ

Όνομα ΑΛΕΞΑΝΔΡΟΣ

Όνομα Πατρός ΜΙΧΑΗΛ

Διεύθυνση ΑΓ. ΠΑΝΤΩΝ 18-20

E-mail dpartsakoulaki@gmail.com

Στοιχεία Γραμματείας

ΟΝΟΜΑ  
ΔΕΣΠΟΙΝΑ

ΕΠΙΘΕΤΟ  
ΠΑΡΤΣΑΚΟΥΛΑΚΗ

30/06/2009 Λογαριασμός Ασθενή Ποσό Χρέωσης  
22:04:54 5555555555 15

ΚΑΤΑΧΩΡΗΣΗ

**Εικόνα 5-13:** ΠΛΗΡΩΜΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ

Όπως βλέπουμε και στην εικόνα 5-13, ο ασθενής παραπέμπεται να πληρώσει για την εξέτασή του. Αρμόδιος χρήστης να εκτελέσει το πρόγραμμα πληρωμής είναι και πάλι η γραμματεία του ακτινολογικού.

Αφού καταχωρήσει τον λογαριασμό του ασθενούς και το ποσό χρέωσης της εξέτασης, θα πρέπει να πατήσει το πλήκτρο «ΚΑΤΑΧΩΡΗΣΗ», ώστε να καταχωρηθεί η πληρωμή στο σύστημα. (βλ. [Εικόνα 5-14](#)).

Πληρωμή Ακτινολογικής Εξέτασης

Πληρωμή Ακτινολογικής Εξέτασης

**ΠΛΗΡΩΜΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ**

ΙΔΡΥΜΑ: ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΘΗΝΩΝ  
 ΤΜΗΜΑ: ΑΚΤΙΝΟΛΟΓΙΚΟ ΤΜΗΜΑ

Στοιχεία Ασθενούς

Message  
 Η πληρωμή σας καταχωρήθηκε επιτυχώς.  
 ΟΚ

Στοιχεία Γραμματείας

ΟΝΟΜΑ  
 ΔΕΣΠΟΙΝΑ

ΕΠΙΘΕΤΟ  
 ΠΑΡΤΣΑΚΟΥΛΑΚΗ

Όνομα Πατρός ΜΙΧΑΗΛ  
 Διεύθυνση ΑΓ. ΠΑΝΤΩΝ 18-20  
 E-mail dpartsakoulaki@gmail.com

30/06/2009  
 22:04:54

Λογαριασμός Ασθενή 555555555  
 Ποσό Χρέωσης 15

ΚΑΤΑΧΩΡΗΣΗ

Εικόνα 5-14: ΠΛΗΡΩΜΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ - Καταχώρηση

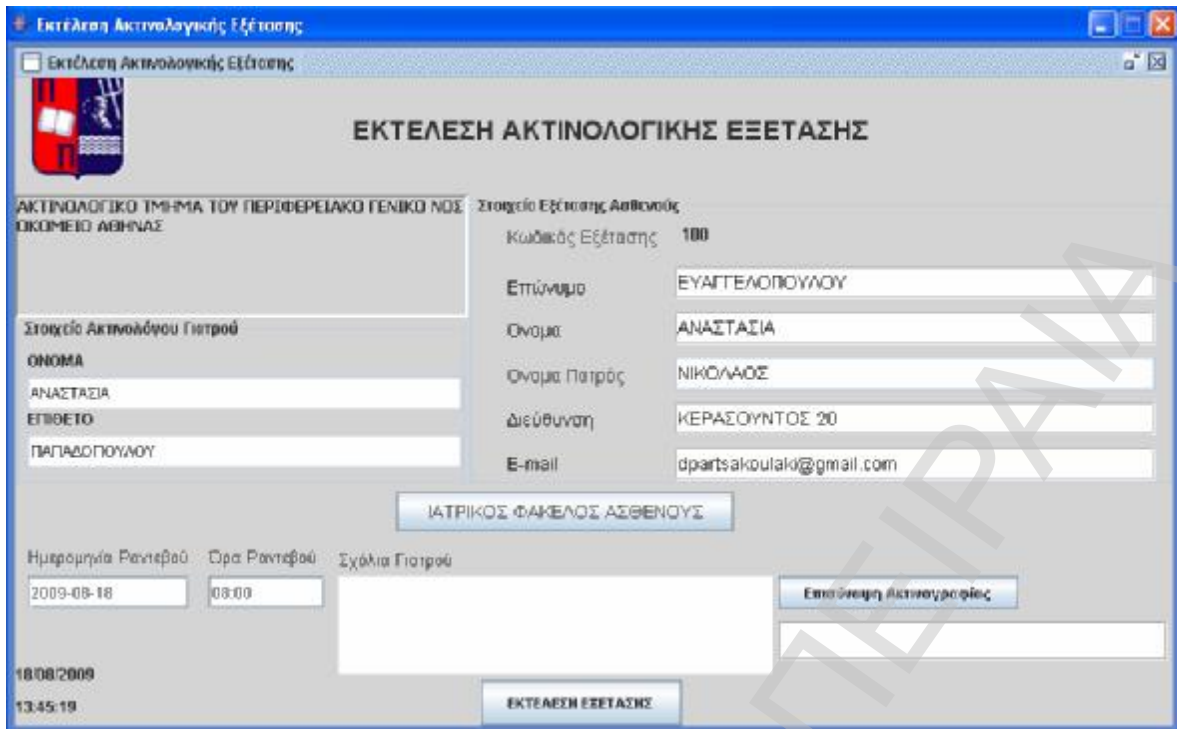
## 5.8 Εκτέλεση ακτινολογικής Εξέτασης

Ο ακτινολόγος γιατρός εν συνεχεία, είναι αρμόδιος να εκτελέσει την εξέταση στον ασθενή. Αφού γίνει η εξέταση έξω- συστημικά από τον βοηθό ακτινολόγου, ο ακτινολόγος θα πρέπει να επισυνάψει την φωτογραφία της ακτινογραφίας ώστε να σταλεί στον φάκελο του ασθενούς αλλά και στον θεράπων ιατρό του που αρχικά την ζήτησε.

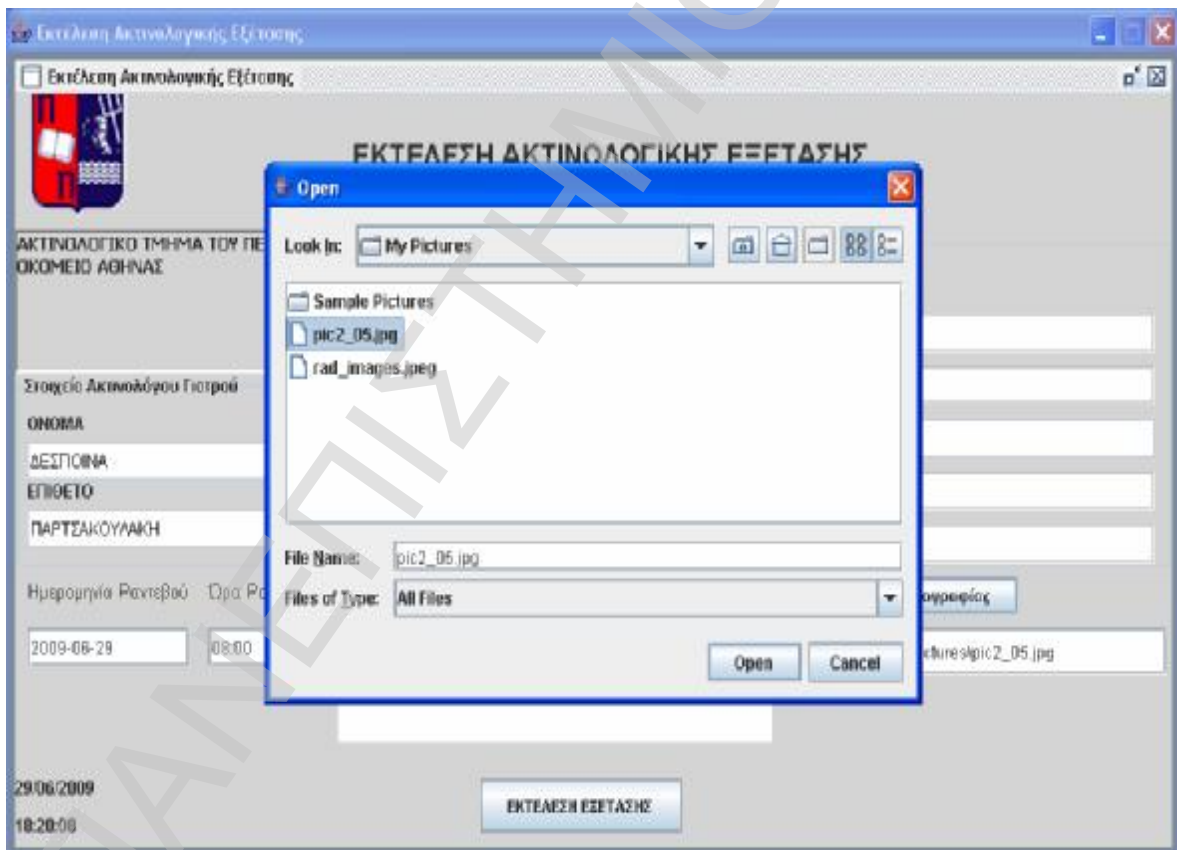
Παράλληλα, θα πρέπει να συμπληρώσει και τα δικά του σχόλια για την ακτινογραφία και την κατάσταση του ασθενούς.

Πατώντας το πλήκτρο «ΕΚΤΕΛΕΣΗ ΕΞΕΤΑΣΗΣ», η εγγραφή καταχωρείται στο σύστημα μαζί με την φωτογραφία στην βάση δεδομένων. Εικόνες 5-15, 5-16, 5-17, 5-18.

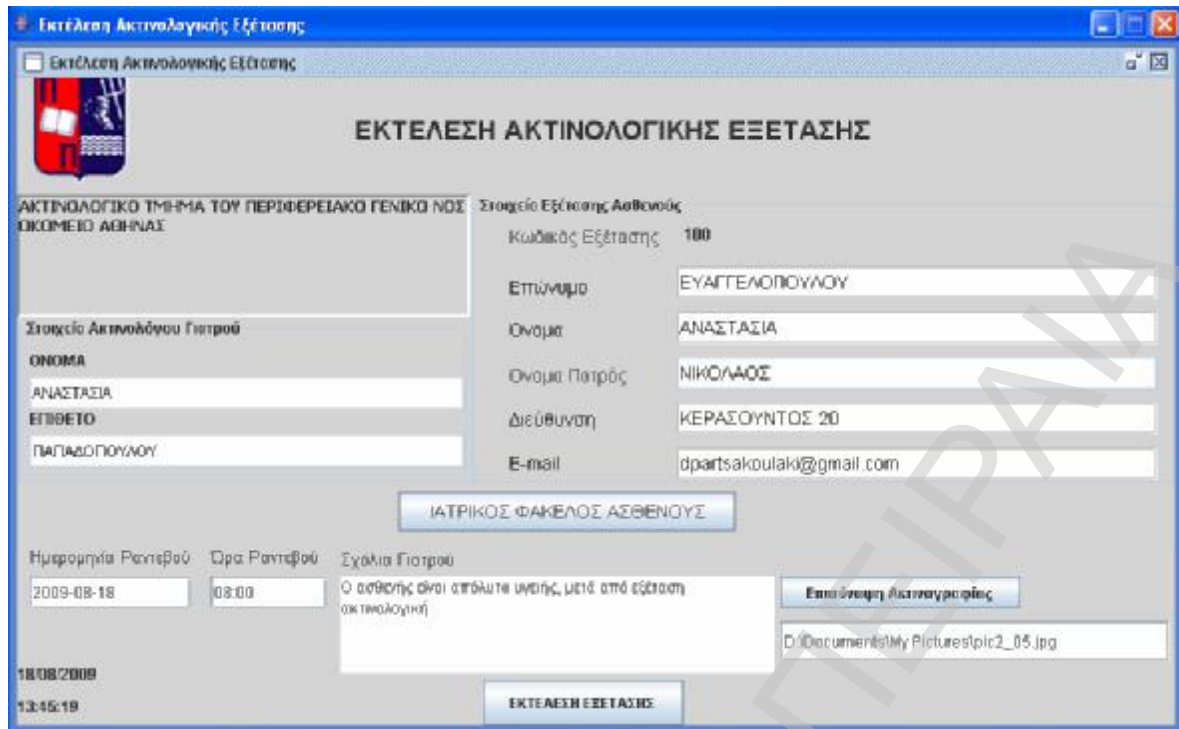




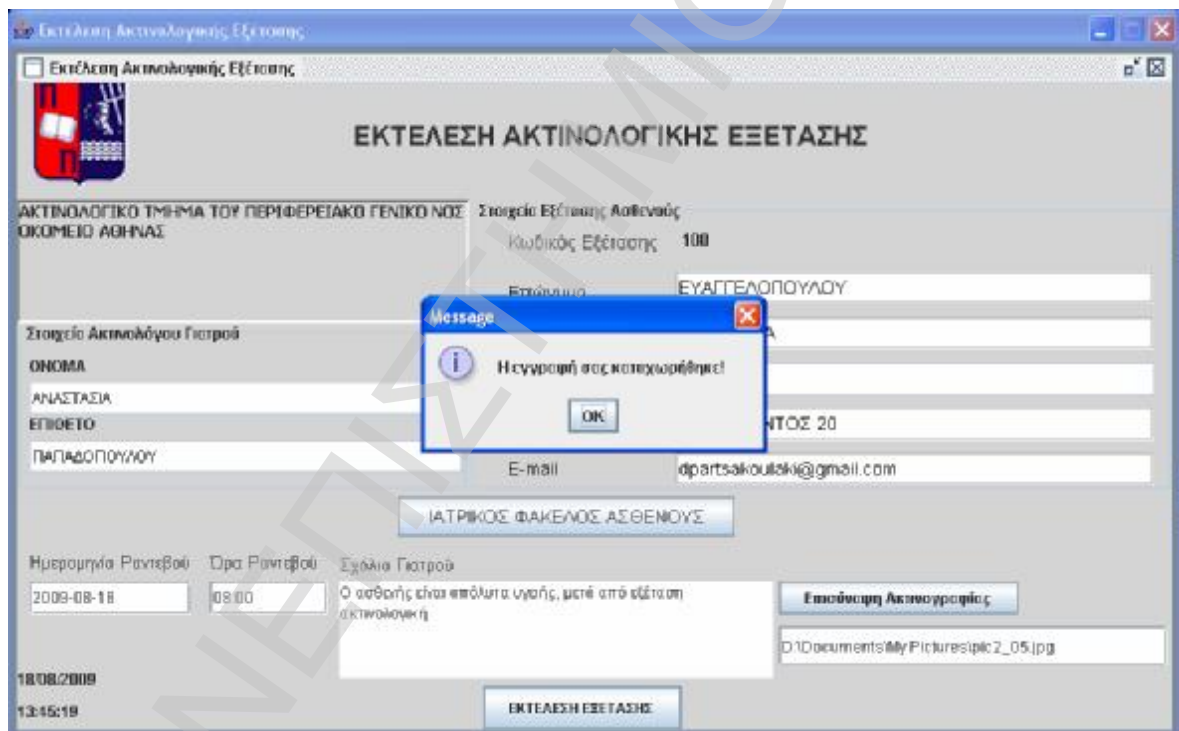
Εικόνα 5-15: ΕΚΤΕΛΕΣΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ



Εικόνα 5-16: ΕΚΤΕΛΕΣΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ-επισύναψη ακτινογραφίας



Εικόνα 5-17: ΕΚΤΕΛΕΣΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ- σχόλια γιατρού



Εικόνα 5-18: ΕΚΤΕΛΕΣΗ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ- καταχώρηση εγγραφής

Παράλληλα, έχει την δυνατότητα να ελέγξει τον ιατρικό φάκελο του ασθενούς, πατώντας το πλήκτρο «ΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ ΑΣΘΕΝΟΥΣ», όπου εμφανίζονται όλα τα σταθερά προσωπικά στοιχεία τους ασθενούς, καθώς και όλες οι προηγούμενες ακτινολογικές εξετάσεις του συγκεκριμένου ασθενούς. [Εικόνα 5-19](#).

Εικόνα 5-19: ΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ ΑΣΘΕΝΟΥΣ

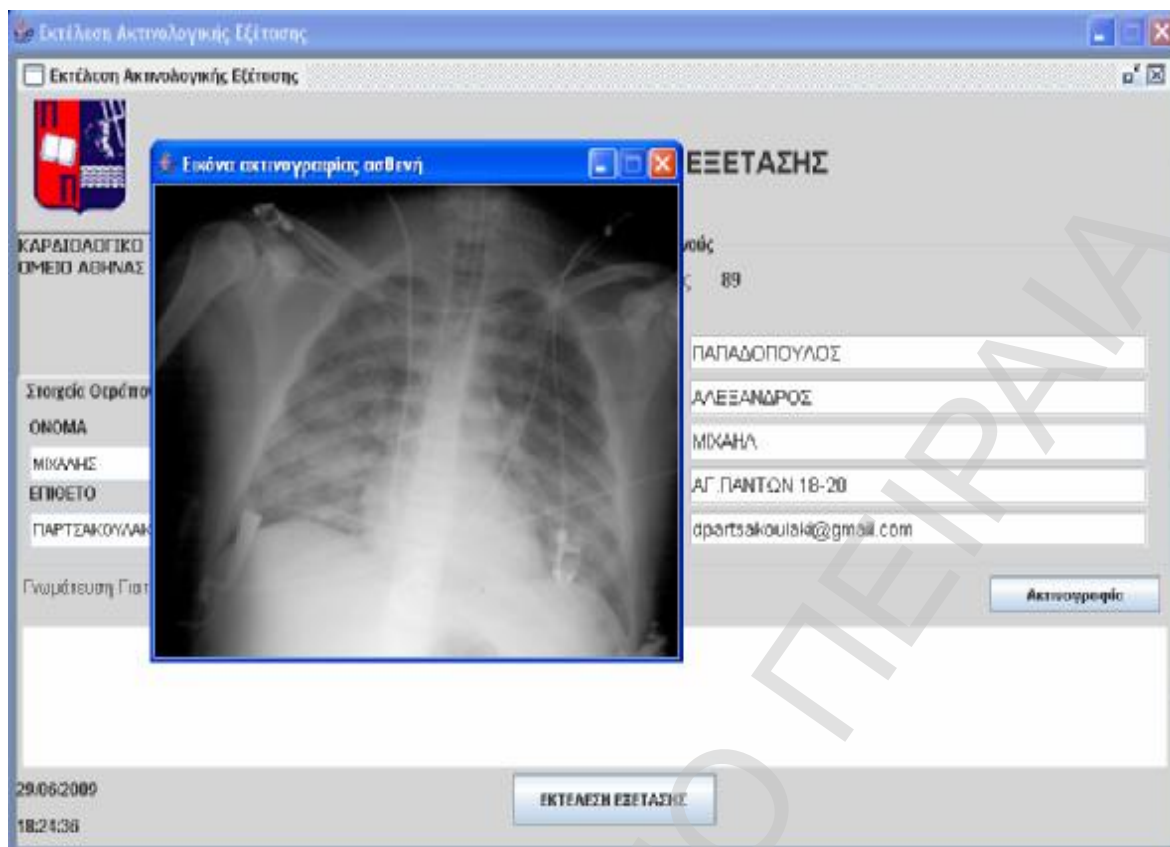
## 5.9 Έλεγχος Ακτινολογικής Εξέτασης

Εφόσον όλα τα προηγούμενα βήματα εκτελέστηκαν επιτυχώς και ο ασθενής έκανε την εξέτασή του, ο θεράπων ιατρός είναι πλέον έτοιμος να δώσει την γνωμάτευσή του. (βλ. Εικόνα 5-20)

Εικόνα 5-20: ΕΛΕΓΧΟΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ

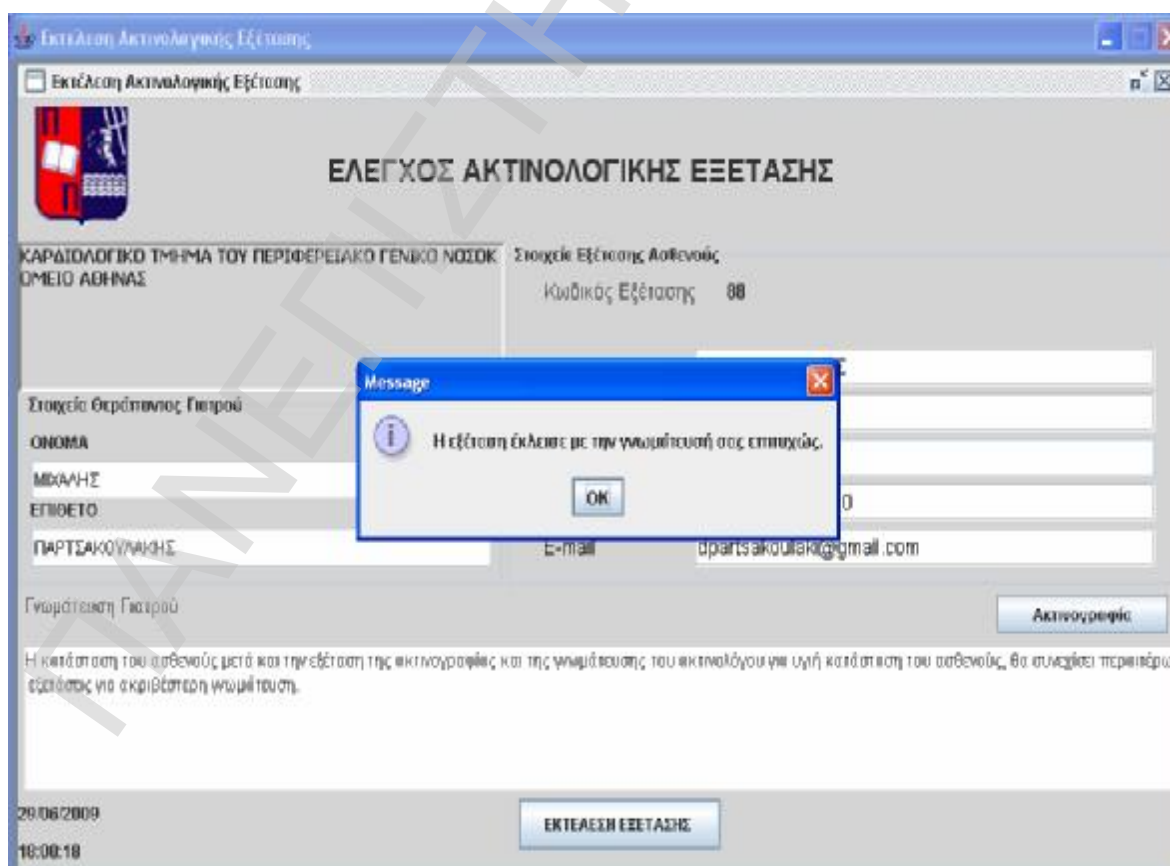
Έχει την δυνατότητα να αντλήσει την ακτινογραφία από την βάση δεδομένων και να συμπληρώσει την γνωμάτευσή του. (βλ. Εικόνα 5-21)





**Εικόνα 5-21:** ΕΛΕΓΧΟΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ – Άντληση Ακτινογραφίας

Εν συνεχεία, πατώντας το πλήκτρο «ΕΚΤΕΛΕΣΗ ΕΞΕΤΑΣΗΣ», η εξέταση κλείνει και καταχωρείται στον φάκελο του ασθενούς. Εικόνα 5-22.



**Εικόνα 5-22:** ΕΛΕΓΧΟΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ – Κλείσιμο εξέτασης

Όλη η παραπάνω διαδικασία φαίνεται ότι εκτελέστηκε μέσα από workflow client, με το παρακάτω διάγραμμα.

## 5.10 Έλεγχος Παλαιότερων Ακτινολογικών Αναφορών Ασθενούς

Ο κάθε ακτινολόγος έχει την δυνατότητα εφόσον εκτελέσει κάποιες ακτινολογικές εξετάσεις, να μπορεί να ξαναδεί παλιότερες εξετάσεις που έχει εκτελέσει ο ίδιος σε διάφορους ασθενείς. Στην φόρμα αυτή έχει την δυνατότητα να επιλέξει έναν κωδικό ασθενή από μια λίστα ασθενών στους οποίους έχει εκτελέσει ακτινογραφία και εν συνεχεία κάποια από τις εξετάσεις αυτού του ασθενούς. Παράλληλα, μπορεί να δει και τον ιατρικό φάκελο του ασθενούς.

Στην παρακάτω εικόνα, επιλέγεται ο κωδικός ενός ασθενούς του και εν συνεχεία πατώντας το πλήκτρο « ⇨ », γεμίζει η λίστα με τους κωδικούς εξετάσεων αυτού του ασθενούς.

ΑΚΤΙΝΟΛΟΓΙΚΗ ΑΝΑΦΟΡΑ ΑΣΘΕΝΟΥΣ

ΙΔΡΥΜΑ:  
ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΘΗΝΑΣ  
ΤΜΗΜΑ:  
ΑΚΤΙΝΟΛΟΓΙΚΟ ΤΜΗΜΑ

Στοιχεία Γιατρού  
ΟΝΟΜΑ  
ΑΝΑΣΤΑΣΙΑ  
ΕΠΙΘΕΤΟ  
ΠΑΠΑΔΟΠΟΥΛΟΥ

Επιλογή Ασθενούς - Εξέτασης  
Κωδικός Ασθενούς  
Κωδικός Εξέτασης  
Επώνυμο  
Όνομα  
Όνομα Πατρός  
Διεύθυνση  
E-mail

Ημερομηνία Εξέτασης  
Γνωμάτευση Ακτινολόγου  
Γνωμάτευση Θεράποντος Ιατρού

ΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ ΑΣΘΕΝΟΥΣ

Ακτινογραφία

ΚΛΕΙΣΙΜΟ ΕΓΓΡΑΦΗΣ

18/08/2009  
13:52:19

Εικόνα 5-23: ΑΚΤΙΝΟΛΟΓΙΚΗ ΑΝΑΦΟΡΑ ΑΣΘΕΝΟΥΣ – επιλογή ασθενούς

Στην συνέχεια, ο χρήστης έχει την δυνατότητα να επιλέξει από την λίστα των εξετάσεων, έναν κωδικό εξέτασης (μια ακτινολογική εξέταση).

**ΑΚΤΙΝΟΛΟΓΙΚΗ ΑΝΑΦΟΡΑ ΑΣΘΕΝΟΥΣ**

ΙΔΡΥΜΑ:  
 ΠΕΡΙΦΕΡΕΙΑΚΟ ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΘΗΝΑΣ  
 ΤΜΗΜΑ:  
 ΑΚΤΙΝΟΛΟΓΙΚΟ ΤΜΗΜΑ

**Επιλογή Ασθενούς - Εξέτασης**  
 Κωδικός Ασθενούς: B100    Κωδικός Εξέτασης: 43  
 Επώνυμο: ΕΥΑΓΓΕΛΟΠΟΥΛΟΥ  
 Όνομα: ΑΝΑΣΤΑΣΙΑ  
 Όνομα Πατρός: ΝΙΚΟΛΑΟΣ  
 Διεύθυνση: ΚΕΡΑΣΟΥΝΤΟΣ 20  
 E-mail: dpartsakoulaki@gmail.com

Ημερομηνία Εξέτασης: 2009-08-17 14:04:06.0  
 Γνωμάτευση Ακτινολόγου: Όλα δείχνουν ότι ο ασθενής δεν πάσχει από τίποτα.  
 Γνωμάτευση Θεράποντος Ιατρού: ο ασθενής είναι απόλυτα υγιής.

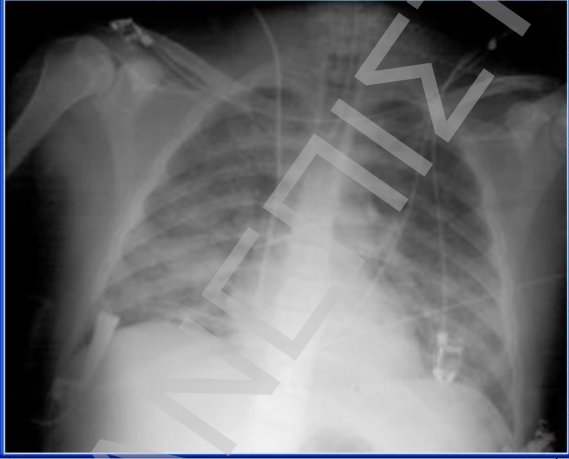
18/08/2009  
 13:52:19

ΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ ΑΣΘΕΝΟΥΣ    Ακτινογραφία    ΚΛΕΙΣΙΜΟ ΕΓΓΡΑΦΗΣ

**Εικόνα 5-24:** ΑΚΤΙΝΟΛΟΓΙΚΗ ΑΝΑΦΟΡΑ ΑΣΘΕΝΟΥΣ – επιλογή εξέτασης

Αφού έχει επιλεγεί η εξέταση, συμπληρώνονται όλα τα πεδία με τα στοιχεία του ασθενούς και της εξέτασης π.χ. ημερομηνία εξέτασης και γνωματεύσεις ιατρών και παράλληλα, μπορεί πατώντας το πλήκτρο «Ακτινογραφία», να δει και ελέγξει την ακτινογραφία αυτής της εξέτασης, αντλώντας την από την βάση δεδομένων.

**Εικόνα ακτινογραφίας ασθενή**



**ΑΚΤΙΝΟΛΟΓΙΚΗ ΑΝΑΦΟΡΑ ΑΣΘΕΝΟΥΣ**

**Επιλογή Ασθενούς - Εξέτασης**  
 Κωδικός Ασθενούς: B100    Κωδικός Εξέτασης: 43  
 Επώνυμο: ΕΥΑΓΓΕΛΟΠΟΥΛΟΥ  
 Όνομα: ΑΝΑΣΤΑΣΙΑ  
 Όνομα Πατρός: ΝΙΚΟΛΑΟΣ  
 Διεύθυνση: ΚΕΡΑΣΟΥΝΤΟΣ 20  
 E-mail: dpartsakoulaki@gmail.com

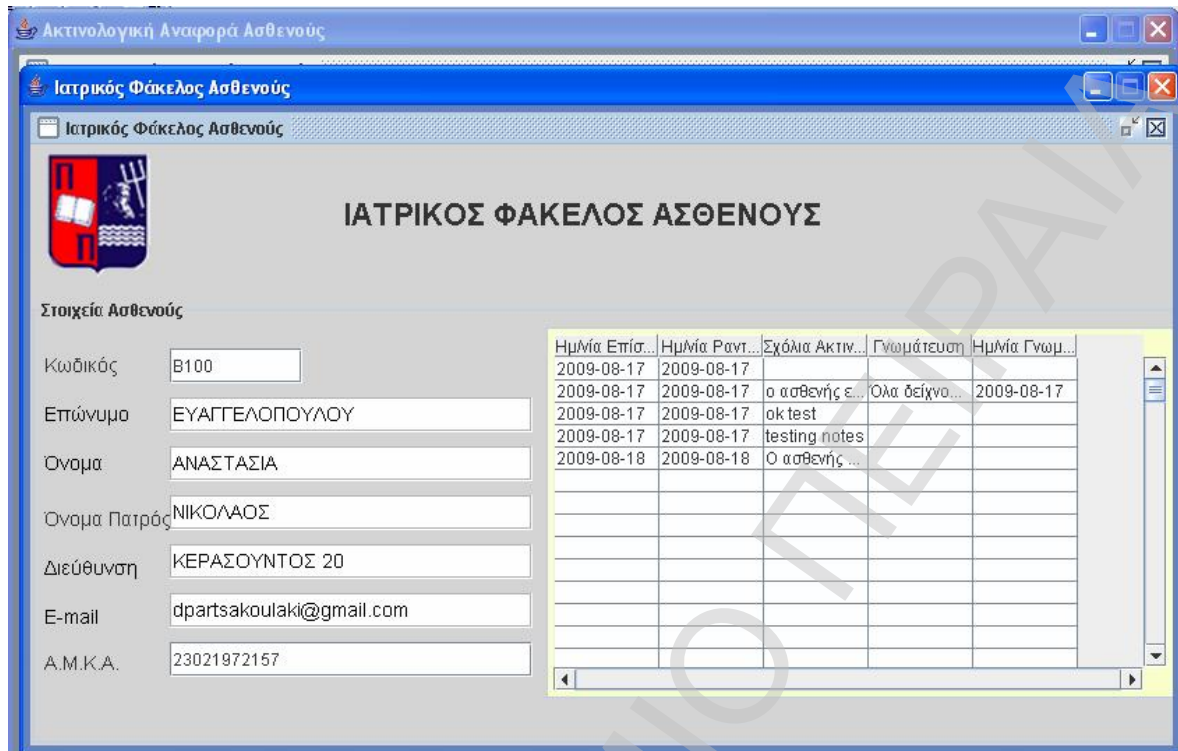
Ημερομηνία Εξέτασης: 2009-08-17 14:04:06.0  
 Γνωμάτευση Ακτινολόγου: Όλα δείχνουν ότι ο ασθενής δεν πάσχει από τίποτα.  
 Γνωμάτευση Θεράποντος Ιατρού: ο ασθενής είναι απόλυτα υγιής.

18/08/2009  
 13:52:19

ΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ ΑΣΘΕΝΟΥΣ    Ακτινογραφία    ΚΛΕΙΣΙΜΟ ΕΓΓΡΑΦΗΣ

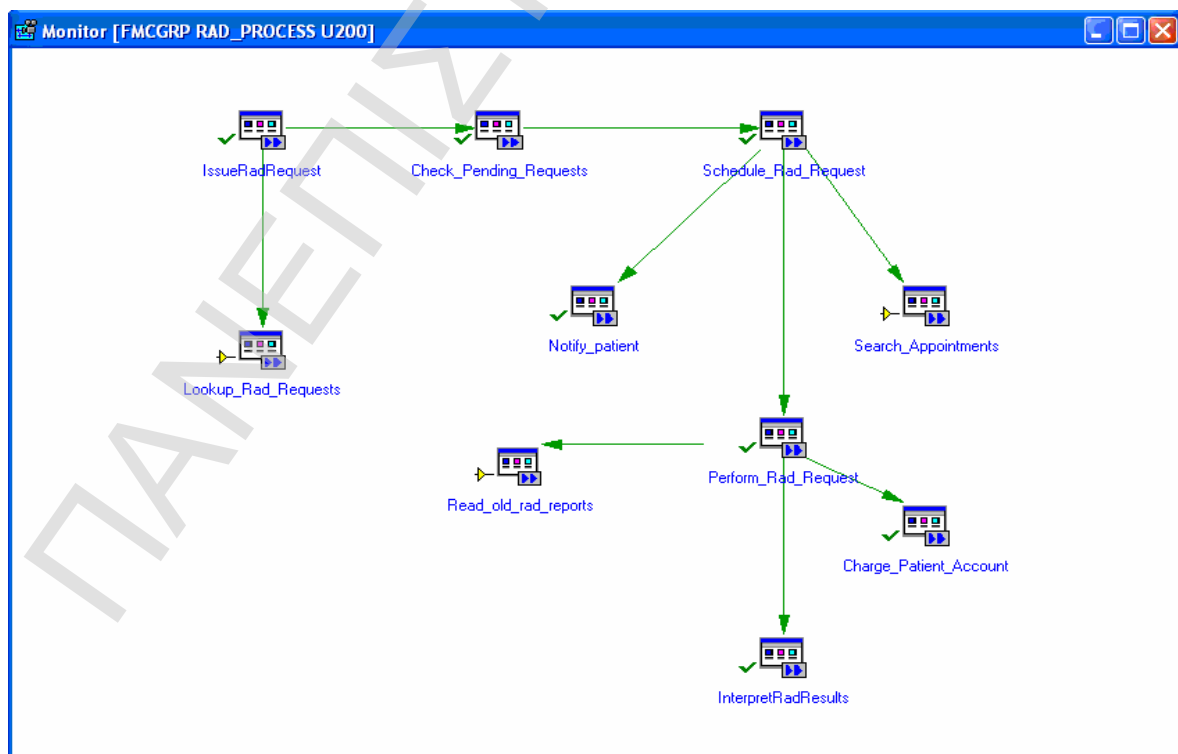
**Εικόνα 5-25:** ΑΚΤΙΝΟΛΟΓΙΚΗ ΑΝΑΦΟΡΑ ΑΣΘΕΝΟΥΣ – έλεγχος ακτινογραφίας

Τέλος, έχει την δυνατότητα ανά πάσα στιγμή να ελέγξει και τον ιατρικό φάκελο του ασθενούς, αφού σε αυτό το σημείο κρίνεται χρήσιμη η επιλογή αυτή. Πατώντας λοιπόν το πλήκτρο «ΙΑΤΡΙΚΟΣ ΦΑΚΕΛΟΣ ΑΣΘΕΝΟΥΣ», μπορεί να δει την εικόνα όλων των εξετάσεων του ασθενούς.



Εικόνα 5-26: ΑΚΤΙΝΟΛΟΓΙΚΗ ΑΝΑΦΟΡΑ ΑΣΘΕΝΟΥΣ – ιατρικός φάκελος ασθενούς

Σε αυτό το σημείο, έχει ολοκληρωθεί όλος ο κύκλος εργασιών της διαδικασίας και αυτό μπορεί να διαπιστωθεί και από το monitor στο workflow client, όπου όλες οι δραστηριότητες έχουν σημειωθεί με πράσινο σημείο επιτυχούς εκτέλεσης, όπως φαίνεται και στο παρακάτω διάγραμμα. Οι δραστηριότητες που έχουν κίτρινο σημείο, μπορούν απλώς να ξαναεκτελεστούν.



Εικόνα 5-27: Διάγραμμα ροής εκτέλεσης της διαδικασίας

# ΠΑΡΑΡΤΗΜΑ 1

## ΠΙΝΑΚΕΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

### ΣΧΗΜΑ ΒΑΣΗΣ HOSPITAL

#### 1) ΠΙΝΑΚΑΣ ΙΔΡΥΜΑ (HOSPITAL.IDRIMA)

```
CREATE TABLE HOSPITAL.IDRIMA
```

```
(  
  IDRIMA_KOD  VARCHAR2(10 BYTE)          NOT NULL,  
  IDRIMA_TYPE VARCHAR2(5 BYTE)           NOT NULL,  
  IDRIMA_TITLE VARCHAR2(50 BYTE)         NOT NULL,  
  IDRIMA_DESCR VARCHAR2(150 BYTE),  
  ADDRESS     VARCHAR2(150 BYTE),  
  TAX_KOD     VARCHAR2(10 BYTE),  
  PHONE1      NUMBER,  
  PHONE2      NUMBER
```

```
)  
TABLESPACE USERS  
PCTUSED 0  
PCTFREE 10  
INITRANS 1  
MAXTRANS 255  
STORAGE (  
  INITIAL      64K  
  MINEXTENTS  1  
  MAXEXTENTS  UNLIMITED  
  PCTINCREASE 0  
  BUFFER_POOL DEFAULT
```

```
)  
LOGGING  
NOCOMPRESS  
NOCACHE  
NOPARALLEL  
MONITORING;
```

```
COMMENT ON COLUMN HOSPITAL.IDRIMA.IDRIMA_KOD IS 'Κωδικός Ιδρύματος (Νοσοκομείου, ΠΓΝ, ΚΥ)';
```

```
COMMENT ON COLUMN HOSPITAL.IDRIMA.IDRIMA_TYPE IS "'1': ΝΟΣΟΚΟΜΕΙΟ, '2': Π.Γ.Ν., '3': ΚΕΝΤΡΟ ΥΓΕΙΑΣ';
```

```
COMMENT ON COLUMN HOSPITAL.IDRIMA.IDRIMA_TITLE IS 'Τίτλος Ιδρύματος';
```

```
COMMENT ON COLUMN HOSPITAL.IDRIMA.IDRIMA_DESCR IS 'Περιγραφή';
```

```
COMMENT ON COLUMN HOSPITAL.IDRIMA.ADDRESS IS 'Ταχ.Διεύθυνση Ιδρύματος';
```

```
COMMENT ON COLUMN HOSPITAL.IDRIMA.TAX_KOD IS 'Ταχ.Κώδικας';
```

```
COMMENT ON COLUMN HOSPITAL.IDRIMA.PHONE1 IS 'Τηλέφωνο1';
```

```
COMMENT ON COLUMN HOSPITAL.IDRIMA.PHONE2 IS 'Τηλέφωνο2';
```

```
CREATE UNIQUE INDEX HOSPITAL.PK_IDRIMA ON HOSPITAL.IDRIMA  
(IDRIMA_KOD)
```

```
LOGGING  
TABLESPACE USERS  
PCTFREE 10  
INITRANS 2  
MAXTRANS 255
```

```

STORAGE (
    INITIAL      64K
    MINEXTENTS  1
    MAXEXTENTS  UNLIMITED
    PCTINCREASE 0
    BUFFER_POOL  DEFAULT
)
NOPARALLEL;

```

```

ALTER TABLE HOSPITAL.IDRIMA ADD (
    CONSTRAINT PK_IDRIMA
    PRIMARY KEY
    (IDRIMA_KOD)
    USING INDEX
    TABLESPACE USERS
    PCTFREE 10
    INITRANS 2
    MAXTRANS 255
    STORAGE (
        INITIAL      64K
        MINEXTENTS  1
        MAXEXTENTS  UNLIMITED
        PCTINCREASE  0
    ));

```

## 2) ΠΙΝΑΚΑΣ ΤΜΗΜΑΤΑ (HOSPITAL.DEPARTMENTS)

```

CREATE TABLE HOSPITAL.DEPARTMENTS

```

```

(
    DEPT_KOD   VARCHAR2(10 BYTE)      NOT NULL,
    IDRIMA_KOD VARCHAR2(10 BYTE),
    DEPT_TYPE  VARCHAR2(5 BYTE)       NOT NULL,
    DEPT_TITLE VARCHAR2(150 BYTE)
)

```

```

TABLESPACE USERS

```

```

PCTUSED 0

```

```

PCTFREE 10

```

```

INITRANS 1

```

```

MAXTRANS 255

```

```

STORAGE (

```

```

    INITIAL      64K

```

```

    MINEXTENTS  1

```

```

    MAXEXTENTS  UNLIMITED

```

```

    PCTINCREASE 0

```

```

    BUFFER_POOL  DEFAULT
)

```

```

LOGGING

```

```

NOCOMPRESS

```

```

NOCACHE

```

```

NOPARALLEL

```

```

MONITORING;

```

```

COMMENT ON COLUMN HOSPITAL.DEPARTMENTS.DEPT_KOD IS 'Κωδικός Τμήματος';

```

```

COMMENT ON COLUMN HOSPITAL.DEPARTMENTS.IDRIMA_KOD IS 'Κωδικός Ιδρύματος που ανήκει';

```

```

COMMENT ON COLUMN HOSPITAL.DEPARTMENTS.DEPT_TYPE IS 'Τύπος Τμήματος "ΚΤ": ΚΛΙΝΙΚΟ ΤΜΗΜΑ, "ΑΤ":
ΑΚΤΙΝΟΛΟΓΙΚΟ, "ΕΞ": ΕΞΩΤΕΡΙΚΑ ΙΑΤΡΕΙΑ, "ΕΠ": ΕΠΕΙΓΟΝΤΑ';

```

```

COMMENT ON COLUMN HOSPITAL.DEPARTMENTS.DEPT_TITLE IS 'Περιγραφή Τμήματος';

```

```

CREATE UNIQUE INDEX HOSPITAL.PK_DEPARTMENTS ON HOSPITAL.DEPARTMENTS
(DEPT_KOD)
LOGGING
TABLESPACE USERS
PCTFREE 10
INITRANS 2
MAXTRANS 255
STORAGE (
    INITIAL 64K
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    PCTINCREASE 0
    BUFFER_POOL DEFAULT
)
NOPARALLEL;

```

```

ALTER TABLE HOSPITAL.DEPARTMENTS ADD (
    CONSTRAINT PK_DEPARTMENTS
    PRIMARY KEY
    (DEPT_KOD)
    USING INDEX
    TABLESPACE USERS
    PCTFREE 10
    INITRANS 2
    MAXTRANS 255
    STORAGE (
        INITIAL 64K
        MINEXTENTS 1
        MAXEXTENTS UNLIMITED
        PCTINCREASE 0
    ));

```

```

ALTER TABLE HOSPITAL.DEPARTMENTS ADD (
    CONSTRAINT FK_DEPT_IDRIMA
    FOREIGN KEY (IDRIMA_KOD)
    REFERENCES HOSPITAL.IDRIMA (IDRIMA_KOD));

```

### 3) ΠΙΝΑΚΑΣ ΓΙΑΤΡΟΙ (HOSPITAL.DOCTORS)

```

CREATE TABLE HOSPITAL.DOCTORS
(
    DOCT_KOD VARCHAR2(10 BYTE) NOT NULL,
    EIDIK_KOD VARCHAR2(10 BYTE) NOT NULL,
    DEPT_KOD VARCHAR2(10 BYTE) NOT NULL,
    DOCT_NAME VARCHAR2(20 BYTE) NOT NULL,
    DOCT_SURNAME VARCHAR2(50 BYTE) NOT NULL,
    DOCT_FATHERNAME VARCHAR2(20 BYTE)
)
TABLESPACE USERS
PCTUSED 0
PCTFREE 10
INITRANS 1
MAXTRANS 255
STORAGE (
    INITIAL 64K
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    PCTINCREASE 0
    BUFFER_POOL DEFAULT
)
LOGGING

```

```
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;
```

```
COMMENT ON COLUMN HOSPITAL.DOCTORS.DOCT_KOD IS 'ΚΩΔΙΚΟΣ ΓΙΑΤΡΟΥ';
COMMENT ON COLUMN HOSPITAL.DOCTORS.EIDIK_KOD IS 'ΚΩΔΙΚΟΣ ΕΙΔΙΚΟΤΗΤΑΣ ΓΙΑΤΡΟΥ';
COMMENT ON COLUMN HOSPITAL.DOCTORS.DEPT_KOD IS 'ΚΩΔΙΚΟΣ ΤΜΗΜΑΤΟΣ ΣΤΟ ΟΠΟΙΟ ΕΡΓΑΖΕΤΑΙ';
COMMENT ON COLUMN HOSPITAL.DOCTORS.DOCT_NAME IS 'ΟΝΟΜΑ ΓΙΑΤΡΟΥ';
COMMENT ON COLUMN HOSPITAL.DOCTORS.DOCT_SURNAME IS 'ΕΠΙΘΕΤΟ ΓΙΑΤΡΟΥ';
COMMENT ON COLUMN HOSPITAL.DOCTORS.DOCT_FATHERNAME IS 'ΟΝΟΜΑ ΠΑΤΕΡΑ';
```

```
CREATE UNIQUE INDEX HOSPITAL.PK_DOCTORS ON HOSPITAL.DOCTORS
(DOCT_KOD)
LOGGING
TABLESPACE USERS
PCTFREE 10
INITRANS 2
MAXTRANS 255
STORAGE (
    INITIAL 64K
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    PCTINCREASE 0
    BUFFER_POOL DEFAULT
)
NOPARALLEL;
```

```
ALTER TABLE HOSPITAL.DOCTORS ADD (
    CONSTRAINT PK_DOCTORS
    PRIMARY KEY
    (DOCT_KOD)
    USING INDEX
    TABLESPACE USERS
    PCTFREE 10
    INITRANS 2
    MAXTRANS 255
    STORAGE (
        INITIAL 64K
        MINEXTENTS 1
        MAXEXTENTS UNLIMITED
        PCTINCREASE 0
    )
);
```

```
ALTER TABLE HOSPITAL.DOCTORS ADD (
    CONSTRAINT FK_DOCT_DEPT
    FOREIGN KEY (DEPT_KOD)
    REFERENCES HOSPITAL.DEPARTMENTS (DEPT_KOD),
    CONSTRAINT FK_DOCT_EIDIK
    FOREIGN KEY (EIDIK_KOD)
    REFERENCES HOSPITAL.EIDIKOTHTES (EIDIK_KOD),
    CONSTRAINT DOCTOR_KOD_FK
    FOREIGN KEY (DOCT_KOD)
    REFERENCES SQ_HOSPITAL.SQ_USERS (USR));
```

#### **4) ΠΙΝΑΚΑΣ ΑΣΘΕΝΗΣ (HOSPITAL.PATIENTS)**

```
CREATE TABLE HOSPITAL.PATIENTS
(
    PAT_KOD VARCHAR2(10 BYTE) NOT NULL,
    PAT_NAME VARCHAR2(20 BYTE) NOT NULL,
```



```

PAT_SURNAME  VARCHAR2(50 BYTE)          NOT NULL,
PAT_FATHER   VARCHAR2(20 BYTE),
PAT_MOTHER   VARCHAR2(20 BYTE),
PAT_BIRTHDATE DATE,
PAT_ADDRESS  VARCHAR2(150 BYTE),
PAT_TAXKOD   VARCHAR2(5 BYTE),
PAT_PHONE1   NUMBER,
PAT_PHONE2   NUMBER,
PAT_TAMEIO   VARCHAR2(50 BYTE),
PAT_AFM      VARCHAR2(9 BYTE),
PAT_ARMHTROOY VARCHAR2(15 BYTE),
PAT_EMAIL    VARCHAR2(100 BYTE),
PAT_AMKA     VARCHAR2(20 BYTE)
)
TABLESPACE USERS
PCTUSED 0
PCTFREE 10
INITRANS 1
MAXTRANS 255
STORAGE (
    INITIAL 64K
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    PCTINCREASE 0
    BUFFER_POOL DEFAULT
)
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;

COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_KOD IS 'ΚΩΔΙΚΟΣ ΑΣΘΕΝΟΥΣ';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_NAME IS 'ΟΝΟΜΑ ΑΣΘΕΝΟΥΣ';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_SURNAME IS 'ΕΠΙΘΕΤΟ ΑΣΘΕΝΟΥΣ';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_FATHER IS 'ΟΝΟΜΑ ΠΑΤΕΡΑ';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_MOTHER IS 'ΟΝΟΜΑ ΜΗΤΕΡΑΣ';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_BIRTHDATE IS 'ΗΜ.ΓΕΝΝΗΣΗΣ';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_ADDRESS IS 'ΔΙΕΥΘΥΝΣΗ';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_TAXKOD IS 'ΤΑΧ.ΚΩΔ.';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_PHONE1 IS 'ΤΗΛΕΦΩΝΟ 1';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_PHONE2 IS 'ΤΗΛΕΦΩΝΟ 2';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_TAMEIO IS 'ΤΑΜΕΙΟ ΑΣΘΕΝΟΥΣ';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_AFM IS 'ΑΦΜ';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_ARMHTROOY IS 'ΑΡ.ΜΗΤΡΩΟΥ ΤΑΜΕΙΟΥ';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_EMAIL IS 'E-MAIL';
COMMENT ON COLUMN HOSPITAL.PATIENTS.PAT_AMKA IS 'ΑΡ.ΜΗΤΡΩΟΥ ΚΟΙΝΩΝΙΚΗΣ ΑΣΦΑΛΙΣΗΣ';

CREATE UNIQUE INDEX HOSPITAL.PK_PATIENTS ON HOSPITAL.PATIENTS
(PAT_KOD)
LOGGING
TABLESPACE USERS
PCTFREE 10
INITRANS 2
MAXTRANS 255
STORAGE (
    INITIAL 64K
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    PCTINCREASE 0
    BUFFER_POOL DEFAULT
)
NOPARALLEL;

```

```

ALTER TABLE HOSPITAL.PATIENTS ADD (
  CONSTRAINT PK_PATIENTS
  PRIMARY KEY
  (PAT_KOD)
  USING INDEX
  TABLESPACE USERS
  PCTFREE 10
  INITRANS 2
  MAXTRANS 255
  STORAGE (
    INITIAL 64K
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    PCTINCREASE 0
  ));

```

### 5) ΠΙΝΑΚΑΣ ΓΙΑΤΡΟΙ-ΑΣΘΕΝΕΙΣ (HOSPITAL.DOCTORS\_PATIENTS)

```

CREATE TABLE HOSPITAL.DOCTORS_PATIENTS

```

```

(
  DOCT_KOD VARCHAR2(10 BYTE),
  PAT_KOD VARCHAR2(10 BYTE),
  VISIT_DATE DATE
)

```

```

TABLESPACE USERS

```

```

PCTUSED 0

```

```

PCTFREE 10

```

```

INITRANS 1

```

```

MAXTRANS 255

```

```

STORAGE (

```

```

  INITIAL 64K

```

```

  MINEXTENTS 1

```

```

  MAXEXTENTS UNLIMITED

```

```

  PCTINCREASE 0

```

```

  BUFFER_POOL DEFAULT
)

```

```

LOGGING

```

```

NOCOMPRESS

```

```

NOCACHE

```

```

NOPARALLEL

```

```

MONITORING;

```

```

CREATE UNIQUE INDEX HOSPITAL.PK_DOCTORS_PATIENTS ON HOSPITAL.DOCTORS_PATIENTS

```

```

(DOCT_KOD, PAT_KOD)

```

```

LOGGING

```

```

TABLESPACE USERS

```

```

PCTFREE 10

```

```

INITRANS 2

```

```

MAXTRANS 255

```

```

STORAGE (

```

```

  INITIAL 64K

```

```

  MINEXTENTS 1

```

```

  MAXEXTENTS UNLIMITED

```

```

  PCTINCREASE 0

```

```

  BUFFER_POOL DEFAULT
)

```

```

NOPARALLEL;

```

```

ALTER TABLE HOSPITAL.DOCTORS_PATIENTS ADD (

```

```

CONSTRAINT PK_DOCTORS_PATIENTS
PRIMARY KEY
(DOCT_KOD, PAT_KOD)
USING INDEX
TABLESPACE USERS
PCTFREE 10
INITRANS 2
MAXTRANS 255
STORAGE (
    INITIAL      64K
    MINEXTENTS   1
    MAXEXTENTS   UNLIMITED
    PCTINCREASE  0
));

```

```

ALTER TABLE HOSPITAL.DOCTORS_PATIENTS ADD (
    CONSTRAINT FK1_DOCTOR_PATIENTS
    FOREIGN KEY (DOCT_KOD)
    REFERENCES HOSPITAL.DOCTORS (DOCT_KOD),
    CONSTRAINT FK2_DOCTORS_PATIENTS
    FOREIGN KEY (PAT_KOD)
    REFERENCES HOSPITAL.PATIENTS (PAT_KOD));

```

## 6) ΠΙΝΑΚΑΣ ΕΙΔΙΚΟΤΗΤΩΝ ΓΙΑΤΡΩΝ (HOSPITAL.EIDIKOTHTES)

```

CREATE TABLE HOSPITAL.EIDIKOTHTES
(
    EIDIK_KOD  VARCHAR2(10 BYTE)      NOT NULL,
    EIDIK_DESCR VARCHAR2(100 BYTE)    NOT NULL
)
TABLESPACE USERS
PCTUSED 0
PCTFREE 10
INITRANS 1
MAXTRANS 255
STORAGE (
    INITIAL      64K
    MINEXTENTS   1
    MAXEXTENTS   UNLIMITED
    PCTINCREASE  0
    BUFFER_POOL  DEFAULT
)
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;

COMMENT ON COLUMN HOSPITAL.EIDIKOTHTES.EIDIK_KOD IS 'ΚΩΔΙΚΟΣ ΕΙΔΙΚΟΤΗΤΑΣ ΓΙΑΤΡΟΥ';
COMMENT ON COLUMN HOSPITAL.EIDIKOTHTES.EIDIK_DESCR IS 'ΠΕΡΙΓΡΑΦΗ ΕΙΔΙΚΟΤΗΤΑΣ ΓΙΑΤΡΟΥ';

```

```

CREATE UNIQUE INDEX HOSPITAL.PK_EIDIKOTHTES ON HOSPITAL.EIDIKOTHTES
(EIDIK_KOD)
LOGGING
TABLESPACE USERS
PCTFREE 10
INITRANS 2
MAXTRANS 255
STORAGE (
    INITIAL      64K
    MINEXTENTS   1

```

```

        MAXEXTENTS    UNLIMITED
        PCTINCREASE   0
        BUFFER_POOL   DEFAULT
    )
NOPARALLEL;

```

```

ALTER TABLE HOSPITAL.EIDIKOTHTES ADD (
    CONSTRAINT PK_EIDIKOTHTES
    PRIMARY KEY
    (EIDIK_KOD)
    USING INDEX
    TABLESPACE USERS
    PCTFREE 10
    INITRANS 2
    MAXTRANS 255
    STORAGE (
        INITIAL        64K
        MINEXTENTS     1
        MAXEXTENTS     UNLIMITED
        PCTINCREASE    0
    ));

```

## 7) ΠΙΝΑΚΑΣ ΛΟΓΑΡΙΑΣΜΟΣ ΑΣΘΕΝΟΥΣ (HOSPITAL.PATIENT\_ACCOUNT)

```

CREATE TABLE HOSPITAL.PATIENT_ACCOUNT
(
    PAT_ACCOUNT VARCHAR2(30 BYTE)          NOT NULL,
    PAT_KOD     VARCHAR2(10 BYTE)         NOT NULL,
    CHARGE_DATE DATE,
    CHARGE_POSO NUMBER
)
TABLESPACE USERS
PCTUSED 0
PCTFREE 10
INITRANS 1
MAXTRANS 255
STORAGE (
    INITIAL        64K
    MINEXTENTS     1
    MAXEXTENTS     UNLIMITED
    PCTINCREASE    0
    BUFFER_POOL   DEFAULT
)
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;

```

```

COMMENT ON COLUMN HOSPITAL.PATIENT_ACCOUNT.PAT_ACCOUNT IS 'ΚΩΔΙΚΟΣ ΛΟΓΑΡΙΑΣΜΟΥ ΑΣΘΕΝΟΥΣ';
COMMENT ON COLUMN HOSPITAL.PATIENT_ACCOUNT.PAT_KOD IS 'ΚΔΙΚΟΣ ΑΣΘΕΝΟΥΣ';
COMMENT ON COLUMN HOSPITAL.PATIENT_ACCOUNT.CHARGE_DATE IS 'ΗΜΕΡΟΜΗΝΙΑ ΧΡΕΩΣΗΣ';
COMMENT ON COLUMN HOSPITAL.PATIENT_ACCOUNT.CHARGE_POSO IS 'ΠΟΣΟ ΧΡΕΩΣΗΣ';

```

```

CREATE UNIQUE INDEX HOSPITAL.PK_PATIENT_ACCOUNT ON HOSPITAL.PATIENT_ACCOUNT
(PAT_ACCOUNT)
LOGGING
TABLESPACE USERS
PCTFREE 10
INITRANS 2
MAXTRANS 255

```

```

STORAGE (
  INITIAL      64K
  MINEXTENTS  1
  MAXEXTENTS  UNLIMITED
  PCTINCREASE 0
  BUFFER_POOL  DEFAULT
)
NOPARALLEL;

```

```

ALTER TABLE HOSPITAL.PATIENT_ACCOUNT ADD (
  CONSTRAINT PK_PATIENT_ACCOUNT
  PRIMARY KEY
  (PAT_ACCOUNT)
  USING INDEX
  TABLESPACE USERS
  PCTFREE 10
  INITRANS 2
  MAXTRANS 255
  STORAGE (
    INITIAL      64K
    MINEXTENTS  1
    MAXEXTENTS  UNLIMITED
    PCTINCREASE  0
  ));

```

```

ALTER TABLE HOSPITAL.PATIENT_ACCOUNT ADD (
  CONSTRAINT FK_PATACNT_PATIENT
  FOREIGN KEY (PAT_KOD)
  REFERENCES HOSPITAL.PATIENTS (PAT_KOD));

```

## 8) ΠΙΝΑΚΑΣ ΕΝΤΟΛΩΝ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ(HOSPITAL.RAD\_REQUESTS)

```

CREATE TABLE HOSPITAL.RAD_REQUESTS
(
  RAD_ID          NUMBER          NOT NULL,
  PAT_KOD         VARCHAR2(10 BYTE),
  DOCT_KOD        VARCHAR2(10 BYTE),
  RAD_APPOINT_DATE DATE,
  RAD_APPOINT_TIME VARCHAR2(5 BYTE),
  DEPT_KOD        VARCHAR2(10 BYTE),
  RAD_STATUS      VARCHAR2(1 BYTE),
  NOTES           VARCHAR2(2000 BYTE),
  RAD_DOCT_KOD    VARCHAR2(10 BYTE),
  RAD_DATE_CLOSED DATE
)
TABLESPACE USERS
PCTUSED 0
PCTFREE 10
INITRANS 1
MAXTRANS 255
STORAGE (
  INITIAL      64K
  MINEXTENTS  1
  MAXEXTENTS  UNLIMITED
  PCTINCREASE  0
  BUFFER_POOL  DEFAULT
)
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;

```

```

CREATE UNIQUE INDEX HOSPITAL.PK_RAD_REQUESTS ON HOSPITAL.RAD_REQUESTS
(RAD_ID)
LOGGING
TABLESPACE USERS
PCTFREE 10
INITRANS 2
MAXTRANS 255
STORAGE (
    INITIAL 64K
    MINEXTENTS 1
    MAXEXTENTS UNLIMITED
    PCTINCREASE 0
    BUFFER_POOL DEFAULT
)
NOPARALLEL;

```

```

ALTER TABLE HOSPITAL.RAD_REQUESTS ADD (
    CONSTRAINT PK_RAD_REQUESTS
    PRIMARY KEY
    (RAD_ID)
    USING INDEX
    TABLESPACE USERS
    PCTFREE 10
    INITRANS 2
    MAXTRANS 255
    STORAGE (
        INITIAL 64K
        MINEXTENTS 1
        MAXEXTENTS UNLIMITED
        PCTINCREASE 0
    ));

```

```

ALTER TABLE HOSPITAL.RAD_REQUESTS ADD (
    CONSTRAINT FK1_RAD_REQUESTS
    FOREIGN KEY (PAT_KOD)
    REFERENCES HOSPITAL.PATIENTS (PAT_KOD),
    CONSTRAINT FK2_RAD_REQUESTS
    FOREIGN KEY (DOCT_KOD)
    REFERENCES HOSPITAL.DOCTORS (DOCT_KOD),
    CONSTRAINT FK3_RAD_REQUESTS
    FOREIGN KEY (RAD_DOCT_KOD)
    REFERENCES HOSPITAL.DOCTORS (DOCT_KOD));

```

## 9) ΠΙΝΑΚΑΣ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΑΝΑΦΟΡΩΝ (HOSPITAL.RAD\_REPORTS)

```

CREATE TABLE HOSPITAL.RAD_REPORTS
(
    REP_ID NUMBER,
    RAD_ID NUMBER,
    REP_NOTES VARCHAR2(2000 BYTE),
    REP_DATE DATE,
    RAD_PHOTO BLOB
)
TABLESPACE USERS
PCTUSED 0
PCTFREE 10
INITRANS 1
MAXTRANS 255
STORAGE (
    INITIAL 64K

```

```

        MINEXTENTS    1
        MAXEXTENTS    UNLIMITED
        PCTINCREASE   0
        BUFFER_POOL   DEFAULT
    )
    LOGGING
    NOCOMPRESS
    LOB (RAD_PHOTO) STORE AS
    ( TABLESPACE USERS
      ENABLE STORAGE IN ROW
      CHUNK      8192
      PCTVERSION 10
      NOCACHE
      INDEX (
        TABLESPACE USERS
        STORAGE (
          INITIAL      64K
          NEXT         1
          MINEXTENTS   1
          MAXEXTENTS   UNLIMITED
          PCTINCREASE  0
          BUFFER_POOL   DEFAULT
        ))
      STORAGE (
        INITIAL      64K
        MINEXTENTS   1
        MAXEXTENTS   UNLIMITED
        PCTINCREASE  0
        BUFFER_POOL   DEFAULT
      )
    )
    NOCACHE
    NOPARALLEL
    MONITORING;

```

```

CREATE UNIQUE INDEX HOSPITAL.RAD_REPORTS_PK ON HOSPITAL.RAD_REPORTS
(REP_ID)
LOGGING
TABLESPACE USERS
PCTFREE 10
INITRANS 2
MAXTRANS 255
STORAGE (
  INITIAL      64K
  MINEXTENTS   1
  MAXEXTENTS   UNLIMITED
  PCTINCREASE  0
  BUFFER_POOL   DEFAULT
)
NOPARALLEL;

```

```

ALTER TABLE HOSPITAL.RAD_REPORTS ADD (
  CONSTRAINT RAD_REPORTS_PK
  PRIMARY KEY
  (REP_ID)
  USING INDEX
  TABLESPACE USERS
  PCTFREE 10
  INITRANS 2
  MAXTRANS 255
  STORAGE (
    INITIAL      64K

```

```
MINEXTENTS 1
MAXEXTENTS UNLIMITED
PCTINCREASE 0
));
```

```
ALTER TABLE HOSPITAL.RAD_REPORTS ADD (
  CONSTRAINT RAD_REPORTS_FK1
  FOREIGN KEY (RAD_ID)
  REFERENCES HOSPITAL.RAD_REQUESTS (RAD_ID));
```

## 10) ΠΙΝΑΚΑΣ ΦΑΚΕΛΟΥ ΑΣΘΕΝΟΥΣ (HOSPITAL.PATIENT\_RECORDS)

```
CREATE TABLE HOSPITAL.PATIENT_RECORDS
```

```
(
  PAT_REC_ID NUMBER NOT NULL,
  PAT_KOD VARCHAR2(10 BYTE),
  DOCT_KOD VARCHAR2(10 BYTE),
  VISIT_DATE DATE,
  SYMPTOM VARCHAR2(500 BYTE),
  NOTES VARCHAR2(2000 BYTE),
  RAD_ID NUMBER,
  REP_ID NUMBER
)
```

```
TABLESPACE USERS
```

```
PCTUSED 0
```

```
PCTFREE 10
```

```
INITRANS 1
```

```
MAXTRANS 255
```

```
STORAGE (
```

```
  INITIAL 64K
```

```
  MINEXTENTS 1
```

```
  MAXEXTENTS UNLIMITED
```

```
  PCTINCREASE 0
```

```
  BUFFER_POOL DEFAULT
)
```

```
LOGGING
```

```
NOCOMPRESS
```

```
NOCACHE
```

```
NOPARALLEL
```

```
MONITORING;
```

```
COMMENT ON COLUMN HOSPITAL.PATIENT_RECORDS.PAT_REC_ID IS 'ΑΡΙΘΜΟΣ ΕΓΓΡΑΦΗΣ';
```

```
COMMENT ON COLUMN HOSPITAL.PATIENT_RECORDS.PAT_KOD IS 'ΚΩΔΙΚΟΣ ΑΣΘΕΝΟΥΣ';
```

```
COMMENT ON COLUMN HOSPITAL.PATIENT_RECORDS.DOCT_KOD IS 'ΚΩΔΙΚΟΣ ΓΙΑΤΡΟΥ ΠΕΡΙΣΤΑΤΙΚΟΥ';
```

```
COMMENT ON COLUMN HOSPITAL.PATIENT_RECORDS.VISIT_DATE IS 'ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ';
```

```
COMMENT ON COLUMN HOSPITAL.PATIENT_RECORDS.SYMPTOM IS 'ΣΥΜΠΤΩΜΑ';
```

```
COMMENT ON COLUMN HOSPITAL.PATIENT_RECORDS.NOTES IS 'ΣΗΜΕΙΩΣΕΙΣ ΓΙΑΤΡΟΥ';
```

```
COMMENT ON COLUMN HOSPITAL.PATIENT_RECORDS.RAD_ID IS 'PK ΤΗΣ ΕΝΤΟΛΗΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΕΞΕΤΑΣΗΣ';
```

```
COMMENT ON COLUMN HOSPITAL.PATIENT_RECORDS.REP_ID IS 'PK ΤΗΣ ΑΚΤΙΝΟΛΟΓΙΚΗΣ ΑΝΑΦΟΡΑΣ';
```

```
CREATE UNIQUE INDEX HOSPITAL.PK_PATIENT_RECORDS ON HOSPITAL.PATIENT_RECORDS
```

```
(PAT_REC_ID)
```

```
LOGGING
```

```
TABLESPACE USERS
```

```
PCTFREE 10
```

```
INITRANS 2
```

```
MAXTRANS 255
```

```
STORAGE (
```

```
  INITIAL 64K
```

```
  MINEXTENTS 1
```

```
  MAXEXTENTS UNLIMITED
```

```
  PCTINCREASE 0
)
```



```

        BUFFER_POOL    DEFAULT
    )
NOPARALLEL;

```

```

ALTER TABLE HOSPITAL.PATIENT_RECORDS ADD (
  CONSTRAINT PK_PATIENT_RECORDS
  PRIMARY KEY
  (PAT_REC_ID)
  USING INDEX
  TABLESPACE USERS
  PCTFREE 10
  INITRANS 2
  MAXTRANS 255
  STORAGE (
    INITIAL          64K
    MINEXTENTS      1
    MAXEXTENTS      UNLIMITED
    PCTINCREASE     0
  ));

```

```

ALTER TABLE HOSPITAL.PATIENT_RECORDS ADD (
  CONSTRAINT FK_PATREC_DOCT
  FOREIGN KEY (DOCT_KOD)
  REFERENCES HOSPITAL.DOCTORS (DOCT_KOD),
  CONSTRAINT FK_PATREC_PATIENT
  FOREIGN KEY (PAT_KOD)
  REFERENCES HOSPITAL.PATIENTS (PAT_KOD),
  CONSTRAINT FK_PATIENT_RECORDS
  FOREIGN KEY (RAD_ID)
  REFERENCES HOSPITAL.RAD_REQUESTS (RAD_ID),
  CONSTRAINT PATIENT_RECORDS_R01
  FOREIGN KEY (REP_ID)
  REFERENCES HOSPITAL.RAD_REPORTS (REP_ID));

```

## **ΣΧΗΜΑ ΒΑΣΗΣ ΑΣΦΑΛΕΙΑΣ (SECURITY SCHEMA- SQ\_HOSPITAL)**

### **1) ΠΙΝΑΚΑΣ ΧΡΗΣΤΩΝ (SQ\_HOSPITAL.SQ\_USERS)**

```

CREATE TABLE SQ_HOSPITAL.SQ_USERS
(
  USR    VARCHAR2(30 BYTE)          NOT NULL,
  PSW    VARCHAR2(30 BYTE),
  YPHR   VARCHAR2(1000 BYTE),
  REMARKS VARCHAR2(100 BYTE)
)
TABLESPACE SYSTEM
PCTUSED 40
PCTFREE 10
INITRANS 1
MAXTRANS 255
STORAGE (
  INITIAL          400K
  MINEXTENTS      1
  MAXEXTENTS      UNLIMITED
  PCTINCREASE     0
  FREELISTS       1
  FREELIST GROUPS 1
  BUFFER_POOL     DEFAULT
)

```

```
)  
LOGGING  
NOCOMPRESS  
NOCACHE  
NOPARALLEL  
MONITORING;
```

```
CREATE UNIQUE INDEX SQ_HOSPITAL.PK_SQ_USERS ON SQ_HOSPITAL.SQ_USERS  
(USR)
```

```
LOGGING  
TABLESPACE SYSTEM  
PCTFREE 10  
INITRANS 2  
MAXTRANS 255  
STORAGE (  
    INITIAL 400K  
    MINEXTENTS 1  
    MAXEXTENTS UNLIMITED  
    PCTINCREASE 0  
    FREELISTS 1  
    FREELIST GROUPS 1  
    BUFFER_POOL DEFAULT  
)
```

```
NOPARALLEL;
```

```
ALTER TABLE SQ_HOSPITAL.SQ_USERS ADD (  
    CONSTRAINT PK_SQ_USERS  
    PRIMARY KEY  
    (USR)
```

```
    USING INDEX  
    TABLESPACE SYSTEM  
    PCTFREE 10  
    INITRANS 2  
    MAXTRANS 255  
    STORAGE (  
        INITIAL 400K  
        MINEXTENTS 1  
        MAXEXTENTS UNLIMITED  
        PCTINCREASE 0  
        FREELISTS 1  
        FREELIST GROUPS 1  
    ));
```

```
GRANT ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ON COMMIT REFRESH, QUERY REWRITE,  
DEBUG, FLASHBACK ON SQ_HOSPITAL.SQ_USERS TO HOSPITAL;
```

## 2) ΠΙΝΑΚΑΣ ΡΟΛΩΝ (SQ\_HOSPITAL.SQ\_ROLES)

```
CREATE TABLE SQ_HOSPITAL.SQ_ROLES
```

```
(  
    ROLE_KOD VARCHAR2(50 BYTE),  
    ROLE_DESCR VARCHAR2(500 BYTE)  
)
```

```
TABLESPACE SYSTEM  
PCTUSED 40  
PCTFREE 10  
INITRANS 1  
MAXTRANS 255  
STORAGE (  
    INITIAL 64K  
    MINEXTENTS 1
```

```

        MAXEXTENTS    UNLIMITED
        PCTINCREASE   0
        FREELISTS     1
        FREELIST GROUPS 1
        BUFFER_POOL   DEFAULT
    )
    LOGGING
    NOCOMPRESS
    NOCACHE
    NOPARALLEL
    MONITORING;

```

```

CREATE UNIQUE INDEX SQ_HOSPITAL.SQ_ROLES_IDX ON SQ_HOSPITAL.SQ_ROLES
(ROLE_KOD)
LOGGING
TABLESPACE SYSTEM
PCTFREE 10
INITRANS 2
MAXTRANS 255
STORAGE (
    INITIAL        64K
    MINEXTENTS    1
    MAXEXTENTS    UNLIMITED
    PCTINCREASE   0
    FREELISTS     1
    FREELIST GROUPS 1
    BUFFER_POOL   DEFAULT
)
NOPARALLEL;

```

```

ALTER TABLE SQ_HOSPITAL.SQ_ROLES ADD (
    CONSTRAINT SQ_ROLES_IDX
    PRIMARY KEY
    (ROLE_KOD)
    USING INDEX
    TABLESPACE SYSTEM
    PCTFREE 10
    INITRANS 2
    MAXTRANS 255
    STORAGE (
        INITIAL        64K
        MINEXTENTS    1
        MAXEXTENTS    UNLIMITED
        PCTINCREASE   0
        FREELISTS     1
        FREELIST GROUPS 1
    ));

```

### 3) ΠΙΝΑΚΑΣ ΧΡΗΣΤΩΝ-ΡΟΛΩΝ (SQ\_HOSPITAL.USER\_ROLES)

```

CREATE TABLE SQ_HOSPITAL.USER_ROLES
(
    USER_ROLES_ID NUMBER NOT NULL,
    USERNAME VARCHAR2(10 BYTE),
    ROLE_KOD VARCHAR2(50 BYTE)
)
TABLESPACE SYSTEM
PCTUSED 40
PCTFREE 10
INITRANS 1
MAXTRANS 255

```

```

STORAGE (
  INITIAL      64K
  MINEXTENTS  1
  MAXEXTENTS  UNLIMITED
  PCTINCREASE 0
  FREELISTS   1
  FREELIST GROUPS 1
  BUFFER_POOL DEFAULT
)
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;

```

```

CREATE UNIQUE INDEX SQ_HOSPITAL.IDX1_USER_ROLES ON SQ_HOSPITAL.USER_ROLES
(USERNAME, ROLE_KOD)

```

```

LOGGING
TABLESPACE SYSTEM
PCTFREE 10
INITRANS 2
MAXTRANS 255
STORAGE (
  INITIAL      64K
  MINEXTENTS  1
  MAXEXTENTS  UNLIMITED
  PCTINCREASE 0
  FREELISTS   1
  FREELIST GROUPS 1
  BUFFER_POOL DEFAULT
)
NOPARALLEL;

```

```

ALTER TABLE SQ_HOSPITAL.USER_ROLES ADD (
  CONSTRAINT PK_USER_ROLES
  PRIMARY KEY
  (USER_ROLES_ID)
  USING INDEX
  TABLESPACE SYSTEM
  PCTFREE 10
  INITRANS 2
  MAXTRANS 255
  STORAGE (
    INITIAL      64K
    MINEXTENTS  1
    MAXEXTENTS  UNLIMITED
    PCTINCREASE 0
    FREELISTS   1
    FREELIST GROUPS 1
  ));

```

```

ALTER TABLE SQ_HOSPITAL.USER_ROLES ADD (
  CONSTRAINT FK_SQ_USER
  FOREIGN KEY (USERNAME)
  REFERENCES SQ_HOSPITAL.SQ_USERS (USR),
  CONSTRAINT FK_SQ_ROLE
  FOREIGN KEY (ROLE_KOD)
  REFERENCES SQ_HOSPITAL.SQ_ROLES (ROLE_KOD));

```

```

GRANT SELECT ON SQ_HOSPITAL.USER_ROLES TO HOSPITAL;

```

#### 4) ΠΙΝΑΚΑΣ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ (SQ\_HOSPITAL.SQ\_TASKS)

```
CREATE TABLE SQ_HOSPITAL.SQ_TASKS
```

```
(  
  TASK_KOD  VARCHAR2(50 BYTE),  
  TASK_DESCR VARCHAR2(500 BYTE)  
)
```

```
TABLESPACE SYSTEM
```

```
PCTUSED  40
```

```
PCTFREE  10
```

```
INITRANS 1
```

```
MAXTRANS 255
```

```
STORAGE (  
  INITIAL      64K
```

```
  MINEXTENTS  1
```

```
  MAXEXTENTS  UNLIMITED
```

```
  PCTINCREASE 0
```

```
  FREELISTS   1
```

```
  FREELIST_GROUPS 1
```

```
  BUFFER_POOL DEFAULT  
)
```

```
LOGGING
```

```
NOCOMPRESS
```

```
NOCACHE
```

```
NOPARALLEL
```

```
MONITORING;
```

```
CREATE UNIQUE INDEX SQ_HOSPITAL.SQ_TASK_IDX ON SQ_HOSPITAL.SQ_TASKS
```

```
(TASK_KOD)
```

```
LOGGING
```

```
TABLESPACE SYSTEM
```

```
PCTFREE  10
```

```
INITRANS 2
```

```
MAXTRANS 255
```

```
STORAGE (  
  INITIAL      64K
```

```
  MINEXTENTS  1
```

```
  MAXEXTENTS  UNLIMITED
```

```
  PCTINCREASE 0
```

```
  FREELISTS   1
```

```
  FREELIST_GROUPS 1
```

```
  BUFFER_POOL DEFAULT  
)
```

```
NOPARALLEL;
```

```
ALTER TABLE SQ_HOSPITAL.SQ_TASKS ADD (  
  CONSTRAINT SQ_TASK_IDX
```

```
  PRIMARY KEY
```

```
(TASK_KOD)
```

```
  USING INDEX
```

```
  TABLESPACE SYSTEM
```

```
  PCTFREE  10
```

```
  INITRANS 2
```

```
  MAXTRANS 255
```

```
  STORAGE (  
    INITIAL      64K
```

```
    MINEXTENTS  1
```

```
    MAXEXTENTS  UNLIMITED
```

```
    PCTINCREASE 0
```

```
    FREELISTS   1
```

```
    FREELIST_GROUPS 1  
  ));
```

## 5) ΠΙΝΑΚΑΣ ΑΡΜΟΔΙΟΤΗΤΩΝ (SQ\_HOSPITAL.SQ\_AUTHORIZATIONS)

```
CREATE TABLE SQ_HOSPITAL.SQ_AUTHORIZATIONS
```

```
(  
  ACCESS_TYPE CHAR(1 BYTE),  
  DESCRIPTION VARCHAR2(20 BYTE)
```

```
)
```

```
TABLESPACE SYSTEM
```

```
PCTUSED 40
```

```
PCTFREE 10
```

```
INITRANS 1
```

```
MAXTRANS 255
```

```
STORAGE (
```

```
  INITIAL 64K
```

```
  MINEXTENTS 1
```

```
  MAXEXTENTS UNLIMITED
```

```
  PCTINCREASE 0
```

```
  FREELISTS 1
```

```
  FREELIST GROUPS 1
```

```
  BUFFER_POOL DEFAULT
```

```
)
```

```
LOGGING
```

```
NOCOMPRESS
```

```
NOCACHE
```

```
NOPARALLEL
```

```
MONITORING;
```

```
CREATE UNIQUE INDEX SQ_HOSPITAL.SQ_AUTH_IDX ON SQ_HOSPITAL.SQ_AUTHORIZATIONS
```

```
(ACCESS_TYPE)
```

```
LOGGING
```

```
TABLESPACE SYSTEM
```

```
PCTFREE 10
```

```
INITRANS 2
```

```
MAXTRANS 255
```

```
STORAGE (
```

```
  INITIAL 64K
```

```
  MINEXTENTS 1
```

```
  MAXEXTENTS UNLIMITED
```

```
  PCTINCREASE 0
```

```
  FREELISTS 1
```

```
  FREELIST GROUPS 1
```

```
  BUFFER_POOL DEFAULT
```

```
)
```

```
NOPARALLEL;
```

```
ALTER TABLE SQ_HOSPITAL.SQ_AUTHORIZATIONS ADD (
```

```
  CONSTRAINT SQ_AUTH_IDX
```

```
  PRIMARY KEY
```

```
  (ACCESS_TYPE)
```

```
  USING INDEX
```

```
  TABLESPACE SYSTEM
```

```
  PCTFREE 10
```

```
  INITRANS 2
```

```
  MAXTRANS 255
```

```
  STORAGE (
```

```
    INITIAL 64K
```

```
    MINEXTENTS 1
```

```
    MAXEXTENTS UNLIMITED
```

```
    PCTINCREASE 0
```

```
    FREELISTS 1
```

```
FREELIST GROUPS 1
));
```

## 6) ΠΙΝΑΚΑΣ ΠΡΟΣΒΑΣΗΣ ΧΡΗΣΤΩΝ (SQ\_HOSPITAL.SQ\_ACCESS)

```
CREATE TABLE SQ_HOSPITAL.SQ_ACCESS
```

```
(
  SQ_ACCESS_ID NUMBER          NOT NULL,
  ROLE_KOD   VARCHAR2(50 BYTE) NOT NULL,
  TASK_KOD   VARCHAR2(50 BYTE) NOT NULL,
  ACCESS_TYPE CHAR(1 BYTE)     NOT NULL
)
```

```
TABLESPACE SYSTEM
```

```
PCTUSED 40
```

```
PCTFREE 10
```

```
INITRANS 1
```

```
MAXTRANS 255
```

```
STORAGE (
```

```
  INITIAL 64K
```

```
  MINEXTENTS 1
```

```
  MAXEXTENTS UNLIMITED
```

```
  PCTINCREASE 0
```

```
  FREELISTS 1
```

```
  FREELIST GROUPS 1
```

```
  BUFFER_POOL DEFAULT
```

```
)
```

```
LOGGING
```

```
NOCOMPRESS
```

```
NOCACHE
```

```
NOPARALLEL
```

```
MONITORING;
```

```
CREATE UNIQUE INDEX SQ_HOSPITAL.SQ_ACCESS_IDX ON SQ_HOSPITAL.SQ_ACCESS
(ROLE_KOD, TASK_KOD, ACCESS_TYPE)
```

```
LOGGING
```

```
TABLESPACE SYSTEM
```

```
PCTFREE 10
```

```
INITRANS 2
```

```
MAXTRANS 255
```

```
STORAGE (
```

```
  INITIAL 64K
```

```
  MINEXTENTS 1
```

```
  MAXEXTENTS UNLIMITED
```

```
  PCTINCREASE 0
```

```
  FREELISTS 1
```

```
  FREELIST GROUPS 1
```

```
  BUFFER_POOL DEFAULT
```

```
)
```

```
NOPARALLEL;
```

```
CREATE UNIQUE INDEX SQ_HOSPITAL.SQ_ACCESS_PK ON SQ_HOSPITAL.SQ_ACCESS
(SQ_ACCESS_ID)
```

```
LOGGING
```

```
TABLESPACE SYSTEM
```

```
PCTFREE 10
```

```
INITRANS 2
```

```
MAXTRANS 255
```

```
STORAGE (
```

```
  INITIAL 64K
```

```
  MINEXTENTS 1
```

```
  MAXEXTENTS UNLIMITED
```

```
        PCTINCREASE 0
        FREELISTS 1
        FREELIST GROUPS 1
        BUFFER_POOL DEFAULT
    )
NOPARALLEL;
```

```
ALTER TABLE SQ_HOSPITAL.SQ_ACCESS ADD (
    CONSTRAINT SQ_ACCESS_PK
    PRIMARY KEY
    (SQ_ACCESS_ID)
    USING INDEX
    TABLESPACE SYSTEM
    PCTFREE 10
    INITRANS 2
    MAXTRANS 255
    STORAGE (
        INITIAL 64K
        MINEXTENTS 1
        MAXEXTENTS UNLIMITED
        PCTINCREASE 0
        FREELISTS 1
        FREELIST GROUPS 1
    ));
```

```
ALTER TABLE SQ_HOSPITAL.SQ_ACCESS ADD (
    CONSTRAINT FK_ACCESS_TASK
    FOREIGN KEY (TASK_KOD)
    REFERENCES SQ_HOSPITAL.SQ_TASKS (TASK_KOD),
    CONSTRAINT FK_ACCESS_ROLE
    FOREIGN KEY (ROLE_KOD)
    REFERENCES SQ_HOSPITAL.SQ_ROLES (ROLE_KOD),
    CONSTRAINT FK_ACCESS_TYPE
    FOREIGN KEY (ACCESS_TYPE)
    REFERENCES SQ_HOSPITAL.SQ_AUTHORIZATIONS (ACCESS_TYPE));
```

```
GRANT SELECT ON SQ_HOSPITAL.SQ_ACCESS TO HOSPITAL;
```



## ΠΑΡΑΡΤΗΜΑ 2

### ΛΕΙΤΟΥΡΓΙΕΣ ΣΤΗΝ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ HOSPITAL

#### 1) FUNCTION HOSPITAL.set\_appointment\_date

Λειτουργία εύρεσης της μέρας του ραντεβού για συγκεκριμένο γιατρό, εξαιρώντας τα Σαββατοκύριακα και υποθέτοντας ότι κάθε γιατρός έχει το πολύ 10 ραντεβού μέσα στην μέρα.

```
CREATE OR REPLACE FUNCTION HOSPITAL.set_appointment_date(indoct_kod
IN VARCHAR2) RETURN date IS
cursor cur is
select rad_appoint_date app,count(*) cnt
from rad_requests
where rad_doct_kod=indoct_kod
and rad_date_closed is null
and rad_appoint_date=trunc(sysdate)
group by rad_appoint_date
;
vdate date;
BEGIN
  vdate:=null;
  for rec in cur loop
    if rec.cnt=0 then
      if to_char(sysdate,'Day')='Σάββατο ' then
        vdate:= trunc(sysdate)+2;
        return vdate;
      end if;
      if to_char(sysdate,'Day')='Κυριακή ' then
        vdate:= trunc(sysdate)+1;
        return vdate;
      end if;
      vdate:=trunc(sysdate);
      return vdate;
    else
      if rec.cnt < 10 then
        vdate:=rec.app;
      else
        if to_char(rec.app+1,'Day')='Σάββατο ' then
          vdate:=rec.app+3;
        elsif to_char(rec.app+1,'Day')='Κυριακή ' then
          vdate:=rec.app+2;
        else
          vdate:=rec.app+1;
        end if;
      end if;
    end if;
  end loop;
  vdate:=trunc(sysdate);
  RETURN vdate;
END;
```

/

## 2) FUNCTION HOSPITAL.set\_appointment\_doctor

Λειτουργία εύρεσης του λιγότερο απασχολούμενου ακτινολόγου γιατρού ενός τμήματος.

```
CREATE OR REPLACE FUNCTION HOSPITAL.set_appointment_doctor(indept_kod
IN VARCHAR2) RETURN varchar2 IS
cursor cur is
select s.doct_kod from
(
  select a.doct_kod ,count(b.rad_id) acc
  from doctors a, rad_requests b
  where a.DOCT_KOD = b.rad_doct_kod(+)
  and a.dept_kod=indept_kod
  and a.eidik_kod='10'
  group by a.doct_kod
) s
where s.acc in
(select min(count(b.rad_doct_kod))
from doctors a, rad_requests b
where a.doct_kod=b.rad_doct_kod(+)
and a.dept_kod=indept_kod
and a.eidik_kod='10'
group by b.rad_doct_kod
)
;
vkod varchar2(10);
BEGIN
  vkod:=null;
  for rec in cur loop
    vkod:= rec.doct_kod;
  end loop;
  RETURN vkod;
END;
/
```

## 3) FUNCTION HOSPITAL.set\_appointment\_time

Λειτουργία εύρεσης της ώρας του ραντεβού, ενός συγκεκριμένου γιατρού και για συγκεκριμένη μέρα, υποθέτοντας ότι τα ραντεβού ξεκινάνε στις 8:00 π.μ., κάθε ραντεβού διαρκεί το πολύ μισή ώρα και το σύνολο των ραντεβού ενός γιατρού είναι 10 μέσα σε μια μέρα.

```
CREATE OR REPLACE FUNCTION HOSPITAL.set_appointment_time(indoct_kod
IN VARCHAR2,indate date) RETURN varchar2 IS
cursor cur is
select nvl(max(rad_appoint_time),'00:00') app,count(*) cnt
from rad_requests
where rad_doct_kod=indoct_kod
and rad_appoint_date = indate
;
vtime varchar2(10);
vnum number;
```

```

BEGIN
  vtime:=null;
  for rec in cur loop
    if rec.cnt=0 then
      vtime:='08:00';
      return vtime;
    else
      vnum := 8+ (rec.cnt*0.5);
      if rec.cnt < 10 then
        vtime:= lpad(ltrim((replace(to_char(vnum,'99D99'),'50','30'))),5,'0');
        vtime:= replace(vtime,',',':');
      else
        vtime:='08:00';
      end if;
    end if;
  end loop;
  RETURN vtime;
END;
/

```

#### 4) FUNCTION HOSPITAL.find\_aktinologiko

Λειτουργία εύρεσης του ακτινολογικού τμήματος που αντιστοιχεί σε κάποιο τμήμα οποιουδήποτε ιδρύματος της περιφέρειας. Εάν το τμήμα ανήκει κατευθείαν στο γενικό περιφερειακό νοσοκομείο ή σε νοσοκομείο που διαθέτει ακτινολογικό, τότε επιλέγεται το ακτινολογικό τμήμα αυτού, ενώ εάν το τμήμα ανήκει σε ίδρυμα που δεν διαθέτει ακτινολογικό, τότε επιλέγεται το ακτινολογικό του Π.Γ.Ν.

```

CREATE OR REPLACE FUNCTION HOSPITAL.find_aktinologiko(indept_kod IN
VARCHAR2) RETURN VARCHAR2 IS
cursor cur is
select a.idrima_kod,b.idrima_type
from departments a, idrima b
where a.IDRIMA_KOD = b.IDRIMA_KOD(+)
and a.dept_kod=indept_kod
;
vkod varchar2(10);
vcount number;
BEGIN
  vkod:=null;
  for rec in cur loop
    if rec.idrima_type in ('1','3') then
      select max(dept_kod)
      into vkod
      from departments
      where dept_type ='AT'
      and idrima_kod='10';---aktinologiko toy periferieiakou
      return vkod;
    end if;
    if rec.idrima_type='2' then
      select count(*)
      into vcount
      from departments
      where dept_type='AT'
      and idrima_kod=rec.idrima_kod;
      if vcount<>0 then ----to nosokomeio exei aktinologiko

```

```

        select max(dept_kod)
        into vkod
        from departments
        where dept_type='AT'
        and idrima_kod=rec.idrima_kod;
        return vkod;
    else ----den exei diko tou
        select max(dept_kod)
        into vkod
        from departments
        where dept_type='AT'
        and idrima_kod='10';----aktinologiko toy periferiakiou
        return vkod;
    end if;
end if;
end loop;
RETURN vkod;
END;
/

```

## 5) FUNCTION HOSPITAL.dept\_descr

Λειτουργία εύρεσης της περιγραφής ενός τμήματος κάποιου ιδρύματος, ώστε να εμφανίζεται στις οθόνες.

```

CREATE OR REPLACE FUNCTION HOSPITAL.dept_descr(indept_kod IN
VARCHAR2) RETURN varchar2 IS
cursor cur is
select a.dept_title || ' TOY '||b.idrima_descr descr
from departments a, idrima b
where a.idrima_kod=b.idrima_kod
and a.DEPT_KOD =indept_kod
;
vdescr varchar2(200);
BEGIN
    vdescr:="";
    for rec in cur loop
        vdescr:= rec.descr;
    end loop;
    RETURN vdescr;
END;
/

```

## ΛΕΙΤΟΥΡΓΙΕΣ ΣΤΗΝ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ SQ\_HOSPITAL

Λειτουργία που βρίσκει εάν ο χρήστης P\_USR έχει είδος πρόσβασης P\_ACCESS σε συγκεκριμένη λειτουργία P\_TASK.

Επιστρέφεται '1' εάν η πρόσβαση είναι επιτρεπτή και '0' ένα όχι.

```

FUNCTION check_access(P_USR IN VARCHAR2,P_TASK IN VARCHAR2,P_ACCESS
IN VARCHAR2) RETURN NUMBER IS

```

```

i NUMBER:=0;
BEGIN
    IF P_USR='ADMIN' THEN RETURN '1'; END IF;
    BEGIN

```

```
SELECT 1 INTO i
FROM SQ_ACCESS A, USER_ROLES B
WHERE A.ROLE_KOD = B.ROLE_KOD
AND B.USERNAME = P_USR
AND A.TASK_KOD = P_TASK
AND A.ACCESS_TYPE = P_ACCESS;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RETURN 0;
WHEN OTHERS THEN
RETURN 0;
END;
RETURN 1;
END;
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

## ΠΑΡΑΡΤΗΜΑ 3

### ΚΩΔΙΚΑΣ ΤΟΥ FDL ΑΡΧΕΙΟΥ

```
/*  
*  
* FDL definitions generated by MQ Workflow at 2009-06-30, 10:20:41 μμ.  
*  
*/
```

```
CODEPAGE 1253  
FM_RELEASE V3R6 0
```

```
/*  
* STRUCTUREs  
*/
```

```
STRUCTURE 'LoginData'  
  'userkod': STRING;  
  'password': STRING;  
END 'LoginData'
```

```
STRUCTURE 'LoginDataOut'  
  'userkod': STRING;  
  'departmentkod': STRING;  
  'patientkod': STRING  
  DESCRIPTION "";  
END 'LoginDataOut'
```

```
STRUCTURE 'DOCTOR_DS'  
  'doct_kod': STRING;  
  'doct_name': STRING;  
  'doct_surname': STRING;  
END 'DOCTOR_DS'
```

```
STRUCTURE 'department'  
  'dept_kod': STRING;  
  'dept_name': STRING;  
  'dept_type': STRING;  
  'idrima_kod': STRING;  
  'idrima_descr': STRING;  
END 'department'
```

```
STRUCTURE 'patient_data'  
  'patient_kod': STRING;  
  'patient_name': STRING;  
  'patient_surname': STRING;  
  'patient_father': STRING;  
  'patient_address': STRING;  
  'patient_email': STRING;  
END 'patient_data'
```

```
STRUCTURE 'patient_account'  
  'charge_poso': LONG  
  DESCRIPTION "";  
  'charge_date': STRING  
  DESCRIPTION "";  
  'rad_id': LONG  
  DESCRIPTION "";  
  'pat_kod': STRING
```

```

        DESCRIPTION "";
        'pat_account': STRING
        DESCRIPTION "";
    END 'patient_account'

STRUCTURE 'radissueDS'
    'rad_id': LONG
    DESCRIPTION "";
    'DOCTOR_DS': 'DOCTOR_DS'
    DESCRIPTION "";
    'department': 'department'
    DESCRIPTION "";
    'patient_data': 'patient_data'
    DESCRIPTION "";
    'rad_doctor_kod': 'DOCTOR_DS'
    DESCRIPTION "";
    'exam_department': 'department'
    DESCRIPTION "";
    'appointment_date': STRING;
    'appointment_time': STRING;
END 'radissueDS'

STRUCTURE 'radreport'
    'rep_id': LONG;
    'radrequest': 'radissueDS';
    'rep_notes': STRING;
    'rep_photo': STRING;
    'rep_date': STRING;
END 'radreport'

/*
 * PROGRAMS
 */
PROGRAM 'issueradrequest' ( 'radissueDS', 'radissueDS' )
    DESCRIPTION ""
    NO STRUCTURES_FROM_ACTIVITY
    NOT UNATTENDED
    WINNT EXE PATH_AND_FILENAME "C:\mywork\issueradrequest.cmd"
    WORKING_DIRECTORY "C:\mywork"
    INHERIT ENVIRONMENT
    STYLE VISIBLE
    START FOREGROUND
END 'issueradrequest'

PROGRAM 'sq_login' ( 'LoginData', 'LoginDataOut' )
    DESCRIPTION ""
    NO STRUCTURES_FROM_ACTIVITY
    NOT UNATTENDED
    WINNT EXE PATH_AND_FILENAME "C:\mywork\sq_login.cmd"
    WORKING_DIRECTORY "C:\mywork"
    INHERIT ENVIRONMENT
    STYLE VISIBLE
    START FOREGROUND
END 'sq_login'

PROGRAM 'check_pending' ( 'radissueDS', 'radissueDS' )
    DESCRIPTION ""
    NO STRUCTURES_FROM_ACTIVITY
    NOT UNATTENDED
    WINNT EXE PATH_AND_FILENAME "C:\mywork\checkpending.cmd"

```

```
WORKING_DIRECTORY "C:\mywork"  
INHERIT ENVIRONMENT  
STYLE VISIBLE  
START FOREGROUND  
END 'check_pending'
```

```
PROGRAM 'schedule_appointment' ( 'radissueDS', 'radissueDS' )  
DESCRIPTION ""  
NO STRUCTURES_FROM_ACTIVITY  
NOT UNATTENDED  
WINNT EXE PATH_AND_FILENAME "C:\mywork\schedule_appointment.cmd"  
WORKING_DIRECTORY "C:\mywork"  
INHERIT ENVIRONMENT  
STYLE VISIBLE  
START FOREGROUND  
END 'schedule_appointment'
```

```
PROGRAM 'search_appointments' ( 'radissueDS', 'radissueDS' )  
DESCRIPTION ""  
NO STRUCTURES_FROM_ACTIVITY  
NOT UNATTENDED  
WINNT EXE PATH_AND_FILENAME "C:\mywork\search_appointments.cmd"  
WORKING_DIRECTORY "C:\mywork"  
INHERIT ENVIRONMENT  
STYLE VISIBLE  
START FOREGROUND  
END 'search_appointments'
```

```
PROGRAM 'notify_patient' ( 'radissueDS', 'Default Data Structure' )  
DESCRIPTION ""  
NO STRUCTURES_FROM_ACTIVITY  
NOT UNATTENDED  
WINNT EXE PATH_AND_FILENAME "C:\mywork\notify_patient.cmd"  
WORKING_DIRECTORY "C:\mywork"  
INHERIT ENVIRONMENT  
STYLE VISIBLE  
START FOREGROUND  
END 'notify_patient'
```

```
PROGRAM 'lookup_rad_requests' ( 'radissueDS', 'Default Data Structure' )  
DESCRIPTION ""  
NO STRUCTURES_FROM_ACTIVITY  
NOT UNATTENDED  
WINNT EXE PATH_AND_FILENAME "C:\mywork\lookup_rad_requests.cmd"  
WORKING_DIRECTORY "C:\mywork"  
INHERIT ENVIRONMENT  
STYLE VISIBLE  
START FOREGROUND  
END 'lookup_rad_requests'
```

```
PROGRAM 'charge_patient_account' ( 'radreport', 'patient_account' )  
DESCRIPTION ""  
NO STRUCTURES_FROM_ACTIVITY  
NOT UNATTENDED  
WINNT EXE PATH_AND_FILENAME "C:\mywork\charge_patient_account.cmd"  
WORKING_DIRECTORY "C:\mywork"  
INHERIT ENVIRONMENT  
STYLE VISIBLE  
START FOREGROUND  
END 'charge_patient_account'
```



```

PROGRAM 'perform_rad_request' ( 'radissueDS', 'radreport' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE_PATH_AND_FILENAME "C:\mywork\perform_rad_request.cmd"
  WORKING_DIRECTORY "C:\mywork"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'perform_rad_request'

PROGRAM 'Interpret_Rad_Results' ( 'radreport', 'radreport' )
  DESCRIPTION ""
  NO STRUCTURES_FROM_ACTIVITY
  NOT UNATTENDED
  WINNT EXE_PATH_AND_FILENAME "C:\mywork\interpret_rad_results.cmd"
  WORKING_DIRECTORY "C:\mywork"
  INHERIT ENVIRONMENT
  STYLE VISIBLE
  START FOREGROUND
END 'Interpret_Rad_Results'

/*
 * PERSONs
 */
PERSON 'U100'
  DESCRIPTION ""
  LAST_NAME "ΠΑΠΑΔΟΠΟΥΛΟΣ"
  FIRST_NAME "ΔΗΜΗΤΡΗΣ"
  AUTHORIZED_FOR_PROCESS_CATEGORY 'RAD_REQUESTS' AS_ADMINISTRATOR
  PASSWORD %B29FF3474D17686D40569863B9CA79BB
  IS_NOT_ABSENT
  DO NOT RESET_ABSENT
  GROUP 'FMCGRP'
  SYSTEM 'FMCSYS'
END 'U100'

PERSON 'U200'
  DESCRIPTION ""
  LAST_NAME "ΚΑΛΟΓΕΡΟΠΟΥΛΟΥ"
  FIRST_NAME "ΕΙΡΗΝΗ"
  AUTHORIZED_FOR_PROCESS_CATEGORY 'RAD_REQUESTS' AS_ADMINISTRATOR
  AUTHORIZED_FOR_WORKITEMS_OF ALL
  PASSWORD %CAD7B6ECC1C86C2772D0FE04EA1E5854
  IS_NOT_ABSENT
  DO NOT RESET_ABSENT
  GROUP 'FMCGRP'
  SYSTEM 'FMCSYS'
END 'U200'

PERSON 'U300'
  DESCRIPTION ""
  LAST_NAME "ΙΣΚΙΟΥΠΗ"
  FIRST_NAME "ΚΑΤΕΡΙΝΑ"
  AUTHORIZED_FOR_PROCESS_CATEGORY 'RAD_REQUESTS' AS_ADMINISTRATOR
  PASSWORD %70AE54E70A3214450C1E714C021C37CE
  IS_NOT_ABSENT
  DO NOT RESET_ABSENT
  GROUP 'FMCGRP'

```

```
SYSTEM 'FMCSYS'  
END 'U300'
```

```
PERSON 'U400'  
DESCRIPTION ''''  
LAST_NAME "ΔΗΜΗΤΡΙΟΥ"  
FIRST_NAME "ΑΝΤΩΝΗΣ"  
AUTHORIZED_FOR PROCESS_CATEGORY 'RAD_REQUESTS' AS_ADMINISTRATOR  
PASSWORD %82086A2E081512C14F0F7B940E85E00D  
IS_NOT_ABSENT  
DO NOT RESET_ABSENT  
GROUP 'FMCGRP'  
SYSTEM 'FMCSYS'  
END 'U400'
```

```
PERSON 'U500'  
DESCRIPTION ''''  
LAST_NAME "ΝΙΚΟΛΑΟΥ"  
FIRST_NAME "ΝΙΚΟΣ"  
AUTHORIZED_FOR PROCESS_CATEGORY 'RAD_REQUESTS' AS_ADMINISTRATOR  
PASSWORD %FDFBB09024196E1C6382BC8E6B8C90C7  
IS_NOT_ABSENT  
DO NOT RESET_ABSENT  
GROUP 'FMCGRP'  
SYSTEM 'FMCSYS'  
END 'U500'
```

```
PERSON 'U600'  
DESCRIPTION ''''  
LAST_NAME "ΚΟΡΟΛΙΟΣ"  
FIRST_NAME "ΑΡΙΣΤΕΙΔΗΣ"  
AUTHORIZED_FOR PROCESS_CATEGORY 'RAD_REQUESTS' AS_ADMINISTRATOR  
PASSWORD %4BF5CAB6429D127143675F70FE9759EC  
IS_NOT_ABSENT  
DO NOT RESET_ABSENT  
GROUP 'FMCGRP'  
SYSTEM 'FMCSYS'  
END 'U600'
```

```
PERSON 'DESPINA'  
DESCRIPTION ''''  
AUTHORIZED_FOR PROCESS_CATEGORY 'RAD_REQUESTS' AS_ADMINISTRATOR  
AUTHORIZED_FOR WORKITEMS_OF ALL  
SUBSTITUTE 'U200'  
PASSWORD %D88CA9F8A0C0ACEA9C67B3C655FC47FD  
IS_NOT_ABSENT  
DO NOT RESET_ABSENT  
GROUP 'FMCGRP'  
SYSTEM 'FMCSYS'  
END 'DESPINA'
```

```
/*  
* ROLES  
*/
```

```
ROLE 'RADIOLOGY_DOCTORS'  
DESCRIPTION ''''  
RELATED_PERSON 'U400' 'U500' 'U600'  
END 'RADIOLOGY_DOCTORS'
```

```
ROLE 'DOCTORS'
```

```

DESCRIPTION ""
RELATED_PERSON 'DESPINA' 'U200' 'U300'
END 'DOCTORS'

ROLE 'GRAMMATEIA'
DESCRIPTION ""
RELATED_PERSON 'U100'
END 'GRAMMATEIA'

/*
 * ORGANIZATIONs
 */
ORGANIZATION 'HOSPITAL'
DESCRIPTION ""
MANAGER 'ADMIN'
RELATED_PERSON 'DESPINA' 'U100' 'U200' 'U300' 'U400' 'U500' 'U600'
END 'HOSPITAL'

/*
 * PROCESSES
 */
/*
 * PROCESS RAD_PROCESS
 */
PROCESS 'RAD_PROCESS' ( 'radissueDS', 'Default Data Structure' )
DESCRIPTION "Περιλαμβάνει όλες τις υπομέρους διαδικασίες του συστήματος"
CATEGORY 'RAD_REQUESTS'
DO NOT PROMPT_AT_PROCESS_START
WINDOW VIEW_REFERENCE_POINT XPOS -37 YPOS 650
WINDOW SHOW ALL CONNECTORS
WINDOW SHOW NO CONDITIONS
PROGRAM_ACTIVITY 'Charge_Patient_Account' ( 'radreport', 'patient_account' )
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC WHEN "_RC=1" XPOS 700 YPOS 252
LAYOUT XPOS 700 YPOS 400
NAME_POSITION XPOS 700 YPOS 325
PRIORITY DEFINED_IN INPUT_CONTAINER
DONE_BY MEMBER OF ROLE 'GRAMMATEIA'
PROGRAM 'charge_patient_account'
SYNCHRONIZATION NESTED
END 'Charge_Patient_Account'
PROGRAM_ACTIVITY 'Check_Pending_Requests' ( 'radissueDS', 'radissueDS' )
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC WHEN "_RC=1" XPOS -200 YPOS 1152
LAYOUT XPOS -200 YPOS 1300
NAME_POSITION XPOS -200 YPOS 1225
PRIORITY DEFINED_IN INPUT_CONTAINER
DONE_BY MEMBER OF ROLE 'GRAMMATEIA'
PROGRAM 'check_pending'
SYNCHRONIZATION NESTED
END 'Check_Pending_Requests'
PROGRAM_ACTIVITY 'InterpretRadResults' ( 'radreport', 'radreport' )
START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
EXIT AUTOMATIC WHEN "_RC=1" XPOS 400 YPOS -48
LAYOUT XPOS 400 YPOS 100
NAME_POSITION XPOS 400 YPOS 25
PRIORITY DEFINED_IN INPUT_CONTAINER
DONE_BY STARTER_OF_ACTIVITY "IssueRadRequest"
PROGRAM 'Interpret_Rad_Results'
SYNCHRONIZATION NESTED

```

```

END 'InterpretRadResults'
PROGRAM_ACTIVITY 'IssueRadRequest' ( 'radissueDS', 'radissueDS' )
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC WHEN "_RC=1" XPOS -700 YPOS 1152
  LAYOUT XPOS -700 YPOS 1300
  NAME_POSITION XPOS -700 YPOS 1225
  PRIORITY DEFINED_IN INPUT_CONTAINER
  DONE_BY MEMBER OF ROLE 'DOCTORS'
  PROGRAM 'issueradrequest'
  SYNCHRONIZATION NESTED
END 'IssueRadRequest'
PROGRAM_ACTIVITY 'Lookup_Rad_Requests' ( 'radissueDS', 'Default Data Structure' )
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT MANUAL
  LAYOUT XPOS -700 YPOS 800
  NAME_POSITION XPOS -700 YPOS 725
  PRIORITY DEFINED_IN INPUT_CONTAINER
  DONE_BY STARTER_OF_ACTIVITY "IssueRadRequest"
  PROGRAM 'lookup_rad_requests'
  SYNCHRONIZATION NESTED
END 'Lookup_Rad_Requests'
PROGRAM_ACTIVITY 'Notify_patient' ( 'radissueDS', 'Default Data Structure' )
  START AUTOMATIC WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC WHEN "_RC=1" XPOS 0 YPOS 752
  LAYOUT XPOS 0 YPOS 900
  NAME_POSITION XPOS 0 YPOS 825
  PRIORITY DEFINED_IN INPUT_CONTAINER
  DONE_BY STARTER_OF_ACTIVITY "Schedule_Rad_Request"
  PROGRAM 'notify_patient'
  SYNCHRONIZATION NESTED
END 'Notify_patient'
PROGRAM_ACTIVITY 'Perform_Rad_Request' ( 'radissueDS', 'radreport' )
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC WHEN "_RC=1" XPOS 400 YPOS 452
  LAYOUT XPOS 400 YPOS 600
  NAME_POSITION XPOS 400 YPOS 525
  PRIORITY DEFINED_IN INPUT_CONTAINER
  DONE_BY MEMBER OF ROLE 'RADIOLOGY_DOCTORS'
  PROGRAM 'perform_rad_request'
  SYNCHRONIZATION NESTED
END 'Perform_Rad_Request'
PROGRAM_ACTIVITY 'Schedule_Rad_Request' ( 'radissueDS', 'radissueDS' )
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC WHEN "_RC=1" XPOS 400 YPOS 1152
  LAYOUT XPOS 400 YPOS 1300
  NAME_POSITION XPOS 400 YPOS 1225
  PRIORITY DEFINED_IN INPUT_CONTAINER
  DONE_BY STARTER_OF_ACTIVITY "Check_Pending_Requests"
  PROGRAM 'schedule_appointment'
  SYNCHRONIZATION NESTED
END 'Schedule_Rad_Request'
PROGRAM_ACTIVITY 'Search_Appointments' ( 'radissueDS', 'radissueDS' )
  START MANUAL WHEN AT_LEAST_ONE CONNECTOR TRUE
  EXIT AUTOMATIC WHEN "_RC=1" XPOS 700 YPOS 752
  LAYOUT XPOS 700 YPOS 900
  NAME_POSITION XPOS 700 YPOS 825
  PRIORITY DEFINED_IN INPUT_CONTAINER
  DONE_BY MEMBER OF ROLE 'RADIOLOGY_DOCTORS'
  PROGRAM 'search_appointments'
  SYNCHRONIZATION NESTED

```

```

END 'Search_Appointments'
CONTROL
  FROM 'IssueRadRequest' TO 'Check_Pending_Requests'
CONTROL
  FROM 'IssueRadRequest' TO 'Lookup_Rad_Requests'
LAYOUT
  XPOS -700 YPOS 1100
CONTROL
  FROM 'Check_Pending_Requests' TO 'Schedule_Rad_Request'
CONTROL
  FROM 'Schedule_Rad_Request' TO 'Search_Appointments'
CONTROL
  FROM 'Schedule_Rad_Request' TO 'Notify_patient'
CONTROL
  FROM 'Schedule_Rad_Request' TO 'Perform_Rad_Request'
LAYOUT
  XPOS 400 YPOS 800
CONTROL
  FROM 'Perform_Rad_Request' TO 'Charge_Patient_Account'
CONTROL
  FROM 'Perform_Rad_Request' TO 'InterpretRadResults'
DATA
  FROM 'Check_Pending_Requests' TO 'Schedule_Rad_Request'
  MAP '_STRUCT' TO '_STRUCT'
LAYOUT
  XPOS -100 YPOS 1400
DATA
  FROM 'IssueRadRequest' TO 'Lookup_Rad_Requests'
  MAP '_STRUCT' TO '_STRUCT'
LAYOUT
  XPOS -600 YPOS 1100
DATA
  FROM 'Schedule_Rad_Request' TO 'Search_Appointments'
  MAP '_STRUCT' TO '_STRUCT'
LAYOUT
  XPOS 700 YPOS 1300
DATA
  FROM 'Schedule_Rad_Request' TO 'Notify_patient'
  MAP '_STRUCT' TO '_STRUCT'
LAYOUT
  XPOS 100 YPOS 1200
  XPOS 0 YPOS 1000
DATA
  FROM 'IssueRadRequest' TO 'Check_Pending_Requests'
  MAP '_STRUCT' TO '_STRUCT'
LAYOUT
  XPOS -600 YPOS 1400
  XPOS -600 YPOS 1400
DATA
  FROM 'Perform_Rad_Request' TO 'Charge_Patient_Account'
  MAP '_STRUCT' TO '_STRUCT'
LAYOUT
  XPOS 600 YPOS 700
DATA
  FROM 'Perform_Rad_Request' TO 'InterpretRadResults'
  MAP '_STRUCT' TO '_STRUCT'
LAYOUT
  XPOS 100 YPOS 400
DATA
  FROM 'Schedule_Rad_Request' TO 'Perform_Rad_Request'

```

```
MAP '_STRUCT' TO '_STRUCT'  
LAYOUT  
  XPOS 500 YPOS 900  
END 'RAD_PROCESS'
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ