



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ**

«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός νοσηλευτικών ιδρυμάτων».

Ιωάννα Ζορμπά

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
Υπεύθυνος
Ευάγγελος Σακκόπουλος
Επικ. Καθ. ΠΒ 407/80**

Λαμία, 2011

**«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»**

Ιωάννα Ζορμπά

ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ
ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ
ΒΙΒΛΙΟΘΗΚΗ



ΠΑΝΕΠΙΣΤΗΜΙΟ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ
ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ
9214
21/5/2012

**Πανεπιστήμιο Στερεάς Ελλάδας
Τμήμα Πληροφορική με Εφαρμογές στη Βιοϊατρική**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός νοσηλευτικών ιδρυμάτων».

ΙΩΑΝΝΑ ΖΟΡΜΠΙΑ
Α.Μ. ΠΒ0012

Τριμελής Επιτροπή

Μιχαήλ Βασιλακόπουλος

Ηλίας Μαγκλογιάννης

Ευάγγελος Σακκόπουλος

Αναπλ. Καθηγητής

Επικ. Καθηγητής

Επικ. Καθ. ΠΔ 407/80

Λαμία, Οκτώβριος 2011



**«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»**

Ιωάννα Ζορμπά

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τους διδάσκοντες του τμήματος, με τους οποίους μαζί ξεκινήσαμε στην αρχή της δημιουργίας του συγκεκριμένου τμήματος, για την προσπάθεια που κατέβαλαν ώστε να διδάξουν σε ένα νεοσύστατο και πρωτόπορο τμήμα. Ιδιαίτερος θα ήθελα να ευχαριστήσω τον καθ. κ. Χρ. Κόκκινο ο οποίος ήταν πάντα εκεί για να μας καθοδηγεί όχι μόνο με τις Πανεπιστημιακές του γνώσεις, αλλά ακόμα και με τις πατρικές του συμβουλές.

Ένα μεγάλο ευχαριστώ στον κ. Ευαγ. Σακκόπουλο, ο οποίος με εμπιστεύθηκε επιλέγοντάς με να εκπονήσω τη συγκεκριμένη εργασία.

Ευχαριστώ από καρδίας την οικογένεια μου για την αγάπη και τη συμπαράσταση τους που μου δίνουν ανελλιπώς, τους φίλους μου και τέλος τον Ανδρέα για την κατανόηση του κατά τη διάρκεια εκπόνησης της εργασίας.

Ιωάννα Ζορμπά

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας Πτυχιακής Εργασίας ήταν να δημιουργηθεί μια εφαρμογή, η οποία θα είναι προσαρμοσμένη σε κινητά και φορητές συσκευές (PDA's), με στόχο τη χρησιμοποίηση της από το Προσωπικό Υγείας εντός των Νοσηλευτικών Ιδρυμάτων. Η εφαρμογή αυτή είναι σχεδιασμένη έτσι ώστε να απευθύνεται σε Γιατρούς και Νοσηλευτές, σε όλους τους χώρους του Νοσηλευτικού Ιδρύματος, είτε αν εκείνη τη στιγμή βρίσκονται συνδεδεμένοι σε κάποιο ασύρματο δίκτυο είτε όχι.

Συγκεκριμένα οι λειτουργίες που θα μπορούν να χρησιμοποιούν οι Γιατροί είναι: να μπορούν να ενημερωθούν για το ιστορικό του ασθενούς, όπως είναι οι διαγνώσεις, η φαρμακευτική αγωγή που ακολουθείται και αν έχει συγκεκριμένες αλλεργίες. Μια άλλη λειτουργία για τους γιατρούς είναι να μπορούν να ενημερωθούν για την πορεία του ασθενή, είτε να μπορούν να αναγνώσουν το θερμομετρικό διάγραμμα. Μια σημαντική λειτουργία είναι ακόμη η άμεση ενημέρωση για τα εργαστηριακά αποτελέσματα. Τέλος μια βασική λειτουργία είναι η αποστολή παραπεμπτικών άμεσα προς το Αιματολογικό, Μικροβιολογικό και Βιοχημικό εργαστήριο.

Από την πλευρά των Νοσηλευτών θα μπορούν να καταγράψουν τις μετρήσεις της θερμοκρασίας του ασθενή και να συμπληρώσουν στον φάκελο οποιαδήποτε πληροφορία θα μπορούσε να ήταν χρήσιμη για τον ιατρό, όπως θα μπορούσε να είναι η αντίδραση του ασθενή στη φαρμακευτική αγωγή.

Μέσω της συγκεκριμένης εφαρμογής εξοικονομείται χρόνος ο οποίος καταναλώνεται είτε στο γραφείο του Ιατρού είτε στο σταθμό Νοσηλευτών για τη συμπλήρωση διαφόρων εγγράφων. Επίσης γίνεται μια καλύτερη καταγραφή δεδομένων προς τον Ηλεκτρονικό Φάκελο Υγείας του ασθενούς. Εκτός όμως από την εξοικονόμηση του χρόνου από πλευράς Ιατρών και Νοσηλευτών, επίσης επιτρέπεται καλύτερος προγραμματισμός στα εργαστήρια, τα οποία ενημερώνονται για τα παραπεμπτικά εγκαίρως και μπορούν να προγραμματίσουν τις εργασίες τους καλύτερα.

Λέξεις Κλειδιά: Ηλεκτρονικός Φάκελος Υγείας, Smartphone, Merge Replication, SQL Server, C#

«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»

Ιωάννα Ζορμπά

Abstract

The purpose of this project was to create an application that is adapted to mobile devices (PDA's), to the use of the health personnel within the Hospitals. This application is designed to intended to doctors and nurses throughout the hospital, either if they are currently connected to a wireless network or not.

Specifically the operations that they might use the Doctors are: to be informed for the history of the patient, as are the diagnoses, medication that is followed and if it has concrete allergies. Another operation for the doctors is to be informed for the course of the patient, or be possible to informed for the temperature diagram. An important operation is to be informed for the laboratorial results. Finally the most basic operation is to send referral to Aimatological, Microbiological and Biochemical laboratories.

From the side of the Nurses they might record the measurements of temperature of the patient and supplement in the file, any information it could be useful for the doctor, as it could be the reaction of patient in the medication.

Via the particular application is economized time which is consumed at the Doctor's office or at the station of Nurses for the completion of various documents. Also becomes a better recording of data to the Electronic Medical Record. Except for the saving of time from side of Doctors and Nurses, it is possible a better planning at the laboratories, which are

Keywords: Electronic Medical Record, Smartphone, Merge Replication, SQL Server, C#

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ	1
1.1 Αντικείμενο πτυχιακής.....	1
1.2 Οργάνωση πτυχιακής.....	1
2. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΕΦΑΡΜΟΓΗΣ	3
2.1 Πληροφοριακά Συστήματα Νοσοκομείου.....	3
2.2 Κωδικοποίηση ασθενειών.....	4
2.2.1 Σύστημα SNOMED.....	5
2.2.2 Σύστημα ICD.....	5
2.3 Λειτουργικές Απαιτήσεις Εφαρμογής.....	5
2.4 Υπάρχοντα συστήματα σε Νοσοκομεία της χώρας.....	6
3. ΓΕΝΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	
3.1 Τεχνικές Συγχρονισμού.....	8
3.1.1 Τεχνική RDA.....	8
3.1.2 Web Services.....	8
3.1.3 Η Τεχνική Merge Replication.....	9
4. ΠΑΡΟΥΣΙΑΣΗ ΤΕΧΝΟΛΟΓΙΩΝ ΣΥΣΤΗΜΑΤΟΣ	13
4.1 Microsoft SQL Server 2008 Developer.....	13
4.2 Microsoft Visual Studio 2008 Professional Edition.....	13
4.3 Windows Mobile 6 Professional SDK v6.0A.....	13
4.4 Internet Information Services (IIS).....	14
4.5 Microsoft SQL Server Compact 3.5 Service Pack 2 Server Tools.....	14
4.6 Windows Mobile Device Center.....	14
4.7 microsoft.NET compact Framework 3.5.....	14
5. ΑΝΑΠΤΥΞΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΤΟΥ SERVER	15
5.1 Microsoft SQL Server 2008 Developer.....	15
5.2 Windows firewall.....	15
5.3 Σύνδεση Internet.....	16
5.4 Επιλογές φακέλων.....	16
5.5 Ενεργοποίηση του IIS.....	17
6. ΑΝΑΠΤΥΞΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΤΟΥ SERVER	18
6.1 Πίνακες της ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	18
6.2 Δημιουργία της ΒΔ.....	20
7. ΛΕΙΤΟΥΡΓΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΔΙΑΧΕΙΡΙΣΗΣ ΔΕΔΟΜΕΝΩΝ	33
7.1 Publication.....	33
7.2 Logins.....	41
7.3 Synchronization.....	42
7.4 SQL server compact subscription.....	48

8. ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΚΙΝΗΤΗΣ ΣΥΣΚΕΥΗΣ	53
8.1 Δημιουργία Project eMedRecordMbl.....	53
8.2 Φόρμες συσκευής.....	55
8.2.1 Αρχική Φόρμα.....	55
8.2.1.1 Η μέθοδος LogMeOut().....	56
8.2.1.2 Η μέθοδος menuItemEx_Click().....	57
8.2.1.3 Η μέθοδος menuItemSun_Click().....	57
8.2.1.4 Η μέθοδος LogMeIn().....	58
8.2.1.5 Η μέθοδος menuItemApos_Click().....	58
8.2.1.6 Η μέθοδος btnEpiloghAsthenh_click().....	58
8.2.1.7 Η μέθοδος buttonApoth_Click.....	59
8.2.2 Φόρμα σύνδεσης.....	59
8.2.2.1 Η μέθοδος menuItem2_click().....	60
8.2.2.2 Η μέθοδος menuItem1_click().....	60
8.2.2.3 Η μέθοδος button1_click().....	62
8.2.3 Φόρμα για την επιλογή ασθενή.....	62
8.2.3.1 Η μέθοδος EpiloghAsthenh_Load().....	63
8.2.3.2 Η μέθοδος cbxClinic_SelectedIndexChanged().....	64
8.2.3.3 Η μέθοδος btnNextFrm_Click().....	66
8.2.4 Η φόρμα ΛΕΙΤΟΥΡΓΙΕΣ.....	66
8.2.4.1 Η μέθοδος leitourgies.....	66
8.2.5 Η Φόρμα ΙΣΤΟΡΙΚΟ.....	67
8.2.6 Η φόρμα ΘΕΡΜΟΜΕΤΡΙΚΟ ΔΙΑΓΡΑΜΜΑ.....	68
8.2.6.1 Η μέθοδος btnUpdate_Click().....	69
8.2.7 Η φόρμα ΕΡΓΑΣΤΗΡΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ.....	70
8.2.7.1 Η μέθοδος btnConfirmation_click().....	71
8.2.8 Η φόρμα ΠΑΡΑΤΗΡΗΣΕΙΣ.....	72
8.2.9 Η φόρμα ΠΑΡΑΠΕΜΠΤΙΚΟ.....	73
8.2.10 Η φόρμα ΠΑΡΑΠΕΜΠΤΙΚΟ ΑΙΜΑΤΟΛΟΓΙΚΟΥ.....	73
8.2.10.1 Αυτόματη χρήση ορίων.....	74
8.2.10.2 Η μέθοδος btnConfirmation_Click().....	75
8.2.11 Η φόρμα ΠΑΡΑΠΕΜΠΤΙΚΟ ΒΙΟΧΗΜΙΚΟΥ-ΜΙΚΡ/ΚΟΥ.....	75
8.2.11.1 Η μέθοδος btnConfirmation_Click.....	76
9. ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΤΕΛΙΚΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΤΩΝ ΛΕΙΤΟΥΡΓΙΩΝ ΤΗΣ	78
ΒΙΒΛΙΟΓΡΑΦΙΑ	81
ΠΑΡΑΡΤΗΜΑ	83

Πίνακας εικόνων

Εικόνα 1: Ζωτικές Τιμές.....	7
Εικόνα 2: Τιμές Σακχάρου.....	7
Εικόνα 3: Ενημέρωση Κεντρικής Βάσης.....	7
Εικόνα 4: Υλοποίηση Merge Replication με ένα Server.....	10
Εικόνα 5: Υλοποίηση Merge Replication με 2 Servers.....	11
Εικόνα 6: Υλοποίηση Merge Replication με πολλαπλούς Servers.....	11
Εικόνα 7: Υλοποίηση Merge Replication με πολλαπλούς Publishers.....	12
Εικόνα 8: SQL Server Configuration Manager.....	15
Εικόνα 9: SQL Server Configuration Manager, Πρωτόκολλα.....	15
Εικόνα 10: Προσθήκη Θύρας SQL Server.....	16
Εικόνα 11: Ρυθμίσεις κοινής χρήσης αρχείων.....	16
Εικόνα 12: Από-επιλογή Οδηγού κοινής χρήσης.....	16
Εικόνα 13: Ενεργοποίηση IIS.....	17
Εικόνα 14: Σύνδεση με SQL Server.....	20
Εικόνα 15: Microsoft SQL Server Management Studio (SSMS), Object Explorer.....	20
Εικόνα 16: SSMS, Δημιουργία Βάσης Δεδομένων.....	21
Εικόνα 17: Δημιουργία Πίνακα PATIENT.....	22
Εικόνα 18: SSMS, Ιδιότητες Στήλης.....	23
Εικόνα 19: SSMS, Template Explorer.....	23
Εικόνα 20: Καθορισμός τιμών στις παραμέτρους.....	24
Εικόνα 21: T-SQL για τη δημιουργία του πίνακα CLINIC.....	24
Εικόνα 22: SSMS, Δημιουργία δύο Primary Keys σε έναν πίνακα.....	30
Εικόνα 23: SSMS, Εμφάνιση δύο Primary Keys σε έναν πίνακα.....	30
Εικόνα 24: SSMS, Relationships.....	31
Εικόνα 25: SSMS, Σχέσεις μεταξύ Primary Keys.....	31
Εικόνα 26: Κοινή χρήση φακέλου Snapshot.....	33
Εικόνα 27: SSMS, Replication.....	34
Εικόνα 28: New Publication Wizard.....	34
Εικόνα 29: Δημιουργία Distributor.....	34

«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»

Ιωάννα Ζορμπά

Εικόνα 30: Δημιουργία Snapshot Φακέλου.....	35
Εικόνα 31: Επιλογή Βάσης Δεδομένων.....	35
Εικόνα 32: Επιλογή Μεθόδου Συγχρονισμού.....	36
Εικόνα 33: Επιλογή εκδόσεων Server για Subscribers.....	36
Εικόνα 34: Επιλογή Πινάκων για Publication.....	37
Εικόνα 35: Ενημέρωση προσθήκης στήλης για το συγχρονισμό.....	37
Εικόνα 36: Φιλτράρισμα δεδομένων.....	38
Εικόνα 37: Επιβεβαίωση Δημιουργίας φακέλου Snapshot.....	38
Εικόνα 38: Επιλογή λογαριασμού SQL Server Agent.....	39
Εικόνα 39: Επιβεβαίωση επιλογής Λογαριασμού.....	39
Εικόνα 40: Επιλογή δημιουργίας της Publication.....	40
Εικόνα 41: Ονομασία Publication.....	40
Εικόνα 42: Επιβεβαίωση δημιουργίας Publication.....	41
Εικόνα 43: εισαγωγή χρηστών για πρόσβαση στην εφαρμογή.....	41
Εικόνα 44: Δικαιώματα σε χρήστες για Βάσεις.....	41
Εικόνα 45: Synchronization Wizard.....	42
Εικόνα 46: Επιλογή SQL Server Compact για Subscriber Type.....	42
Εικόνα 47: Επιλογή Εικονικής Διεύθυνσης.....	43
Εικόνα 48: Εισαγωγή ψευδωνύμου για πρόσβαση στην Εικονική Διεύθυνση.....	43
Εικόνα 49: Εξουσιοδότηση δημιουργίας φακέλου Εικονικής Διεύθυνσης.....	44
Εικόνα 50: Εισαγωγή του .dll αρχείου.....	44
Εικόνα 51: Καθορισμός καναλιού ασφαλούς λειτουργίας SSL.....	44
Εικόνα 52: Επιλογή σύνδεσης χρηστών.....	45
Εικόνα 53: Καθορισμός ανώνυμης πρόσβασης χρηστών.....	45
Εικόνα 54: Καθορισμός μονοπατιού δικτύου και φακέλου.....	46
Εικόνα 55:Επαλήθευση επιλογών για Συγχρονισμό.....	46
Εικόνα 56: Επιβεβαίωση δημιουργίας διαδικασίας Συγχρονισμού.....	47
Εικόνα 57: Απόδοση δικαιωμάτων στον φάκελο SyncHospitaSv.....	47
Εικόνα 58: Ηλεκτρονική Διεύθυνση SyncHospitalSv.....	48

Εικόνα 59: Σύνδεση με SQL Server Compact Edition.....	48
Εικόνα 60: Εμφάνιση του Subscription Wizard.....	49
Εικόνα 61: Επιλογή Server και Publication.....	49
Εικόνα 62 Καθορισμός ονομασίας Subscription.....	50
Εικόνα 63: Καθορισμός URL της Εικονικής Διεύθυνσης.....	50
Εικόνα 64: Επιλογή πιστοποίησης για τον Subscriber.....	51
Εικόνα 65: Επαλήθευση επιλογών του Subscriber.....	51
Εικόνα 66: Επιβεβαίωση επιτυχούς δημιουργίας Subscription.....	52
Εικόνα 67: SSMS, εμφάνιση σύνδεσης Subscription – Publication.....	52
Εικόνα 68: Επιλογή .NET Framework.....	53
Εικόνα 69: Πληροφορίες του project eMedRecordMbl.....	54
Εικόνα 70: Constructor Φόρμας Form1.....	55
Εικόνα 71: Αρχική φόρμα εφαρμογής.....	56
Εικόνα 72: MenuItem's φόρμας.....	56
Εικόνα 73: Φόρμα LogInForm.....	59
Εικόνα 74: Properties αντικειμένου txtpassword.....	60
Εικόνα 75: Εισαγωγή Βάσης Δεδομένων στο project.....	61
Εικόνα 76: Επιλογή πινάκων για τη δημιουργία του PasswordDataSet.....	61
Εικόνα 77: Φόρμα για την επιλογή ασθενή.....	63
Εικόνα 78: Επιλογή τρόπου εμφάνισης αντικειμένων στη φόρμα.....	63
Εικόνα 79: Εμφάνιση αντικειμένων στη φόρμα.....	64
Εικόνα 80: Εισαγωγή Bounds στο cbxClinic.....	64
Εικόνα 81: Γραφική απεικόνιση του patientClinicDataSet.....	65
Εικόνα 82: Φόρμα ΛΕΙΤΟΥΡΓΙΕΣ.....	66
Εικόνα 83: Φόρμα ΙΣΤΟΡΙΚΟ.....	67
Εικόνα 84: Γραφική απεικόνιση historyDataSet.....	68
Εικόνα 85: Φόρμα ΘΕΡΜΟΜΕΤΡΙΚΟ ΔΙΑΓΡΑΜΜΑ.....	69
Εικόνα 86: Γραφική απεικόνιση temperatureDataSet.....	69
Εικόνα 87: Φόρμα ΕΡΓΑΣΤΗΡΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ.....	71
Εικόνα 88: Γραφική απεικόνιση LabResultsDataSet.....	71

«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»

Ιωάννα Ζορμπά

Εικόνα 89: Φόρμα ΠΑΡΑΤΗΡΗΣΕΙΣ.....	72
Εικόνα 90: Φόρμα ΠΑΡΑΠΕΜΠΤΙΚΟ.....	73
Εικόνα 91: Φόρμα ΠΑΡ/ΚΟ ΑΙΜΑΤΟΛΟΓΙΚΟ.....	74
Εικόνα 92: Γραφική απεικόνιση referenceDataSet.....	74
Εικόνα 93: Εισαγωγή bounds σε comboBox εξετάσεων.....	75
Εικόνα 94: Φόρμα ΠΑΡ/ΚΟ ΜΙΚΡΟΒ/ΒΙΟΧ.....	76

Πίνακας Σχεδιαγραμμάτων

Σχ. 1 Πληροφοριακό Σύστημα Νοσοκομείου.....	4
Σχ. 2 Η διαδικασία των Web Services.....	9
Σχ. 3 Η διαδικασία της Merge Replication.....	9
Σχ. 4 Components της τεχνικής Merge Replication.....	10
Σχ. 5 Διάγραμμα Οντοτήτων- Συσχετίσεων της ΒΔ HospitalSv.....	19

Πίνακες

Πιν. 1 Πίνακες Βάσης Δεδομένων.....	18
Πιν. 2 Πίνακας PATIENT.....	21
Πιν. 3 Πίνακας CLINIC.....	23
Πιν. 4 Πίνακας WARD.....	25
Πιν. 5 Πίνακας DOCTOR.....	25
Πιν. 6 Πίνακας NURSE.....	25
Πιν. 7 Πίνακας DRUGLIST.....	26
Πιν. 8 Πίνακας ICD10.....	26
Πιν. 9 Πίνακας ALLERGIES.....	26
Πιν. 10 Πίνακας DIAGNOSIS.....	27
Πιν. 11 Πίνακας EXAMINATIONS.....	27
Πιν. 12 Πίνακας SPECIALITY.....	27
Πιν. 13 Πίνακας SYMPTOMS.....	27
Πιν. 14 Πίνακας MEDICATION.....	28
Πιν. 15 Πίνακας LABRESULTS.....	28
Πιν. 16 Πίνακας REFERENCE.....	29
Πιν. 17 Πίνακας TEMPERATURE DIAGRAM.....	29

**«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»**

Ιωάννα Ζορμπά

1. ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο πτυχιακής

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η δημιουργία εφαρμογής PDA για χρήση από το Ιατρικό και Νοσηλευτικό προσωπικό μιας Νοσοκομειακής Μονάδας κατά την επίσκεψή τους στους θαλάμους όπου και νοσηλεύονται οι ασθενείς.

Σκοπός είναι η πρόσβαση του Προσωπικού Υγείας του Νοσοκομείου στη Βάση Δεδομένων η οποία βρίσκεται σε έναν κεντρικό υπολογιστή (Server) μέσω ενός PDA (Personal Digital Assistant) ή smartphone. Λειτουργίες οι οποίες θα δημιουργηθούν είναι η ανάγνωση του ιστορικού ενός ασθενή όπως διάγνωση, φαρμακευτική αγωγή ή αν έχει συγκεκριμένες αλλεργίες, την ανάγνωση και ανανέωση των ζωτικών σημείων του ασθενή και την αποστολή παραπεμπτικών στα εργαστήρια από τους Ιατρούς.

Αυτό που επιδιώκεται είναι η δημιουργία μια εύχρηστης εφαρμογής η οποία θα είναι ένα βοηθητικό εργαλείο στα χέρια του Προσωπικού Υγείας χωρίς να περιπλανιέται σε δυσανάγνωστα μενού. Να μπορεί ο Ιατρός να ενημερωθεί για λεπτομέρειες που ίσως να τον ενδιαφέρουν τη στιγμή που γίνεται η επίσκεψη στους θαλάμους όπως αν στο παρελθόν, και εφόσον υπάρχουν παλαιότερα δεδομένα, είχε αντιδράσει σε κάποια φαρμακευτική αγωγή ή αν έχει κάποια αλλεργία, ή ακόμα να ενημερωθεί για παλαιότερες διαγνώσεις. Να μπορεί επίσης να στείλει άμεσα παραπεμπτικό στα εργαστήρια τα οποία με τη σειρά τους θα μπορούν έτσι να οργανώσουν καλύτερα το χρόνο τους. Και τέλος να μπορούν να καταγραφούν από τους Νοσηλευτές άμεσα οι ενδείξεις για τα ζωτικά σημεία, χωρίς να χρειάζεται να κάνουν αυτή τη διαδικασία στο σταθμό νοσηλευτών και με αυτό τον τρόπο να περιοριστεί η εργασία γραφείου.

Έτσι δημιουργήθηκε μια κεντρική βάση η οποία θα βρίσκεται στον κεντρικό υπολογιστή (Server) του Νοσοκομείου και μια εφαρμογή για κινητή συσκευή η οποία υποστηρίζει λειτουργικό σύστημα windows Mobile 6. Ο συγχρονισμός των Βάσεων έγινε με την τεχνική της Merge Replication η οποία μαζί με τη τεχνική RDA (Remote Data Access) και τη τεχνική Web Services αποτελούν τεχνικές συγχρονισμού μεταξύ των Η/Υ. Επιλέχθηκε η συγκεκριμένη τεχνική, επειδή αυτό που μας ενδιαφέρει είναι η άμεση ανταλλαγή δεδομένων, αλλά ακόμα και σε περιπτώσεις όπου δεν θα λειτουργεί ή δεν είναι δυνατό να είναι διαθέσιμο το ασύρματο δίκτυο το προσωπικό να μπορεί να επιτελέσει ορισμένες από τις εργασίες.

1.2 Οργάνωση πτυχιακής

Στη συνέχεια ακολουθεί μια σύντομη περιγραφή των κεφαλαίων και τα βήματα που έγιναν για την υλοποίηση της.

Στο κεφάλαιο 2 παρουσιάζονται και εξετάζονται οι λειτουργικές απαιτήσεις για την εφαρμογή καθώς και μια σχετική εφαρμογή. Αναλύονται οι λόγοι για την ανάγκη ύπαρξης μιας τέτοιας εφαρμογής. Ακόμα αναφέρονται συστήματα ταξινόμησης ασθενειών όπως η SNOMED και η ICD10, πως λειτουργεί η κωδικοποίηση και γιατί είναι χρήσιμη η ύπαρξή τους.

Στο κεφάλαιο 3 παρουσιάζεται η γενική αρχιτεκτονική του προτεινόμενου συστήματος.

Στο κεφάλαιο 4 αναφέρονται οι τεχνολογίες και το λογισμικό που χρησιμοποιήθηκαν για τη δημιουργία του συγκεκριμένου project και τυχόν επεκτάσεις που απαιτούνται για τη σωστή λειτουργία τους.

Η δημιουργία της Βάσης Δεδομένων αναλύεται στο κεφάλαιο 5 και στο κεφάλαιο 6 για τις περιπτώσεις της ΒΔ στο κινητό και στον server αντίστοιχα. Αναφέρονται τρόποι δημιουργίας πινάκων μέσω του SSMS (SQL Server Management Studio). Τι είναι ένα Πρωτεύον (Primary key) και τι ένα Ξένο Κλειδί (Foreign key), πως και γιατί τα χρησιμοποιούμε, πως δημιουργούνται σχέσεις 1-N και N-M και γιατί. Ποιοι πίνακες

αποτελούν τη Βάση μας και πως συνδέονται μεταξύ τους.

Στο αμέσως επόμενο κεφάλαιο 7 γίνεται αναφορά για τους τρόπους μεταφοράς δεδομένων από και προς ένα Server από μια κινητή συσκευή. Δίνονται αναλυτικά τα βήματα της Merge Replication, πως δημιουργούμε μια Publication, πως γίνεται ο συγχρονισμός των Βάσεων και πως δημιουργούμε Subscriptions.

Το κεφάλαιο 8 αποτελείται από τη δημιουργία φορμών σε μια κινητή συσκευή με λειτουργικό σύστημα Windows Mobile 6 και τη δημιουργία κώδικα στη γλώσσα προγραμματισμού C#. Χρησιμοποιούνται διάφοροι τρόποι δημιουργίας του γραφικού περιβάλλοντος μιας φόρμας και της επικοινωνίας μεταξύ αντικειμένων. Επίσης όσα αφορούν στην επεξεργασία δεδομένων, δίνονται διαφορετικοί τρόποι χρησιμοποίησης μεθόδων όπως είναι το Insert() και το Fill() σε ένα πίνακα ή σε ένα αντικείμενο αντίστοιχα.

Το τελευταίο κεφάλαιο 9 αποτελεί τη λειτουργική παρουσίαση της εφαρμογής που σχεδιάστηκε και υλοποιήθηκε και αποτελεί ένα οδηγό χρήσης της κινητής συσκευής προς το Προσωπικό Υγείας του Νοσοκομείου.

2. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΕΦΑΡΜΟΓΗΣ

2.1 Πληροφοριακά Συστήματα Νοσοκομείου

Πριν αναπτυχθεί η Ανάλυση Απαιτήσεων της Εφαρμογής, θα ήταν σκόπιμο να αναφερθούν ορισμένα θέματα που αφορούν τα Πληροφοριακά Συστήματα Νοσοκομείου (ΠΣΝ), όπως τι είναι ένα ΠΣΝ, ποια είναι τα κύρια συστατικά του. Επίσης ποιος είναι ο σκοπός του, οι χρήστες και γενικά ποια είναι η δομή του.

Πληροφοριακό Σύστημα Νοσοκομείου (ΠΣΝ)/ Hospital Information System (HIS) είναι ένα ολοκληρωμένο, βοηθούμενο από ηλεκτρονικούς υπολογιστές σύστημα, σχεδιασμένο να αποθηκεύει, διαχειρίζεται και ανακαλεί πληροφορίες σχετιζόμενες με τις διοικητικές και κλινικές προσδοκίες της παροχής φροντίδας μέσα στο Νοσοκομείο. Είναι ένα σύνολο υπολογιστικών συστημάτων και εξοπλισμού τηλεπικοινωνιών, το οποίο είναι σχεδιασμένο ώστε να διαχειρίζεται όλες τις νοσοκομειακές πληροφορίες, ιατρικής και διοικητικής φύσης με στόχο:

- Την υποστήριξη των επαγγελματιών υγείας προκειμένου αυτοί να εκτελούν το έργο τους αποδοτικά και αποτελεσματικά
- Τη βελτίωση της ποιότητας των υπηρεσιών υγείας που παρέχονται στον ασθενή
- Τη μείωση των σχετικών δαπανών [1].

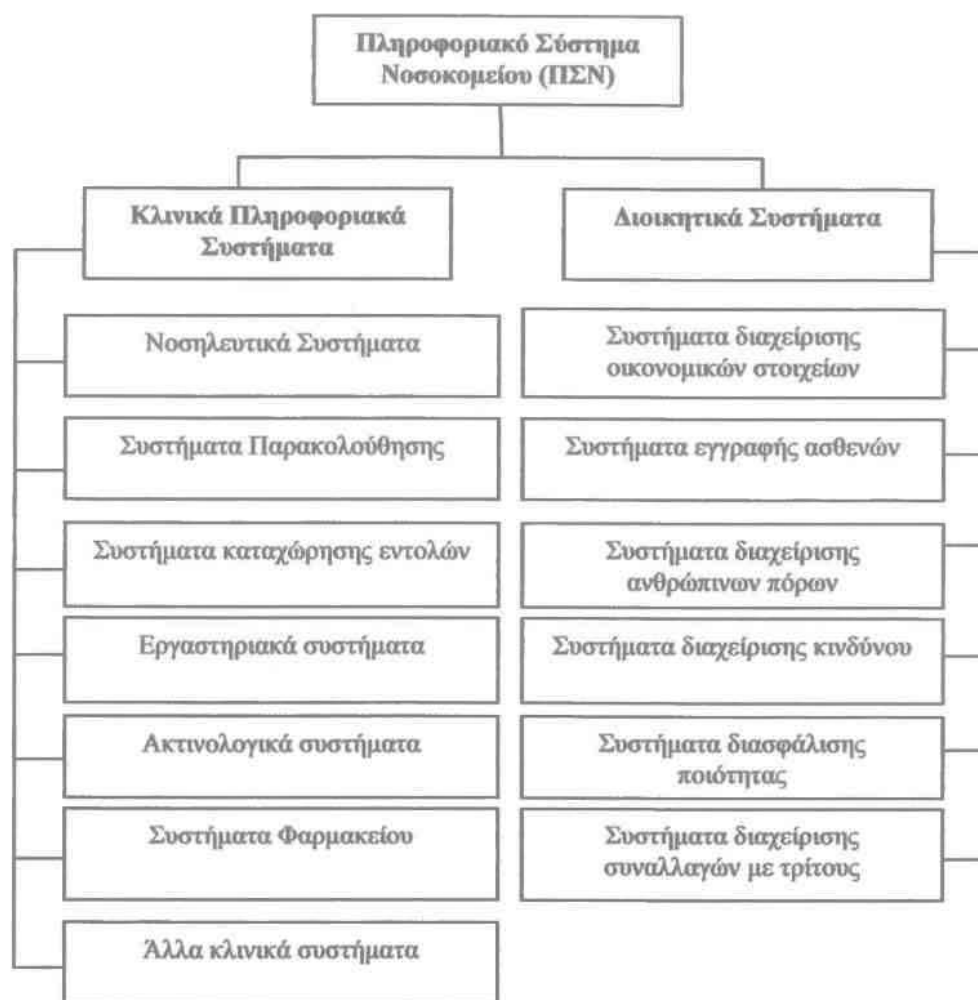
Τα κύρια χαρακτηριστικά ενός ΠΣΝ είναι οι χρήστες, τα δεδομένα, οι διαδικασίες δηλαδή οδηγίες που καθορίζουν τον τρόπο με τον οποίο θα ενεργήσουν οι χρήστες σε συγκεκριμένες καταστάσεις και ο εξοπλισμός.

Κύριοι σκοποί ενός ΠΣΝ είναι η παροχή καλύτερης φροντίδας και η διαχείριση του κόστους. Δευτερεύοντες σκοποί σε σχέση με τη βελτίωση της παροχής φροντίδας είναι η βελτίωση της επικοινωνίας μεταξύ του Προσωπικού, η μείωση του χρόνου αναμονής, η συμβολή στη Λήψη Απόφασης. Ενώ σε σχέση με τη διαχείριση του κόστους είναι η μείωση του χρόνου νοσηλείας, μείωση των διοικητικών εργασιών, η βελτίωση της χρήσης πόρων και η μείωση του κόστους προσωπικού.

Χρήστες ενός ΠΣΝ δεν είναι μόνο το Διοικητικό, Ιατρικό, Νοσηλευτικό και το Τεχνικό προσωπικό αλλά ακόμα και Ασφαλιστικές εταιρείες, Βιομηχανία, Ενώσεις καταναλωτών και Κυβερνητικές υπηρεσίες.

Τύποι ενός ΠΣΝ είναι τα Κλινικά Πληροφοριακά Συστήματα (Clinical Information Systems, CIS) και τα Διοικητικά Πληροφοριακά Συστήματα (Administrative Information Systems, AIS). Στη συγκεκριμένη εφαρμογή θα ασχοληθούμε με τμήμα των Κλινικών Πληροφοριακών Συστημάτων.

Η Δομή ενός ΠΣΝ απεικονίζεται στο παρακάτω σχέδιο (Σχ.1):



Σχ. 1 Πληροφοριακό Σύστημα Νοσοκομείου

Στα Κλινικά συστήματα ΠΣΝ γίνεται σύνδεση μεταξύ των τμημάτων ενός Νοσοκομείου όπως το Παθολογικό, Καρδιολογικό, Χειρουργικό, ΜΕΘ (Μονάδα Εντατικής Θεραπείας) με τα Συστήματα Κλινικής Υποστήριξης όπως είναι το Ακτινολογικό, Βιοχημικό ή το Φαρμακείο. Υπάρχει μεταφορά Κλινικών Δεδομένων, αποτελεσμάτων εξετάσεων και όλων των πληροφοριών που έχουν σχέση με τα ιατρικά ευρήματα και έτσι δημιουργείται ο Ηλεκτρονικός Φάκελος Υγείας(ΗΦΥ) του ασθενούς.

2.2 Κωδικοποίηση ασθενειών

Η ανάγκη αντιμετώπισης προβλημάτων κατά την αποθήκευση και αναπαράσταση της ιατρική πληροφορίας οδήγησε στην ανάπτυξη προτύπων για κλινικά δεδομένα (Κωδικοποιήσεις). Στην κάθε έννοια του χώρου της υγείας αποδίδεται ένας μοναδικός κωδικός, μια βασική-επικρατούσα ορολογία-όνομα που την περιγράφει και συνήθως ένα σύνολο από συναφείς όρους που χρησιμοποιούνται για την περιγραφή της ίδιας έννοιας. Υπάρχουν πολλών ειδών κωδικοποιήσεις, όμως δύο είναι οι επικρατέστερες [2]:

Ονοματολογίες: Αποτελούν ένα σύνολο αποδεκτών ονομάτων. Για παράδειγμα μια ονοματολογία ασθενειών αποτελείται από μια συλλογή κατάλληλων ονομάτων ασθενειών. Παράδειγμα ονοματολογίας αποτελεί η κωδικοποίηση SNOMED.

Ταξινομήσεις: Αποτελούν λογικές ομαδοποιήσεις ομοειδών- συναφών όρων με βάση ένα ή περισσότερα κριτήρια. Παράδειγμα ταξινόμησης αποτελεί η κωδικοποίηση ICD.

2.2.1 Σύστημα SNOMED

Η κωδικοποίηση SNOMED (Systematized Nomenclature of Medicine) είναι ένα περιεκτικό σύστημα για όλα τα καταγεγραμμένα γεγονότα στο ιατρικό αρχείο συμπεριλαμβανομένων των διαγνωστικών όρων και των χειρουργικών διαδικασιών. Συγκεκριμένα, περιέβαλε τόσο τις ιατρικές και νοσηλευτικές διαγνώσεις και διαδικασίες, τα σημάδια και συμπτώματα ασθενών, την επαγγελματική ιστορία και πολλές αιτίες και αιτιολογίες των ασθενειών συμπεριλαμβανομένων των μολυσματικών καταστάσεων, των γενετικών και των φυσικών αιτιών του τραυματισμού [2].

Αποτελείται από 11 άξονες όπου ο καθένας αποτελεί ένα πλήρες ιεραρχικό σύστημα ταξινόμησης και είναι οι ακόλουθοι: 1) Τοπογραφία, 2) Μορφολογία, 3) Λειτουργία, 4) Ζωντανός οργανισμός, 5) Χημική ουσία, φάρμακα και βιολογικά προϊόντα, 6) Φυσικοί παράγοντες, 7) Επαγγέλματα, 8) Κοινωνικό πλαίσιο, 9) Ασθένεια/διάγνωση, 10) διαδικασίες και 11) Γενικοί σύνδεσμοι και Τροποποιητές.

2.2.2 Σύστημα ICD

Η ταξινόμηση ICD (International Classification of Diseases) αναπτύχθηκε και συντηρείται από τον Παγκόσμιο Οργανισμό Υγείας, με σκοπό την κωδικοποίηση των ασθενειών με ενιαίο τρόπο σε όλες τις χώρες, έτσι ώστε να είναι δυνατή η συστηματική συγκέντρωση, στατιστική ανάλυση και σύγκριση στοιχείων νοσηρότητας και θνησιμότητας. Σήμερα η ICD βρίσκεται στην 10η έκδοσή της και έχει υιοθετηθεί παγκοσμίως από πολλές χώρες μεταξύ των οποίων και η Ελλάδα [2].

Για την ονομασία των ασθενειών η ICD χρησιμοποιεί την ονοματολογία ICD (International Nomenclature of Diseases). Οι ασθένειες κωδικοποιούνται με 3 ψηφία (1 γράμμα και 2 αριθμούς), από το A00 έως το Z99 εκτός των κωδικών με U οι οποίοι είναι δεσμευμένοι για ερευνητικούς σκοπούς και ασθένειες μη προσδιορισμένης αιτίας. Οι κωδικοί ασθενειών χωρίζονται σε 21 κεφάλαια και γίνεται με κριτήρια το προσβληθέν όργανο-σύστημα οργανισμού (κεφάλαια III, IV, V, VI, VII, VIII, IX, X, XI, XIII, XIV), την αιτιολογία της ασθένειας (I, II, XVII, XIX, XX) ή άλλα κριτήρια (XV, XVI, XVIII, XXI). Η συγκεκριμένη προσέγγιση ακολουθήθηκε με σκοπό να καλύψει τις διαφοροποιημένες ανάγκες ταξινόμησης των ιατρικών επαγγελματιών. Οι κωδικοί κάθε κεφαλαίου χωρίζονται περαιτέρω σε συναφείς ομάδες ασθενειών βάσει κριτηρίων.

2.3 Απαιτήσεις Εφαρμογής

Το αντικείμενο της συγκεκριμένης Πτυχιακής Εργασίας, όπως έχει αναφερθεί, είναι να δημιουργηθεί μια εφαρμογή η οποία θα είναι εργαλείο στα χέρια του Προσωπικού Υγείας μιας Νοσοκομειακής Μονάδας. Θα αποτελεί τμήμα του ΠΣΝ και τα δεδομένα που θα διαχειρίζεται το Προσωπικό Υγείας του Νοσοκομείου θα βρίσκονται στον Κεντρικό Υπολογιστή (Server) του Νοσοκομείου.

Οι λειτουργίες που θα επιτελεί μια τέτοια εφαρμογή θα πρέπει να υλοποιούνται σε συνεργασία με τον εκάστοτε Διευθυντή της Νοσοκομειακής Μονάδας και ανάλογα με τις απαιτήσεις του κάθε τμήματος. Δηλαδή θα μπορούσε κάποιο Νοσοκομείο να ενδιαφέρεται μόνο για τις μετρήσεις των ζωτικών σημείων του ασθενή και τη λογοδοσία των Νοσηλευτών. Ένα άλλο Νοσοκομείο θα μπορούσε να ενδιαφέρεται και για την αποστολή Παραπεμπτικών στα Εργαστηριακά ή Ακτινολογικά Συστήματα.

Επίσης πληροφορίες για το τι θα περιέχει η εφαρμογή, είτε για το γραφικό περιβάλλον είναι δυνατό να συλλεχθούν από το Προσωπικό Υγείας του Νοσοκομείου μέσω

ερωτηματολογίων που θα συμπλήρωναν οι Ιατροί και Νοσηλευτές. Πρότυπα Ερωτηματολογίων παραθέτονται στο Παράρτημα. Αν απαντηθούν οι ερωτήσεις του Ερωτηματολογίου και σε συνδυασμό με τη συνεργασία με το Διευθυντή του Νοσοκομείου μπορούν να βγουν συμπεράσματα με σκοπό να χρησιμοποιηθούν για τη δημιουργία της εφαρμογής.

Στη συγκεκριμένη εφαρμογή έγινε προσπάθεια να συμπεριληφθούν παραδείγματα λειτουργιών οι οποίες μπορούν να εμπλουτιστούν ανάλογα με τις απαιτήσεις της κάθε μονάδας.

Έτσι μια από τις λειτουργίες είναι να μπορεί ο Ιατρός να αναγνώσει το **Ιστορικό** του ασθενή, το οποίο μπορεί να έχει διάφορες παραλλαγές από εφαρμογή σε εφαρμογή. Στην εφαρμογή έχουν προστεθεί όλες τις **διαγνώσεις** που έχουν γίνει κατά το παρελθόν, τη **φαρμακευτική Αγωγή** του ασθενή και τέλος **τυχόν αλλεργίες** που έχει ο ασθενής, είτε σε κάποιο συγκεκριμένο σκεύασμα είτε σε δραστική ουσία. Σκοπός για τη συγκεκριμένη λειτουργία είναι να μπορεί ο Ιατρός να πληροφορηθεί για θέματα τα οποία ίσως να μην μπορεί να έχει άλλου είδους πρόσβαση.

Μια άλλη λειτουργία είναι το **θερμομετρικό διάγραμμα** του ασθενή. Τις πληροφορίες θα τις επεξεργάζεται ο Νοσηλευτής και θα έχει πρόσβαση και ο Ιατρός. Με τον τρόπο αυτό ο Νοσηλευτής θα εισάγει τα δεδομένα άμεσα στο σύστημα και δεν θα απαιτείται επιπλέον εργασία στο Σταθμό Νοσηλευτών.

Η **λογοδοσία** μεταξύ των Νοσηλευτών σε αλλαγή βάρδιας είναι μια λειτουργία η οποία βοηθάει τους Νοσηλευτές να έχουν μια πλήρη εικόνα για τη φροντίδα του ασθενή.

Η **ενημέρωση των εργαστηριακών αποτελεσμάτων** είναι μια λειτουργία η οποία βοηθάει τον Ιατρό να τα αναγνώσει οποιαδήποτε στιγμή θέλει. Έτσι για παράδειγμα η Βάση Δεδομένων θα μπορεί ενημερωθεί κατά την επίσκεψη του Ιατρού στο θάλαμο, θα μπορεί άμεσα να δει τα αποτελέσματα και ανάλογα να συνεχίσει με την θεραπεία ή φροντίδα του ασθενή.

Μια πολύ βασική λειτουργία είναι η δημιουργία **παραπεμπτικού** από τους Ιατρούς. Με το να αποστέλλονται άμεσα τα Παραπεμπτικά στα Εργαστηριακά ή Ακτινολογικά Συστήματα εκτός του ότι δεν απαιτείται άλλη διαδικασία από το Ιατρικό προσωπικό για την εισαγωγή των δεδομένων στην κεντρική Βάση Δεδομένων, επίσης εξοικονομείται χρόνος. Έτσι τα εργαστήρια μπορούν να οργανώνουν την εργασία τους και να προετοιμάζονται κατάλληλα για την επόμενη εξέταση. Ο Ιατρός με τη σειρά του μετά το πέρας των εξετάσεων θα μπορεί να δει άμεσα τα αποτελέσματα μέσω της κινητής συσκευής.

Με τη δημιουργία των συγκεκριμένων λειτουργιών επιτυγχάνεται καλύτερη διαχείριση του χρόνου του Προσωπικού Υγείας, περιορίζεται η εργασία γραφείου και επιπλέον γίνεται μια πιο σφαιρική φροντίδα προς τον ασθενή, θέματα τα οποία επιδιώκονται για τη σωστή λειτουργία ενός ΠΣΝ.

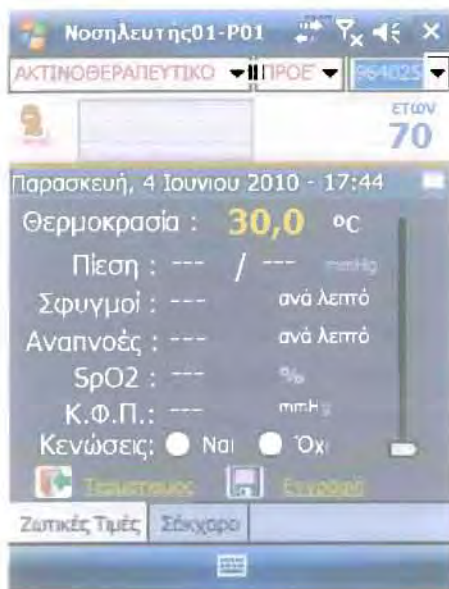
2.4 Υπάρχοντα συστήματα σε Νοσοκομεία της χώρας

Μια ανάλογη εφαρμογή έχει ήδη αναπτυχθεί για τα Δημόσια Νοσοκομεία από την εταιρεία πληροφορικής DATAMED. Συγκεκριμένα έχει υλοποιηθεί για το **Δίκτυο Υγείας Στρατιωτικών Νοσοκομείων ΦΙΛΙΠΠΟΣ** το οποίο περιλαμβάνει τα εξής νοσοκομεία: 401 ΓΣΝΑ, 251 ΓΝΑ, ΝΙΜΙΤΣ, ΝΝΑ Αθηνών, ΝΝΚ Κρήτης, 492 ΓΣΝ Αλεξανδρούπολης. Αυτή τη στιγμή όμως λειτουργεί μόνο στο Νοσοκομείο ΝΝΚ στην Κρήτη.

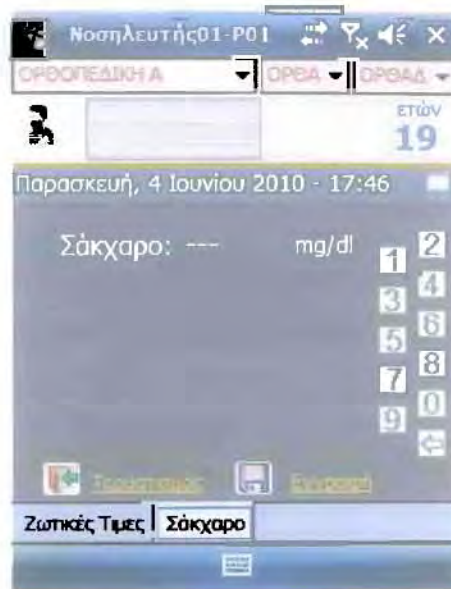
Η εφαρμογή απευθύνεται στο Νοσηλευτικό προσωπικό του νοσοκομείου και περιλαμβάνει τις εξής λειτουργίες: την καταγραφή των ζωτικών σημείων του ασθενή, όπως θερμοκρασία, πίεση, σφυγμούς, αναπνοές, SpO₂, Κ.Φ.Π., κενώσεις (Εικ. 1) και τις τιμές του σακχάρου (Εικ.2).

«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»

Ιωάννα Ζορμπά

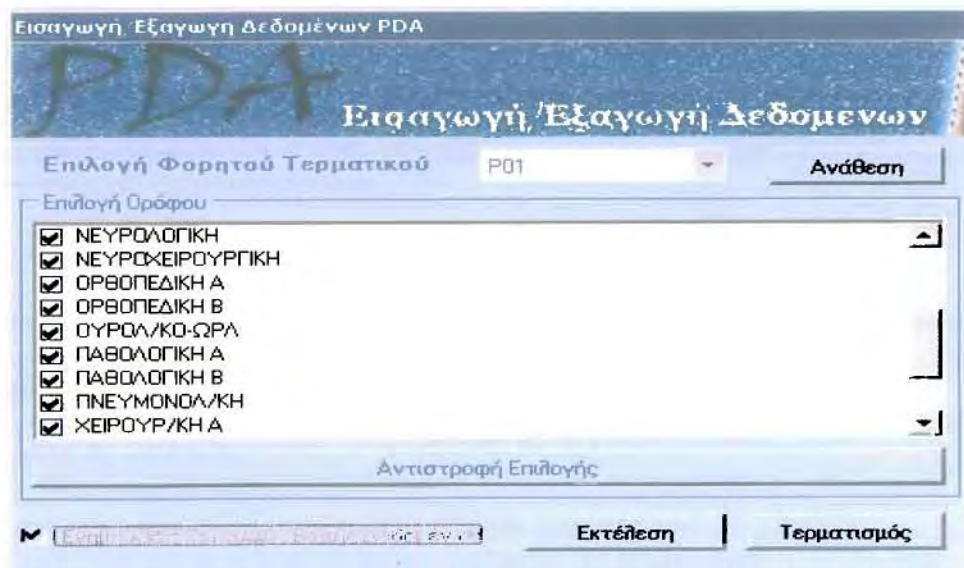


Εικόνα 1: Ζωτικές Τιμές



Εικόνα 1: Τιμές Σακχάρου

Στη συγκεκριμένη εφαρμογή ο νοσηλευτής αφού καταγράψει τις μετρήσεις θα μεταφέρει τα δεδομένα στο Σταθμό εργασίας όπου έπειτα επιλέγει το « Ενημέρωση Κεντρικής Βάσης Δεδομένων» και πατάει «Εκτέλεση» (Εικ. 3). Σε αυτό το σημείο γίνεται η σύνδεση με την Κεντρική Βάση για την ενημέρωση του Η.Φ.Υ. του ασθενούς.



Εικόνα 3: Ενημέρωση Κεντρικής Βάσης



3. ΓΕΝΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Η εφαρμογή που θα αναπτυχθεί θα πρέπει να μπορεί να κάνει τα εξής: Αφού συνδεθεί ο χρήστης μέσω του κινητού τηλεφώνου, να μπορεί να πληροφορηθεί για την καρτέλα του ασθενή και όπου χρειάζεται να μπορεί να κάνει αλλαγές και να στέλνει πίσω στην Κεντρική Βάση τις αλλαγές που τυχόν έχει κάνει ώστε να ανανεώνονται οι πίνακες. Αυτό μπορεί να γίνει με τρεις τεχνικές:

- **RDA (Remote Data Access)**
- **Web Services**
- **Merge Replication**

3.1 Τεχνικές Συγχρονισμού

3.1.1 Τεχνική RDA

Ένας μηχανισμός συγχρονισμού για την SQL Compact Edition είναι το RDA. Ο συγκεκριμένος μηχανισμός προβλέπει μια απλή **push-and-push** λειτουργικότητα μαζί με τη δυνατότητα να υποβάλλουν δηλώσεις σε ένα back-end data store χρησιμοποιώντας την OLE DB. Μέσω της μεθόδου Pull υποβάλλονται εντολές τύπου SQL SELECT στο SQL Server για επεξεργασία. Έτσι δημιουργείται ένας πίνακας στη Βάση που βρίσκεται στο SQLCE Server. Η μέθοδος Pull στέλνει τις αλλαγές που δημιουργήθηκαν από την προηγούμενη μέθοδο στη Βάση που βρίσκεται στον SQL Server [3].

Στο συγκεκριμένο μηχανισμό απαιτείται ο πίνακας που δημιουργείται να μην υπάρχει στον SQLCE κατά τη διαδικασία της μεθόδου Pull, ο RDA δεν είναι ένας καλός σχεδιασμός για εφαρμογές οι οποίες απαιτούν αυξητικά updates, επειδή χρειάζεται να γίνει Pull όλος ο πίνακας ξανά και όχι μόνο οι νέες σειρές.

3.1.2 Web Services

Η βασική πλατφόρμα των web Services είναι η XML (Extensible Markup Language) και το πρωτόκολλο HTTP. Η XML γλώσσα μπορεί να χρησιμοποιηθεί ανεξαρτήτως πλατφόρμας και γλωσσών προγραμματισμού.

Στοιχεία που χρησιμοποιεί είναι το SOAP(Simple Object Access Protocol) το οποίο επιτρέπει στις εφαρμογές να ανταλλάσσουν πληροφορία μέσω του HTTP.

Το UDDI (Universal Description, Discovery and Integration) χώρος αποθήκευσης περιγραφών υπηρεσιών διαδικτύου

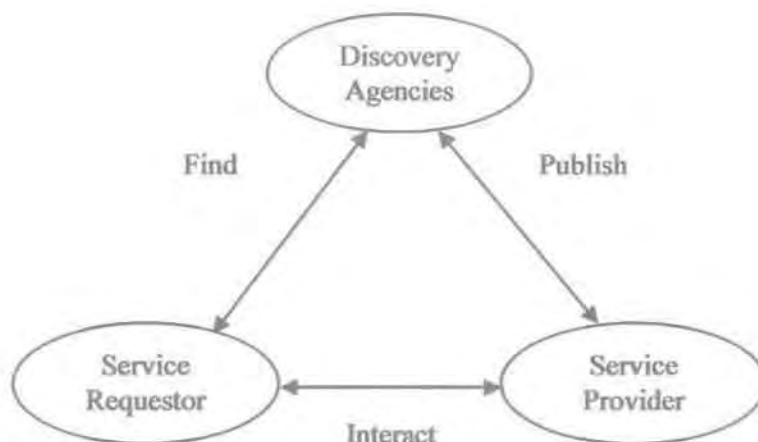
Και τη WSDL (Web Services Description Language) χρησιμοποιείται για την περιγραφή των Web services και για το πώς να αποκτηθεί η πρόσβαση σε αυτές [14].

Γίνεται ευκολότερος ο χειρισμός δεδομένων, υπάρχει ευκολία στην επικοινωνία και απλότητα υποδομής.

Ρόλοι στα Web Services:

- **Service Provider:** είναι η πλατφόρμα που φιλοξενεί την υπηρεσία
- **Service Requestor:** η εφαρμογή που καλεί μία υπηρεσία
- **Discovery Agency:** ένα σύνολο από περιγραφές υπηρεσιών στο οποίο οι service providers δημοσιεύουν τις περιγραφές των υπηρεσιών τους

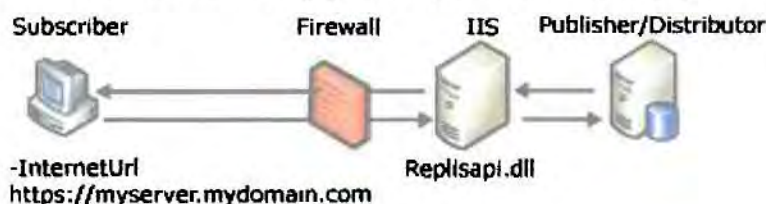
Η αλληλεπίδραση μεταξύ των ρόλων είναι το Find, το Publish και το Interact (Σχ. 4)



Σχ. 2 Η διαδικασία των Web Services

3.1.3 Η Τεχνική Merge Replication

Η διαδικασία της συγκεκριμένης τεχνικής χρησιμοποιεί έναν Publisher/Distributor και απευθύνεται σε Subscriber (Σχ. 3) και μέσω του IIS μεταφέρονται τα δεδομένα [5].



Σχ. 3 Η διαδικασία της Merge Replication

Ο **Publisher** (Σχ. 4) είναι ένας server ο οποίος κάνει διαθέσιμα τα δεδομένα για Subscription σε άλλους Servers. Εκτός από αυτό ο Publisher προσδιορίζει επίσης ποια δεδομένα έχουν αλλάξει στον **Subscriber** κατά τη διαδικασία του συγχρονισμού. Ο **Subscriber** είναι ένας server ο οποίος δέχεται και διατηρεί τα δεδομένα που έχουν γίνει publish (δημοσιευθεί). **Distributor** είναι ο Server ο οποίος διαχειρίζεται τη ροή των δεδομένων στο συγκεκριμένο σύστημα. Διακρίνονται δύο τύποι Distributor, ο ένας είναι ο απομακρυσμένος distributor και ο άλλος ο τοπικός distributor. Ο απομακρυσμένος distributor είναι χωριστά από τον Publisher και είναι σχεδιασμένος να λειτουργεί σαν ο distributor για replication. Ο τοπικός είναι ένας Server ο οποίος είναι σχεδιασμένος να λειτουργεί ως **publisher** και **distributor**. Η SQLCE είναι η Βάση η οποία είναι αποθηκευμένη στην κινητή συσκευή.

Ένα **article** μπορεί να είναι οποιοδήποτε αντικείμενο της Βάσης, όπως είναι οι πίνακες, Όψεις ή Stored Procedures. **Publication** είναι μια συλλογή από articles και **Subscription** είναι το σύνολο των δεδομένων που μπορεί να λάβει μια Βάση ή ένας Server [6]. **Agents** είναι οι διαδικασίες οι οποίες είναι υπεύθυνες για την αντιγραφή και τον διαμερισμό δεδομένων μεταξύ του publisher και του subscriber.

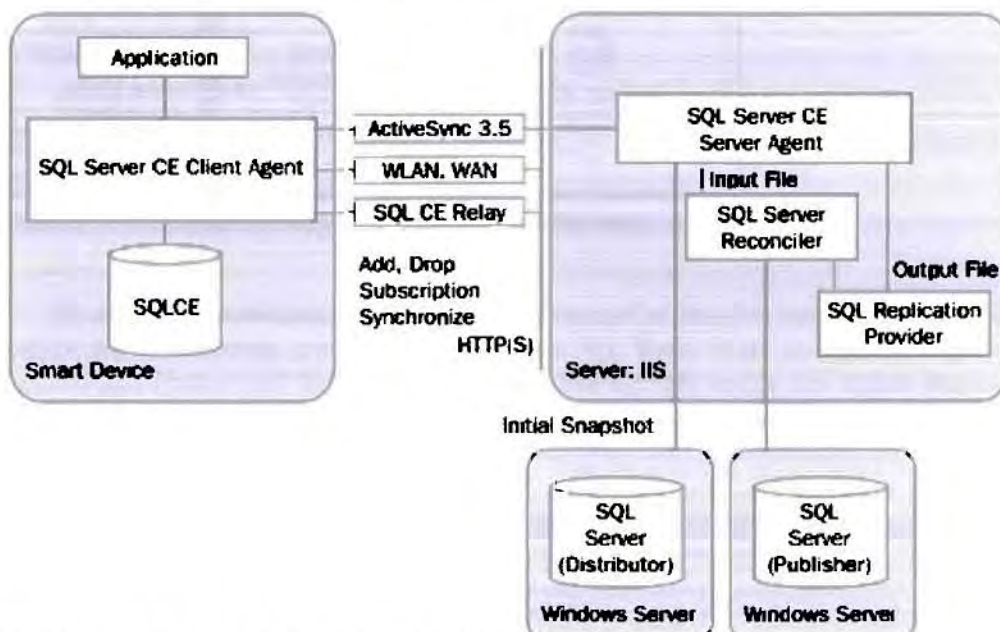
Ο **SQL Server CE Client Agent** είναι το βασικό συστατικό για τη σύνδεση για τις συσκευές οι οποίες είναι βασισμένες στις Windows CE πλατφόρμες και μέσω της τεχνικής Replication μανατζάρει τις συνδέσεις προς τον SQL Server.

Ο **SQL Server CE Server Agent** χειρίζεται τις αιτήσεις HTTP οι οποίες έρχονται από τον SQL Server CE Client Agent. Όταν μια εφαρμογή κάνει μια αίτηση, ο SQL Server CE Client Agent στέλνει το αίτημα στον SQL Server CE Server Agent μέσω του HTTP. Ο SQL

Server CE Server Agent συνδέεται με τον SQL Server και επιστρέφει τα αποτελέσματα των εγγραφών στον SQL Server CE Client Agent μέσω του HTTP πρωτοκόλλου.

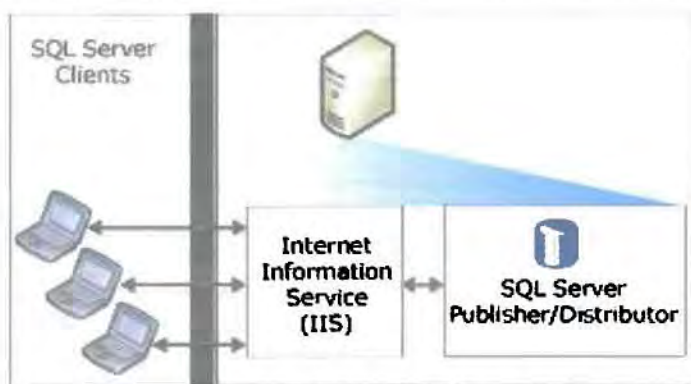
Ο SQL Server CE λειτουργεί μέσω ασύρματων δικτύων WLAN και WAN και μέσω δικτύου Ethernet και μέσω του HTTP πρωτοκόλλου.

Το ActiveSync 3.5 είναι ένα εργαλείο συγχρονισμού δεδομένων για κινητές συσκευές.



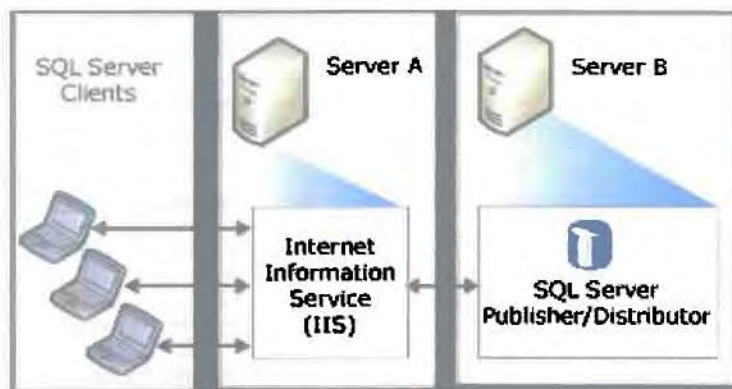
Σχ. 4 Components της τεχνικής Merge Replication

Στην πιο απλή διαμόρφωση της μεθόδου Merge Replication ο IIS (Internet Information Service), ο SQL Server Publisher και ο SQL Server Distributor, βρίσκονται σε έναν μόνο Server (Εικ.4). Οι Subscribers συγχρονίζονται με τη σύνδεσή τους στον IIS ο οποίος είναι εγκατεστημένος στον Publisher. Ο Publisher μπορεί να βρίσκεται πίσω από ένα Firewall.



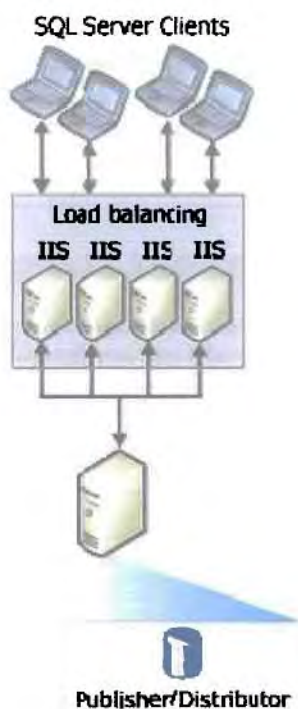
Εικόνα 4: Υλοποίηση Merge Replication με ένα Server

Ένας άλλος τρόπος αρχιτεκτονικής είναι να χρησιμοποιηθούν δύο Servers (Εικ. 5). Στον έναν Server να εγκατασταθεί ο και οι SQL Server Publisher και Distributor σε έναν άλλο. Ο Server ο οποίος έχει εγκατεστημένο τον IIS μπορεί να είναι απομονωμένος πίσω από ένα Firewall. Οι Subscribers συγχρονίζονται με τη σύνδεσή τους στον IIS.



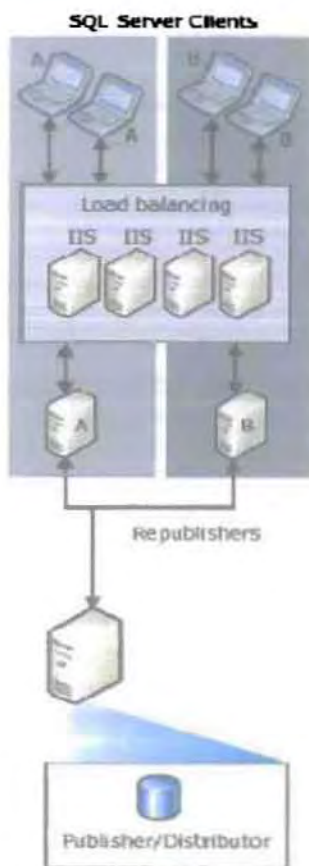
Εικόνα 5: Υλοποίηση Merge Replication με 2 Servers

Σε πολλές περιπτώσεις απαιτείται να υποστηρίξει μεγάλο αριθμό από Subscribers, οι οποίοι θα συνδέονται την ίδια στιγμή (Εικ. 6). Έτσι είναι δυνατό να σχεδιαστεί ένα μεγαλύτερο δίκτυο από Servers που τρέχουν τον IIS. Με αυτόν τον τρόπο διαμοιράζεται η εργασία και δεν φορτώνεται στον ίδιο Server με αποτέλεσμα την αποφυγή συγκρούσεων.



Εικόνα 6: Υλοποίηση Merge Replication με πολλαπλούς Servers

Για την εξισορρόπηση του επιπλέον φόρτου εργασίας στο Server όπου είναι εγκατεστημένος ο SQL Server, είναι δυνατό να δημιουργηθεί μια πολλαπλή ιεραρχία από Republishers (Εικ. 7). Ο βασικός Publisher δημοσιεύει τα δεδομένα στους Republishers, οι οποίοι με τη σειρά τους τα αναδημοσιεύουν, εξισορροπώντας έτσι τα αιτήματα από τους Subscribers.



Εικόνα 7: Υλοποίηση Merge Replication με πολλαπλούς Publishers

Ανάλογα με τις ανάγκες της εφαρμογής κάθε φορά επιλέγεται να σχεδιαστεί η αντίστοιχη υλοποίηση της Merge Replication. Για τις ανάγκες της συγκεκριμένης πτυχιακής υιοθετήθηκε η απλή αρχιτεκτονική της εικόνας 4 λόγω περιορισμένης υλικοτεχνικής υποδομής. Σε κάθε περίπτωση αυτό δεν τροποποίησε τη λειτουργικότητα του τελικού παραδοτέου της πτυχιακής.

4 ΠΑΡΟΥΣΙΑΣΗ ΤΕΧΝΟΛΟΓΙΩΝ ΣΥΣΤΗΜΑΤΟΣ

Για την υλοποίηση της Πτυχιακής εργασίας χρησιμοποιήθηκε το παρακάτω λογισμικό:

- Microsoft SQL Server 2008 Developer
- Microsoft Visual Studio 2008 Professional Edition
- Windows Mobile 6 Professional SDK v6.0A
- Service Pack 2 Server Tools
- Windows Mobile Device Center Internet Information Services (IIS)
- Microsoft SQL Server Compact 3.5
- microsoft.NET compact Framework 3.5

4.1 Microsoft SQL Server 2008 Developer

Ο **Microsoft SQL Server 2008 Developer** που χρησιμοποιήθηκε προμηθεύθηκε από το Web Store του msdn.academic alliance Software Center.

4.2 Microsoft Visual Studio 2008 Professional Edition

Το **Microsoft Visual Studio 2008 Professional Edition** που χρησιμοποιήθηκε προμηθεύθηκε από το Web Store του msdn.academic alliance Software Center. Περιλαμβάνει το .NET Framework 3.5. Στο πρόγραμμα θα πρέπει να εγκατασταθεί το **Visual Studio 2008 Service Pack 1 (SP1)** [7] το οποίο περιλαμβάνει και το **NET Framework 3.5 SP1**. Για το **Visual Studio 2008 Service Pack 1 (SP1)** εφαρμόζεται το παρακάτω εκτελέσιμο:

VS90sp1-KB945140-ENU.exe

Περιέχει εκτός των άλλων βιβλιοθηκών, τη βασική βιβλιοθήκη **System.Data.SqlServerCe** [8] η οποία περιέχει κλάσεις μέσω των οποίων υπάρχει η δυνατότητα να δημιουργηθούν, να διαχειριστούν και να συγχρονιστούν Βάσεις του **SQL Server Compact 3.5 SP1** από έναν υπολογιστή ή μια smart device.

4.3 Windows Mobile 6 Professional SDK v6.0A

Το **Windows Mobile 6 SDKs** [9] περιέχει κώδικες, βιβλιοθήκες, emulator και εργαλεία για το **Visual Studio** ώστε να μπορείς να δημιουργήσεις εφαρμογές για το **Windows Mobile 6**. Emulators που περιέχονται είναι οι παρακάτω.

- Windows Mobile 6 Standard SDK
 - Windows Mobile 6 Standard (176x220 pixels - 96 dpi)
 - Windows Mobile 6 Standard Landscape QVGA (240x320 pixels - 131 dpi)
 - Windows Mobile 6 Standard QVGA (320x240 pixels - 131 dpi)
- Windows Mobile 6 Professional SDK
 - Windows Mobile 6 Classic (240x320 pixels - 96 dpi)
 - Windows Mobile 6 Professional (240x320 pixels - 96 dpi)
 - Windows Mobile 6 Professional Square (240x240 pixels - 96 dpi)

- Windows Mobile 6 Professional Square QVGA (320x320 pixels - 128 dpi)
- Windows Mobile 6 Professional Square VGA (480x480 pixels - 192 dpi)
- Windows Mobile 6 Professional VGA (480x640 pixels - 192 dpi)

Η εφαρμογή δημιουργήθηκε σε λειτουργικό Vista και επιπλέον απαιτήσεις είναι το **microsoft.NET compact Framework 3.5** και το **Windows Mobile Device Center**.

Τα αρχεία που είναι διαθέσιμα είναι τα παρακάτω και ανάλογα με ποιον Emulator θέλουμε να εργαστούμε το κάνουμε download και install. Για τη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε ο **Windows Mobile 6 Professional**.

Windows Mobile 6 Professional SDK Refresh.msi
Windows Mobile 6 Standard SDK Refresh.msi

4.4 Internet Information Services (IIS)

Είναι ο **Web Server της Microsoft** για την επικοινωνία με το **www**. Υποστηρίζει τα πρωτόκολλα **HTTP, HTTPS, FTP, SMTP** και το **NNTP**. Είναι τμήμα του Λειτουργικού Συστήματος αλλά για να χρησιμοποιηθεί απαιτούνται ορισμένες ρυθμίσεις οι οποίες θα αναλυθούν στην παράγραφο με τις Ρυθμίσεις του κάθε συστήματος.

4.5 Microsoft SQL Server Compact 3.5 Service Pack 2 Server Tools

Το συγκεκριμένο εργαλείο χρησιμοποιείται για το συγχρονισμό μεταξύ του υπολογιστή που χρησιμοποιεί τον **IIS** και του **SQL Server**. Περιέχει ένα βασικό εργαλείο το **Configure Web Synchronization wizard** όπου η λειτουργία του αναφέρεται στο κεφάλαιο 7 στην επεξήγηση του συγχρονισμού των δεδομένων.

Υπάρχουν διαθέσιμες δύο εκδόσεις και η επιλογή γίνεται ανάλογα με το σύστημα [10]

SSCEServerTools_x64-ENU.msi
SSCEServerTools_x86-ENU.msi

4.6 Windows Mobile Device Center

Χρησιμοποιείται για το συγχρονισμό μεταξύ του υπολογιστή και της κινητής συσκευής για μεταφορά αρχείων. Για τα Windows Vista απαιτείται το παρακάτω εκτελέσιμο [11].

dvdupdate-x86.exe

4.7 microsoft.NET compact Framework 3.5

Το **Microsoft.NET compact Framework 3.5** περιέχει βιβλιοθήκες οι οποίες είναι χρήσιμες για τον σχεδιασμό και την επικοινωνία των εφαρμογών και οι οποίες συμπεριλαμβάνονται στο πρόγραμμα. Μια βασική βιβλιοθήκη είναι η **System.Data.SqlClient** [12] η οποία περιέχει κλάσεις που απαιτούνται για την προσπέλαση και διεκπεραίωση λειτουργιών της ΒΔ του **SQL Server**.

Αφού εγκατασταθούν τα βασικά προγράμματα θα πρέπει να γίνουν ορισμένες ρυθμίσεις για τη σωστή λειτουργία της εφαρμογής.

5 ΡΥΘΜΙΣΕΙΣ ΤΕΧΝΟΛΟΓΙΩΝ ΣΥΣΤΗΜΑΤΟΣ

5.1 Microsoft SQL Server 2008 Developer

Μέσα στον **SQL Server Configuration Manager** ενεργοποιούνται όλες τις υπηρεσίες για να λειτουργούν αυτόματα και όχι χειροκίνητα (Εικ. 8).



Εικόνα 8: SQL Server Configuration Manager

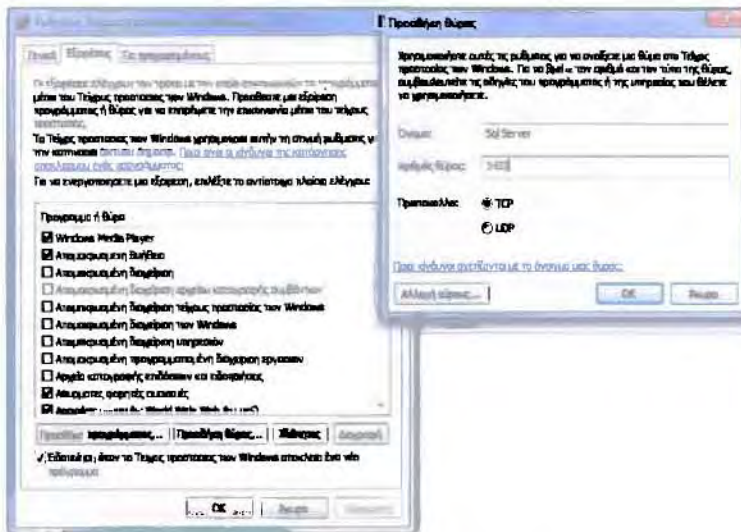
Επίσης στις ρυθμίσεις του **SQL Server Configuration Manager** στην επιλογή **Protocols for MSSQLSERVER** θα πρέπει να ενεργοποιηθούν τα πρωτόκολλα **Named Pipes** και **TCP/IP** τα οποία αρχικά είναι απενεργοποιημένα (Εικ. 9).



Εικόνα 9: SQL Server Configuration Manager, Πρωτόκολλα

5.2 Windows firewall

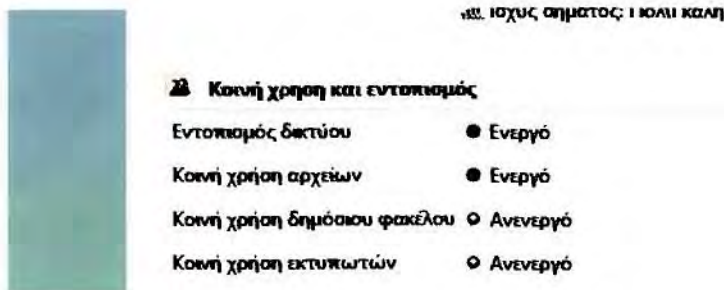
Στις ρυθμίσεις του **Firewall** θα πρέπει να προστεθεί μια εξαίρεση για την **Port 1433** του **SQL Server**. Από τον πίνακα ελέγχου μεταβαίνουμε στο Τείχος Προστασίας των Windows. Στις ρυθμίσεις θα επιλέξουμε τις Εξαιρέσεις και θα επιλέξουμε το tab Προσθήκη θύρας, όπου στο νέο παράθυρο θα πρέπει να προστεθεί η ονομασία και ο αριθμός της θύρας (Εικ. 10)



Εικόνα 10: Προσθήκη Θύρας SQL Server

5.3 Σύνδεση Ίντερνετ

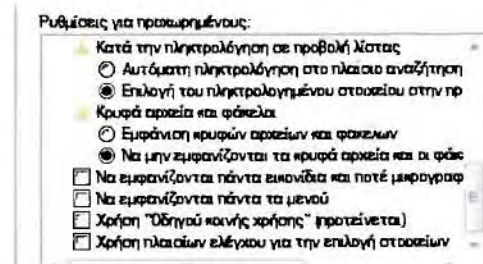
Από τον πίνακα ελέγχου -> Δίκτυο και Ίντερνετ -> Ρυθμίσεις κοινής χρήσης αρχείων και θα ενεργοποιηθεί η Κοινή χρήση αρχείων για να μην υπάρχει πρόβλημα στην πρόσβαση σε αρχεία (Εικ.11).



Εικόνα 2: Ρυθμίσεις κοινής χρήσης αρχείων

5.4 Επιλογές φακέλων

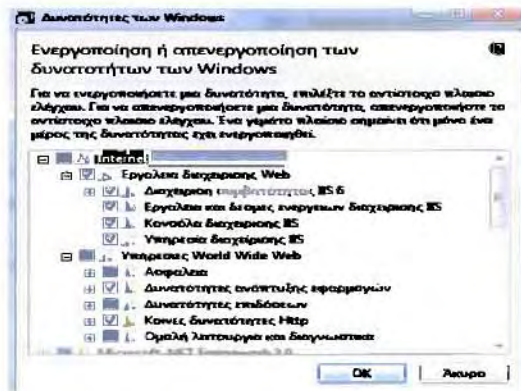
Το επόμενο που θα πρέπει να ρυθμιστεί είναι οι φάκελοι για να μην υπάρχει πρόβλημα όταν κάποιος συνδεθεί μέσω της εφαρμογής και χρειάζεται πρόσβαση σε αυτούς. Από τον πίνακα ελέγχου -> Επιλογές φακέλων -> Προβολή θα γίνει από-επιλογή της επιλογής Χρήση οδηγού κοινής χρήσης (προτείνεται) (Εικ. 12).



Εικόνα 3: Από-επιλογή Οδηγού κοινής χρήσης

5.5 Ενεργοποίηση του IIS

Οι χρήστες θα πρέπει να έχουν πρόσβαση στον Web Server όπου είναι και ο ενδιαμέσος σταθμός για τη διαχείριση των αρχείων. Για να ενεργοποιηθεί η συγκεκριμένη λειτουργία πηγαίνουμε Έναρξη -> Προγράμματα -> Ενεργοποίηση ή απενεργοποίηση των δυνατοτήτων των windows ->Internet Information Services-> Εργαλεία διαχείρισης Web και τσεκάρουμε όλες τις επιλογές, επίσης τις Δυνατότητες ανάπτυξης εφαρμογών και τις κοινές δυνατότητες Http (Εικ. 13).



Εικόνα 13: Ενεργοποίηση IIS

6 ΑΝΑΠΤΥΞΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΤΟΥ SERVER

Ο σχεδιασμός της συγκεκριμένης Βάση Δεδομένων (ΒΔ) δημιουργήθηκε χρησιμοποιώντας κριτήρια δημιουργίας από το σύγγραμμα Πληροφορική της Υγείας των John Mantas και Arie Hasman [13], σημειώσεις που αφορούν τον Ηλεκτρονικό Φάκελο Υγείας (ΗΦΥ) του Καθηγητή κ. Ιωάννη Μαντά από το μάθημα Πληροφοριακά Συστήματα Υγείας [1]. Για τα δεδομένα που έχουν εισαχθεί χρησιμοποιήθηκαν : ο ICD10 κατάλογος από το Υπουργείο Υγείας [14], η Λίστα Φαρμάκων από το ΙΦΕΤ(Ινστιτούτο Φαρμακευτικής Έρευνας & Τεχνολογίας) [15], η λίστα των Ειδικοτήτων από τον Ιατρικό Σύλλογο [16] και οι Ιατρικές Εξετάσεις από την ηλεκτρονική σελίδα της Ιατρικής Σχολής του Πανεπιστημίου Αθηνών [17]

6.1 Πίνακες της ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Για τη δημιουργία της Βάσης Δεδομένων (ΒΔ) χρησιμοποιήθηκε η βιβλιοθήκη msdn της Microsoft [18]. Δημιουργήθηκαν δυο ΒΔ, η μια για τον κεντρικό υπολογιστή (Server) η οποία ονομάζεται **HospitalSv** και μια για την εφαρμογή του κινητού τηλεφώνου η οποία ονομάζεται **Hospital** η οποία θα μείνει προς το παρόν κενή. Η ΒΔ του κεντρικού υπολογιστή που δημιουργήθηκε, σκοπίμως περιέχει αρκετούς πίνακες για δυο λόγους, α) για την διευκόλυνση των N-M σχέσεων και β) γιατί έτσι εντοπίστηκαν αρκετά λάθη που θα μπορούσαν να γίνουν αν δεν γίνει ο σχεδιασμός της ΒΔ προσεκτικά, λόγοι οι οποίοι θα αναλυθούν παρακάτω στην επεξήγηση για τις N-M σχέσεις. Η ΒΔ του κεντρικού υπολογιστή περιέχει τους παρακάτω πίνακες (Πιν.1):

DIAGNOSIS	NURSE
ALLERGIES	PASSWORD
CLINIC	PATIENT
CLINICDOCTOR	PATIENTALLERGIES
CLINICNURSE	PATIENTCLINIC
DOCTOR	PATIENTDIAGNOSIS
DRUGLIST	PATIENTDOCTOR
DRUGLISTMEDICATION	PATIENTLABRESULTS
EXAMINATIONS	PATIENTMEDICATION
EXAMINATIONSBIOCH	PATIENTNURSE
EXAMINATIONSBIOCHREFERENCE	PATIENTSYMPTOMS
ICD10	PATIENTTHERMDIAGRAM
ICD10ALLERGIES	PATIENTWARD
ICD10DIAGNOSIS	REFERENCE
ICD10SYMPTOMS	SPECIALITY
LABRESULTS	SYMPTOMS
LABRESULTSEXAMINATIONS	TEMPERATUREDIAGRAM
MEDICATION	WARD

Πιν. 1 Πίνακες Βάσης Δεδομένων

Η γραφική απεικόνιση του Μοντέλου Οντοτήτων- Συσχετίσεων (Entity-Relationship E-R) μας βοηθάει να δούμε αναλυτικά τους πίνακες που περιέχονται στη Βάση Δεδομένων και τις σχέσεις μεταξύ τους (Σχ.5). Στο σχεδιάγραμμα στους τύπους Οντοτήτων απεικονίζονται οι βασικοί πίνακες και στους τύπους Συσχετίσεων όπου βρίσκονται σχέσεις N-M απεικονίζονται πίνακες οι οποίοι έχουν δημιουργηθεί για τη σύνδεση μεταξύ των πινάκων με τέτοιου είδους σχέσεις μεταξύ τους.

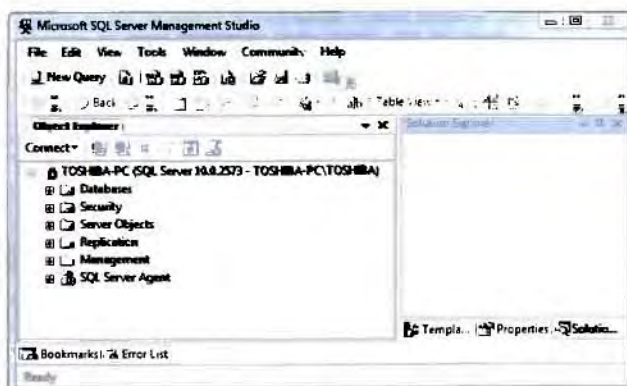
6.2 Δημιουργία της ΒΔ

Η πρώτη δουλειά είναι να δημιουργηθεί η ΒΔ μέσω του εργαλείου SQL Server Management Studio (SSMS). Από την έναρξη -> όλα τα προγράμματα -> Microsoft Sql Server επιλέγεται το SSMS. Θα ανοίξει το παρακάτω παράθυρο(Εικ.14). Στην επιλογή Server type επιλέγεται το **Database Engine** μιας και η συγκεκριμένη βάση θα δημιουργηθεί για τον κεντρικό υπολογιστή. Στην επιλογή **Server name** επιλέγουμε το όνομα του Server μας και στην επιλογή **Authentication** θα συνδεθούμε με **Windows Authentication**. Θα μπορούσε να επιλεγεί η **SQL Server Authentication** και να προστεθεί κάποιο password για την πρόσβαση στη Βάση.



Εικόνα 14: Σύνδεση με SQL Server

Επιλέγοντας **Connect** μεταβαίνουμε στο περιβάλλον του SSMS όπου στον Object Explorer εμφανίζεται μια λίστα με δυνατότητες (Εικ. 15). Θα πάμε στο **Databases** και με δεξί κλικ επιλέγεται **New Database**.



Εικόνα 15: Microsoft SQL Server Management Studio (SSMS), Object Explorer

Στο επόμενο παράθυρο στο **Database name** συμπληρώνεται το όνομα της Βάσης. Επίσης δίνονται πληροφορίες για τον τύπο των αρχείων που θα δημιουργηθούν και για το που θα αποθηκευτούν τα συγκεκριμένα αρχεία. Επιλέγοντας **OK** θα δημιουργηθεί η Βάση και θα προστεθεί στη λίστα με τις βάσεις (Εικ. 16).



Εικόνα 16: SSMS, Δημιουργία Βάσης Δεδομένων

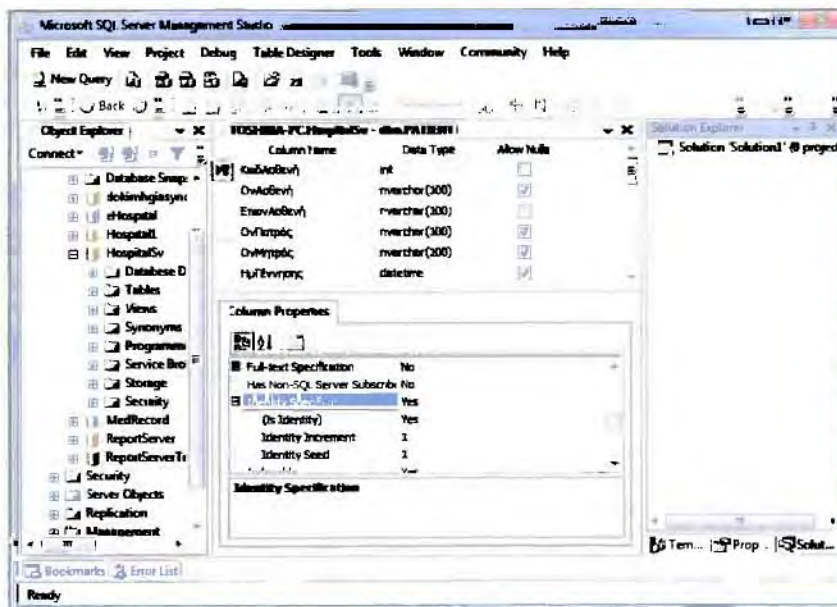
Επιλέγοντας τη Βάση και με δεξί κλικ στο Tables επιλέγεται το New Table για τη δημιουργία ενός νέου πίνακα. Ο πρώτος πίνακας θα είναι ο πίνακας PATIENT, ο οποίος δίνεται αναλυτικά (Πιν. 2). Στην πρώτη στήλη είναι το όνομα της στήλης μας στον πίνακα, στη δεύτερη τι τύπου είναι τα δεδομένα, στην τρίτη αν επιτρέπεται να είναι κενή, δηλαδή να μην εισάγονται δεδομένα, στην τέταρτη αν κάποια από τη στήλη είναι το Primary Key και στην πέμπτη στήλη διάφορες παρατηρήσεις. Στον πίνακα PATIENT εκτός από τα προσωπικά στοιχεία όπως το όνομα και το επίθετο του ασθενή, διακρίνονται και τα δημογραφικά του στοιχεία όπως είναι η διεύθυνση και η περιφέρεια. Επίσης στον ίδιο πίνακα έχουν προστεθεί οι αριθμοί μητρώων όπως είναι το AMKA και το AM του Ασφαλιστικού του ταμείου.

PATIENT				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΑσθενή	int		PRIMARY KEY	IDENTITY(1,1)
ΟνΑσθενή	nvarchar(100)	✓		
ΕπωνΑσθενή	nvarchar(100)			
ΟνΠατρός	nvarchar(100)	✓		
ΟνΜητρός	nvarchar(100)	✓		
ΗμΓέννησης	date	✓		
Διεύθυνση	nvarchar(100)	✓		
Πόλη	nvarchar(100)	✓		
Νομός	nvarchar(100)	✓		
TK	int	✓		
Περιφέρεια	nvarchar(100)	✓		
Τηλέφωνο	int	✓		
Επάγγελμα	nvarchar(100)	✓		
ΟυκΚατάσταση	nvarchar(100)	✓		
Φύλο	nvarchar(100)	✓		
ΑρΤαυτότητας	nvarchar(100)	✓		
ΑΦΜ	nvarchar(100)	✓		
AM	nvarchar(100)	✓		
AMKA	nvarchar(100)	✓		
ΗμερΕισαγωγής	datetime			GETDATE()
ΗμεΞαγωγής	datetime	✓		

Πιν. 2 Πίνακας PATIENT

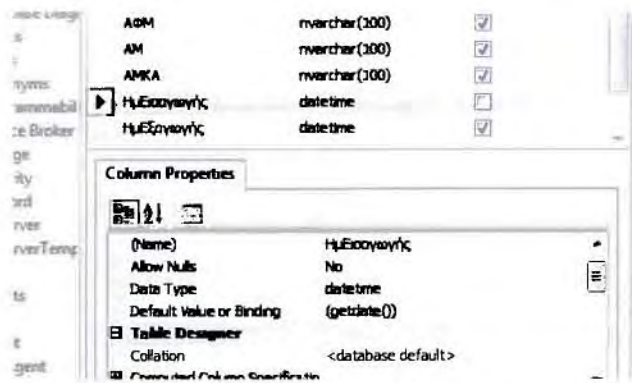
Στην πρώτη στήλη θα γραφτεί η ονομασία της, στη δεύτερη τι τύπου είναι πχ int και στην τρίτη αν επιτρέπεται να είναι κενή (Allow NULL) (Εικ. 17). Μια στήλη η οποία είναι ο κωδικός των εγγραφών δεν επιτρέπεται να είναι ποτέ NULL. Κάνοντας δεξί κλικ στη στήλη του ΚωδΑσθενή με την επιλογή Set Primary Key ορίζουμε ότι η συγκεκριμένη στήλη θα είναι το Primary Key.

Στο παράθυρο Column Properties παρατηρείται ότι έχει γίνει η επιλογή το Identity Specification να είναι yes, το Identity Increment να έχει τιμή 1 και το Identity Seed να έχει τιμή 1. Με τη συγκεκριμένη εντολή δίνεται η σκυτάλη στον SQL Server να διαχειρίζεται μόνος του την στήλη ΚωδΑσθενή, δηλαδή να βάζει αυτόματα τιμές στη συγκεκριμένη στήλη. Τα όρια 1,1 σημαίνουν ότι θα αρχίσει από το ένα και με βήμα 1 θα αυξάνεται σε κάθε εγγραφή, δηλαδή αν η τελευταία εγγραφή είχε κωδικό 5, η επόμενη που θα εισαχθεί θα είναι η τιμή 6. Αυτό βοηθάει στο να μην υπάρχουν διπλό-εγγραφές, που στην περίπτωση κλειδιού αυτό είναι το θεμιτό.



Εικόνα 17: Δημιουργία Πίνακα PATIENT

Παρατηρείται επίσης ότι στη στήλη ΗμΕισαγωγής στο Default Value or Binding έχει οριστεί η έτοιμη μέθοδος GETDATE() (Εικ. 18). Με αυτόν τον τρόπο όταν γίνει η εγγραφή στη συγκεκριμένη στήλη θα προστεθεί αυτόματα η ημερομηνία και ώρα της εισαγωγής του ασθενή. Κάνοντας Save θα ζητείται να δοθεί το όνομα του πίνακα και πατώντας OK θα αποθηκευτεί ο πίνακας στη Βάση Δεδομένων.



Εικόνα 18: SSMS, Ιδιότητες Στήλης

Ο επόμενος πίνακας είναι ο CLINIC στον οποίο κλειδί του πίνακα θα είναι η στήλη ΚωδΚλινικής με τιμές από 1 και με βήμα 1, θα περιέχει την ονομασία της και μια στήλη με την συντομογραφία της Κλινικής (Πιν. 3).

CLINIC				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΚλινικής	int		PRIMARY KEY	IDENTITY(1,1)
ΟνομασίαΚλινικής	nvarchar(100)	✓		
ΣυντΚλινικής	nvarchar(100)	✓		Συντομογραφία

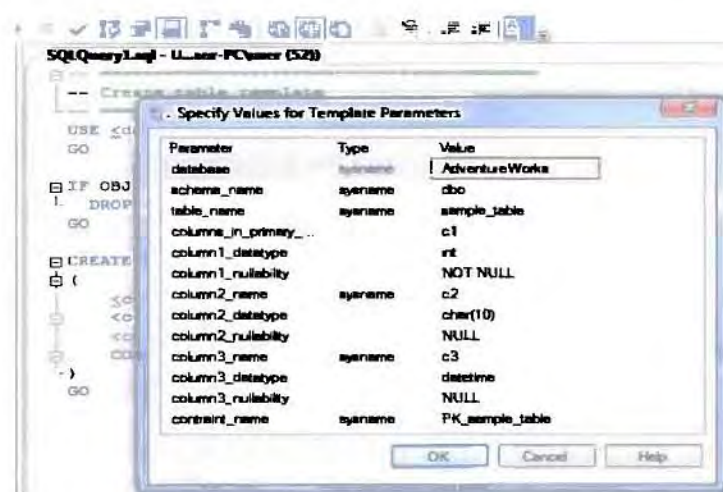
Πιν. 3 Πίνακας CLINIC

Ο συγκεκριμένος πίνακας θα δημιουργηθεί με έναν άλλο τρόπο, χρησιμοποιώντας τη γλώσσα T-SQL(Transact-SQL) και την χρήση έτοιμων προτύπων. Από την γραμμή εργαλείων στην επιλογή View επιλέγεται Template Explorer. Εμφανίζεται ένα παράθυρο το οποίο περιέχει διάφορα πρότυπα. Στην συγκεκριμένη περίπτωση επιλέγοντας τον φάκελο Tables και με διπλό κλικ επιλέγεται το Create Table θα δημιουργηθεί ένας νέος πίνακας (Εικ. 19).



Εικόνα 19:SSMS, Template Explorer

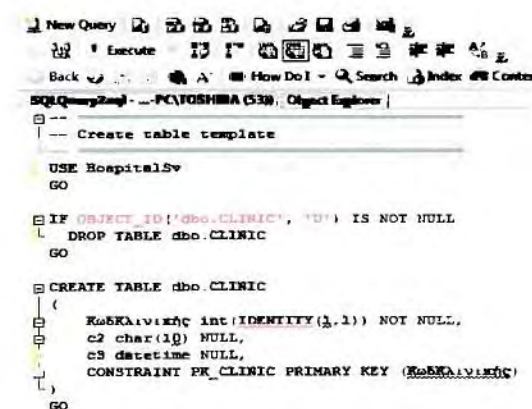
Στον editor θα εμφανιστεί ένας κώδικας γενικού τύπου, όπου εκεί καθορίζονται οι τιμές για τις παραμέτρους. Επιλέγοντας το εικονίδιο A/B όπως φαίνεται στην εικόνα θα εμφανιστεί ένα καινούργιο παράθυρο (Εικ. 20).



Εικόνα 20: Καθορισμός τιμών στις παραμέτρους

Στο πεδίο database γράφεται το όνομα της Βάσης Δεδομένων, στη συγκεκριμένη περίπτωση θα είναι HospitaSv. Στο table_name το όνομα του πίνακα που θα δημιουργηθεί, και στη συνέχεια τις στήλες του πίνακα. Η πρώτη είναι ως συνήθως για την στήλη η οποία θα χρησιμοποιηθεί και ως κλειδί του πίνακα. Στο column1_datatype επιλέγεται ο τύπος, και στο column1_nullability η επιλογή μένει ως έχει δηλαδή NOT NULL μιας και θα είναι το κλειδί και δεν επιτρέπεται να μην παίρνει τιμή. Το τελευταίο πεδίο constraint name είναι για το όνομα του Πρωτεύοντος Κλειδιού(Primary Key). Επιλέγοντας Ok στον editor εμφανίζεται το script, όπως φαίνεται στην εικόνα, το οποίο επιλέγοντας Execute θα δημιουργηθεί ο πίνακας.

Ο κώδικας που εμφανίζεται έχει ως εξής. Χρησιμοποιεί την Βάση που έχει οριστεί, αν υπάρχει πίνακας με την ίδια ονομασία, τον διαγράφει και δημιουργεί έναν καινούργιο. Στο PRIMARY KEY έχει οριστεί αυτόματα η στήλη ΚωδΚλινικής. Στο script τώρα μπορούν να προστεθούν όσες στήλες χρειάζονται με τον τύπο τους αρκεί να χωρίζονται μεταξύ τους με κόμμα (Εικ. 21).



Εικόνα 21: T-SQL για τη δημιουργία του πίνακα CLINIC

Οι επόμενοι πίνακες θα δημιουργηθούν με τον πρώτο τρόπο. Ο πίνακας ΘΑΛΑΜΟΣ θα

**«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»**

Ιωάννα Ζορμπά

περιέχει τη στήλη ΚωδΘαλάμου και είναι το κλειδί του πίνακα, οι τιμές θα ορίζονται από 1 με βήμα 1 και θα αριθμεί τις κλίνες που βρίσκονται στον κάθε θάλαμο (Πιν. 4). Η στήλη ΚωδΚλινικής θα είναι Foreign Key για τον πίνακα CLINIC.

WARD				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΘαλάμου	int		PRIMARY KEY	IDENTITY(1,1)
Κλίνη	int	✓		
ΚωδΚλινικής	int	✓	FOREIGN KEY	Κλειδί για Κλινική

Πιν. 4 Πίνακας WARD

Ο επόμενος θα είναι ο πίνακας ΓΙΑΤΡΟΣ το κλειδί του οποίου θα είναι η στήλη ΚωδΓιατρού η οποία δεν θα δέχεται τιμές NULL, ορίζουμε να παίρνει τιμές από το 1 και με βήμα 1, το Επώνυμο δεν θα δέχεται NULL τιμές . Θα περιέχει την στήλη της ειδικότητας η οποία είναι το Primary Key στον πίνακα SPECIALITY, και στο συγκεκριμένο πίνακα είναι Foreign Key (Πιν.5).

DOCTOR				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΓιατρού	int		PRIMARY KEY	IDENTITY(1,1)
ΟνΓιατρού	nvarchar(100)	✓		
ΕπωνΓιατρού	nvarchar(100)	✓		
ΕιδΓιατρού	nvarchar(100)	✓		
ΤηλΓιατρού	int	✓		
ΚωδΕιδικότητας	int		FOREIGN KEY	Κλειδί για ΕΙΔΙΚΟΤΗΤΕΣ

Πιν. 5 Πίνακας DOCTOR

Συνεχίζουμε με τον πίνακα NURSE. Το κλειδί του πίνακα θα είναι η στήλη ΚωδΝοσηλευτή η οποία δεν θα δέχεται τιμές NULL, ορίζουμε να παίρνει τιμές από το 1 και με βήμα 1. Το Επώνυμο δεν θα δέχεται NULL τιμές και θα περιέχει επίσης την στήλη της ειδικότητας όπου θα αναφέρεται στο Primary Key του πίνακα SPECIALITY (Πιν. 6).

ΝΟΣΗΛΕΥΤΗΣ				
COLUMN NAME	DATA	ALLOW NULLS		
ΚωδΝοσηλευτή	int		PRIMARY KEY	IDENTITY(1,1)
ΟνΝοσηλευτή	nvarchar(100)	✓		
ΕπωνΝοσηλευτή	nvarchar(100)	✓		
ΕιδΝοσηλευτή	nvarchar(100)	✓		
ΤηλέφωνοΝοσηλευτή	int	✓		
ΚωδΕιδικότητας	int		FOREIGN KEY	Κλειδί για ΕΙΔΙΚΟΤΗΤΕΣ

Πιν.6 Πίνακας NURSE

Στον πίνακα DRUGLIST έχει προστεθεί η στήλη ΚΩΔΕΟΦ όπου αναφέρεται ο κωδικός

**«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»**

Ιωάννα Ζορμπά

του κάθε φαρμάκου από τον ΕΟΦ (Εθνικός Οργανισμός Φαρμάκων), τη ΦαρμΜορφή δηλαδή αν είναι ταμπλέτες ή σιρόπι κλπ, τον ATC (Anatomical Therapeutic Chemical) κώδικα [20], τη στήλη ΔραστΣυστατικό όπου είναι η δραστική ουσία του κάθε φαρμάκου και την Περιεκτικότητα αυτού (Πιν. 7).

DRUGLIST				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΛισΦαρμάκου	int		PRIMARY KEY	IDENTITY(1,1)
ΚωδΕΟΦ	nvarchar(100)	✓		Κωδικός του ΕΟΦ
ΟνομΦαρμάκου	nvarchar(250)	✓		
ΦαρμΜορφή	nvarchar(100)	✓		Μορφή φαρμάκου
ΑΤCΚώδικας	nvarchar(100)	✓		Κώδικας ATC
ΔραστΣυστατικό	nvarchar(100)	✓		Δραστική ουσία
Περιεκτικότητα	nvarchar(100)	✓		Περιεκτικότητα

Πιν. 7 Πίνακας DRUGLIST

Ο ICD10 είναι ο επόμενος πίνακας (Πιν. 8). Η πρώτη στήλη είναι ο κωδικός που δίνεται ως πρωτεύον κλειδί. Στη στήλη ΚωδΝόσου περιέχεται ο κωδικός που έχει δώσει ο οργανισμός World Health Organization (WHO) και ΟνΝόσου όπως αναφέρεται στον πίνακα του ICD10.

ICD10				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδICD10	Int		PRIMARY KEY	IDENTITY(1,1)
ΚωδΝόσου	nvarchar(250)	✓		
ΟνΝόσου	nvarchar(250)	✓		

Πιν. 8 Πίνακας ICD10

Στον πίνακα ALLERGIES ο ΚωδΑλλεργίας θα περιέχει την κωδικοποίηση της κάθε αλλεργίας όπως έχει οριστεί από τον WHO και περΔιαγνΑλλεργίας είναι η περιγραφή της κάθε αλλεργίας (Πιν. 9).

ALLERGIES				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΠινΑλλεργίας	Int		PRIMARY KEY	IDENTITY(1,1)
ΚωδΑλλεργίας	nvarchar(100)	✓		
ΠερΔιάγνΑλλεργίας	nvarchar(250)	✓		

Πιν. 9 Πίνακας ALLERGIES

Επόμενος πίνακας είναι ο _DIAGNOSIS με ΚωδΔιάγνωσης IDENTITY(1,1), όπου στη στήλη ΟνΔιάγνωσης αποθηκεύονται οι διαγνώσεις των Ιατρών (Πιν. 10).

_DIAGNOSIS				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΔιάγνωσης	int		PRIMARY KEY	IDENTITY(1,1)
ΟνΔιάγνωσης	nvarchar(300)	✓		
ΗμΔιάγνωσης	datetime	✓		

Πιν. 10 Πίνακας DIAGNOSIS

Στον πίνακα EXAMINATIONS η δεύτερη στήλη περιέχει την ονομασία της εξέτασης αλλά περισσότερο χρησιμοποιείται η συντομογραφία της, η οποία είναι η τρίτη στήλη (Πιν.11). Έχουν προστεθεί σε διαφορετικές στήλες οι τιμές αναφοράς για τις γυναίκες και για τους άντρες.

EXAMINATIONS				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΕξέτασης	int		PRIMARY KEY	IDENTITY(1,1)
ΟνΕξέτασης	nvarchar(100)	✓		
ΣυνΕξέτασης	nvarchar(100)	✓		
ΜονΜέτρησης	nvarchar(100)	✓		
ΤιμέςΑναφοράςΑ	nvarchar(100)	✓		
ΤιμέςΑναφοράςΓ	nvarchar(100)	✓		

Πιν. 11 Πίνακας EXAMINATIONS

Ακολουθεί ο πίνακας SPECIALITY όπου εκεί βρίσκονται οι ειδικότητες του ιατρικού προσωπικού με κλειδί τη στήλη ΚωδΕιδικότητας με τιμές από το 1 και με βήμα 1, με το όνομα της Ειδικότητας και τη συντόμευση αυτή (Πιν.12).

SPECIALITY				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΕιδικότητας	int		PRIMARY KEY	IDENTITY(1,1)
ΟνΕιδικότητας	nvarchar(100)	✓		
ΣυνΕιδικότητας	nvarchar(100)	✓		

Πιν. 12 Πίνακας SPECIALITY

Ο πίνακας SYMPTOMS είναι ο επόμενος, όπου στη στήλη ΟνΣυμπτώματος αποθηκεύονται τα συμπτώματα του ασθενή (Πιν.13). Επίσης αναφέρεται και η ημερομηνία εμφάνισης.

SYMPTOMS				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΣυμπτώματος	int		PRIMARY KEY	IDENTITY(1,1)
ΟνΣυμπτώματος	nvarchar(250)	✓		
ΗμΕμφΣυμπτώματος	datetime	✓		Ημ εμφάνισης

Πιν. 13 Πίνακας SYMPTOMS

**«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»**

Ιωάννα Ζορμπά

Ακολουθεί ο πίνακας MEDICATION όπου στο συγκεκριμένο πίνακα θα αποθηκεύεται η φαρμακευτική αγωγή που δίνει ο Ιατρός και πληροφορίες όπως είναι η ημερομηνία έναρξης και λήξης της φαρμακευτικής αγωγής και η δοσολογία, την οποία θα πρέπει να ακολουθήσει ο Νοσηλευτής για τον κάθε ασθενή (Πιν. 14).

MEDICATION				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΦαρμΑγωγής	int		PRIMARY KEY	IDENTITY(1,1)
ΟνΦαρμΑγωγής	nvarchar(250)	✓		
ΗμΕνΦαρΑγωγής	datetime	✓		
ΗμΛηξΦαρΑγωγής	datetime	✓		
Δοσολογία	nvarchar(100)	✓		

Πιν. 14 Πίνακας MEDICATION

Στον πίνακα LABRESULTS αναφέρεται η ονομασία του εργαστηρίου όπου έχουν γίνει οι εξετάσεις, η περιγραφή της εξέτασης, το αποτέλεσμα, η μονάδα, μέτρησης, η κατάσταση που βρίσκεται αυτή τη στιγμή, για παράδειγμα μπορεί να είναι σε κατάσταση αναμονής μέχρι να τα διαβάσει ο Ιατρός και η ημερομηνία εξέτασης (Πιν.15).

LABRESULTS				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΕργΑποτελέσματος	int		PRIMARY KEY	IDENTITY(1,1)
ΟνΕργαστηρίου	nvarchar(100)	✓		
Περιγραφή	nvarchar(250)	✓		Όνομα εξέτασης γλυκόζης
Αποτέλεσμα	nvarchar(250)	✓		
ΜονΜέτρησης	nvarchar(100)	✓		
ΕΟ	nvarchar(100)	✓		
ΤιμέςΑναφοράςΑ	nvarchar(100)	✓		Εύρος τιμών για άντρες
ΤιμέςΑναφοράςΓ	nvarchar(100)	✓		Εύρος τιμών για γυναίκες
Κατάσταση(Status)	nvarchar(100)	✓		πχ αν αναγνώστηκαν
ΗμερΕξέτασης	datetime	✓		GETDATE()

Πιν. 15 Πίνακας LABRESULTS

Επόμενος πίνακας είναι ο REFERENCE στον οποίο αποθηκεύονται τα παραπεμπτικά που θα στέλνει ο Ιατρός στο κάθε Εργαστήριο. Η στήλη ΚωδΑσθενή είναι το Primary Key του πίνακα PATIENT και δεν πρέπει να παίρνει κενές τιμές (Πιν.16).

REFERENCE				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΠαρακεματικού	int		PRIMARY KEY	IDENTITY(1,1)
ΟνΕργαστηρίου	nvarchar(100)	✓		
ΟνΕξέτασης1	nvarchar(100)	✓		Όνομα εξέτασης πχ
ΟνΕξέτασης2	nvarchar(100)	✓		
ΟνΕξέτασης3	nvarchar(100)	✓		
ΟνΕξέτασης4	nvarchar(100)	✓		
ΟνΕξέτασης5	nvarchar(100)	✓		
ΟνΕξέτασης6	nvarchar(100)	✓		
ΟνΕξέτασης7	nvarchar(100)	✓		
ΟνΕξέτασης8	nvarchar(100)	✓		
ΟνΕξέτασης9	nvarchar(100)	✓		
ΟνΕξέτασης10	nvarchar(100)	✓		
ΗμΕντολής	datetime	✓		GETDATE()
Informed	bit	✓		
StatusExec	bit	✓		
ΚωδΑσθενή	int			FOREIGN KEY

Πιν. 16 Πίνακας REFERENCE

Ο πίνακας TEMPERATUREDIAGRAM περιέχει μια λίστα από τριώρες μετρήσεις της θερμοκρασίας του ασθενούς τις οποίες προσθέτει ο Νοσηλευτής. Στη στήλη Παρατηρήσεις έχει τη δυνατότητα να προσθέσει ότι νομίζει ότι θα βοηθήσει τον Ιατρό πχ κενώσεις και λοιπές νοσηλευτικές εργασίες (Πιν. 17).

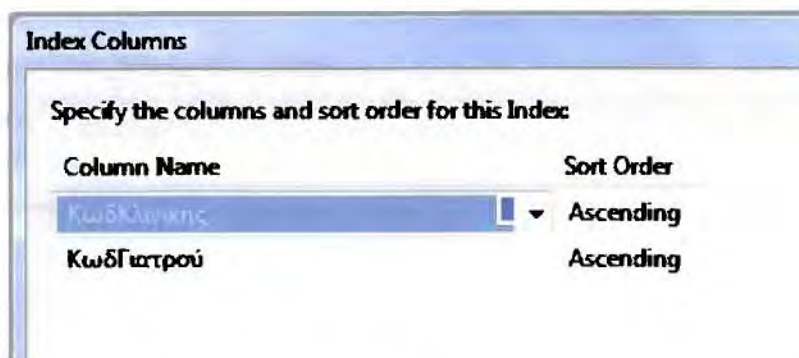
TEMPERATURE DIAGRAM				
COLUMN NAME	DATA TYPE	ALLOW NULLS		
ΚωδΘερμΔιαγράμματ	int		PRIMARY KEY	IDENTITY(1,1)
Ημερομηνία	datetime	✓		
06:00	int	✓		
09:00	int	✓		
12:00	int	✓		
15:00	int	✓		
18:00	int	✓		
21:00	int	✓		
00:00	int	✓		
03:00	int	✓		
Παρατηρήσεις	nvarchar(100)	✓		

Πιν. 17 Πίνακας TEMPERATURE DIAGRAM

Αφού δημιουργηθούν όλοι οι βασικοί πίνακες, πριν δημιουργηθούν οι πίνακες για τις N-M σχέσεις να αναφερθεί ότι είναι δυνατό να οριστούν οι σχέσεις 1-N αλλά οι συγκεκριμένες σχέσεις θα οριστούν παρακάτω αφού έχει ολοκληρωθεί η διαδικασία δημιουργίας όλων των πινάκων.

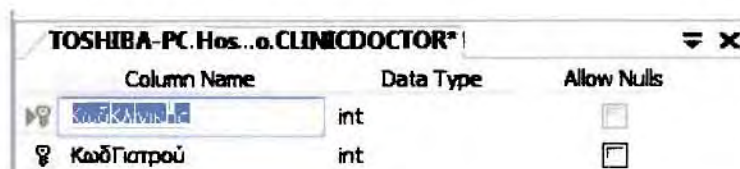
Οι επόμενοι πίνακες που έχουν δημιουργηθεί είναι για τη διαχείριση των σχέσεων N-M. Η σχεδίαση αυτών των πινάκων θέλει αρκετή προσοχή σε ότι αφορά τη σχέση των ξένων κλειδιών για να μην υπάρχουν συγκρούσεις. Οι σχέσεις που θα ορίσουμε μπορεί να δημιουργήσουν προβλήματα στο σχεδιασμό της Βάσης, στον σχεδιασμό της Εφαρμογής για κινητά αλλά ακόμα και στον τελικό χρήστη στην εισαγωγή των δεδομένων.

Ο πρώτος πίνακας είναι αυτός που συνδέει την Κλινική με τον Ιατρό. Η συγκεκριμένη σχέση, όπως και με τη σχέση μεταξύ Κλινικής και Νοσηλευτή, θα μπορούσε να γίνει με μια σχέση 1-N. Επιλέχθηκε όμως η σχέση N-M γιατί θα μπορούσε ένας Γιατρός ή ένας Νοσηλευτής να ανήκει σε παραπάνω από μια Κλινικές. Για να δημιουργηθεί αυτή η σχέση θα πρέπει ο πίνακας να περιέχει δυο Primary Keys, ένα για τον κάθε πίνακα. Για να δημιουργηθεί αυτή τη σχέση θα δημιουργηθεί κανονικά ο πίνακας με το SSMS. Στη συνέχεια ορίζεται μια από τις δυο στήλες ως Primary Key και για τη δεύτερη στήλη με δεξί κλικ μέσα στο περιθώριο και από τις δυνατότητες που εμφανίζονται επιλέγεται η Indexes /Keys. Στο καινούργιο παράθυρο επιλέγονται και οι δυο στήλες και επιλέγεται το OK (Εικ. 22).



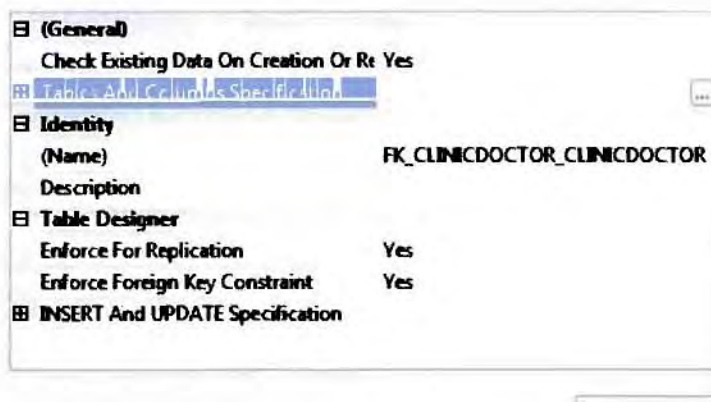
Εικόνα 22: SSMS, Δημιουργία δύο Primary Keys σε έναν πίνακα

Τώρα πλέον θα εμφανιστούν δυο Primary Keys και στις δυο στήλες (Εικ. 23). Δίνεται προσοχή στα Data Types τα οποία θα πρέπει να είναι ίδια, όπως εδώ int, με τα Types που έχουν οριστεί στους αρχικούς πίνακες, ειδικά γίνεται conflict και δεν δέχεται την εισαγωγή.



Εικόνα 23:SSMS, Εμφάνιση δύο Primary Keys σε έναν πίνακα

Στη συνέχεια κάνοντας πάλι δεξί κλικ θα επιλέγεται το Relationships. Θα αναδυθεί ένα νέο παράθυρο όπου θα χρειαστεί να γίνουν κάποιες ρυθμίσεις. Επιλέγοντας το Tables And Columns Specification (Εικ. 24), αναδύεται ένα νέο παράθυρο.



Εικόνα 24: SSMS, Relationships

Στο Relationship name η σχέση θα ονομαστεί FK_CLINICDOCTOR_CLINIC, στο Primary Key Table ανοίγοντας τις επιλογές θα προστεθεί ο πίνακας CLINIC και ακριβώς από κάτω θα επιλέγεται το κλειδί το οποίο έχει δοθεί με την ίδια ονομασία ώστε να είναι εύκολη η αναγνώριση των Primary Keys σε κάθε πίνακα. Στη δεύτερη επιλέγεται το NONE, και επιλέγεται το OK. Έτσι δημιουργήθηκε μια σχέση N-M. Το ίδιο όμως πρέπει να γίνει και για το δεύτερο πίνακα, δηλαδή θα δημιουργηθεί ένα Foreign Key και για τον πίνακα DOCTOR. Το Κλειδί αυτό ονομάζεται FK_CLINICDOCTOR_DOCTOR. Αυτό γίνεται για να οριστούν τα κλειδιά του συγκεκριμένου πίνακα να συνδέονται με τα Primary Keys των δυο πινάκων που συνδέονται (Εικ. 25). Με αυτόν τον τρόπο θα δημιουργηθούν και οι υπόλοιποι πίνακες και σχέσεις.

Με τον ίδιο τρόπο θα δημιουργηθούν και οι σχέσεις 1-N χωρίς όμως να απαιτείται να δημιουργηθούν επιπλέον πίνακες. Οι σχέσεις δημιουργούνται ακολουθώντας τη διαδικασία από το δεύτερο στάδιο δηλαδή πηγαίνοντας στον πίνακα που είναι το Foreign Key, και εφόσον επιλέγεται το Relationships, δηλώνεται με ποιον πίνακα θα συνδέεται και μέσω ποιās στήλης.



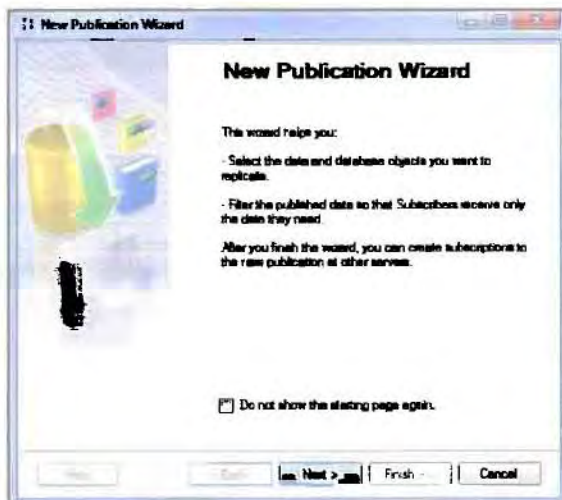
Εικόνα 25: SSMS, Σχέσεις μεταξύ Primary Keys

Οι επόμενοι πίνακες δημιουργήθηκαν για τη σύνδεση μεταξύ δυο πινάκων και για σχέσεις N-M. Ακολουθούνται κάθε φορά τα ίδια βήματα όπως και με τις σχέσεις μεταξύ των πινάκων CLINIC – DOCTOR.

CLINICNURSE (int ΚωδΚλινικής, int ΚωδΝοσηλευτή)
DRUGLISTMEDICATION (int ΚωδΦαρμαγωγής, int ΚωδΛισΦαρμάκου)
EXAMINATIONSBIOCREFERENCE (int ΚωδΕξέτασης, int ΚωδΠαραπεμπτικού)
ICD10ALLERGIES (int ΚωδICD10, int ΚωδΠινΑλλεργίας)
ICD10DIAGNOSIS (int ΚωδICD10, int ΚωδΔιάγνωσης)
ICD10SYMPTOMS (int ΚωδICD10, int ΚωδΣυμπτώματος)
LABRESULTSEXAMINATIONS (int ΚωδΕργΑποτελέσματος, int ΚωδΕξέτασης)
PATIENTALLERGIES (int ΚωδΑσθενή, int ΚωδΠινΑλλεργίας)
PATIENTCLINIC (int ΚωδΑσθενή, int ΚωδΚλινικής)
PATIENTDIAGNOSIS (int ΚωδΦαρμαγωγής, int ΚωδΔιάγνωσης)
PATIENTDOCTOR (int ΚωδΑσθενή, int ΚωδΓιατρού)
PATIENTLABRESULTS (int ΚωδΑσθενή, int ΚωδΕργΑποτελέσματος)
PATIENTMEDICATION (int ΚωδΑσθενή, int ΚωδΦαρμαγωγής)
PATIENTNURSE (int ΚωδΑσθενή, int ΚωδΝοσηλευτή)
PATIENTSYMPTOMS (int ΚωδΑσθενή, int ΚωδΣυμπτώματος)
PATIENT THERMDIAGRAM (int ΚωδΑσθενή, int ΚωδΘερμΔιαγράμματος)
PATIENTWARD (int ΚωδΑσθενή, int ΚωδΘαλάμου)



Εικόνα 27: SSMS, Replication



Εικόνα 28: New Publication Wizard

Την πρώτη φορά που θα εκτελεστεί το Wizard θα πρέπει να οριστεί αν θα χρησιμοποιηθεί ο συγκεκριμένος Server να λειτουργήσει σαν **Distributor**. Εδώ επιλέγεται η πρώτη επιλογή (Εικ. 29).



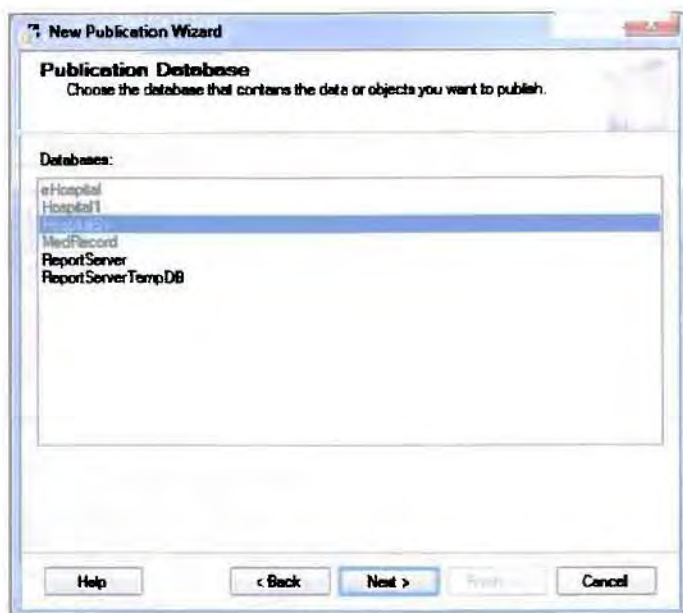
Εικόνα 29: Δημιουργία Distributor

Έπειτα θα εμφανιστεί η επιλογή για το που θα αποθηκευτούν τα δεδομένα και εμφανίζει ένα μονοπάτι. Θα μπορούσε ο φάκελος να δημιουργηθεί στη συγκεκριμένη θέση που δίνει το Wizard. Ο φάκελος όμως έχει δημιουργηθεί στον C: οπότε αυτό που πρέπει να γίνει είναι να αλλαχθεί η διαδρομή και στη συγκεκριμένη θέση να μπει η διαδρομή του φακέλου, δηλαδή \\computer\snapshot όπου computer είναι το όνομα του υπολογιστή και snapshot ο φάκελος (Εικ. 30)



Εικόνα 30: Δημιουργία Snapshot Φακέλου

Στο επόμενο παράθυρο ζητείται να επιλεγεί ποια Βάση θα δημοσιευθεί και επιλέγεται η HospitaSv (Εικ. 31).



Εικόνα 31: Επιλογή Βάσης Δεδομένων

Στη συνέχεια επιλέγεται η μέθοδος συγχρονισμού Merge Publication (Εικ. 32)



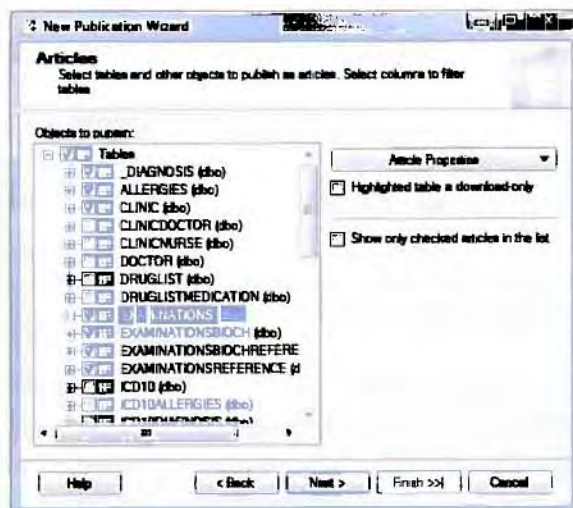
Εικόνα 32: Επιλογή Μεθόδου Συγχρονισμού

Αμέσως μετά θα πρέπει να καθοριστεί σε ποιές εκδόσεις Server θα χρησιμοποιηθεί. Στο σημείο αυτό να διευκρινιστεί ότι μπορούν να δημιουργηθούν όσες **Publications** χρειάζονται ανάλογα με τις ανάγκες κάθε φορά. Μια **Publication** μπορεί να χρησιμοποιηθεί και σε άλλες εφαρμογές, αλλά πρέπει να προσεχθεί να μην αλλαχθούν ορισμένες από τις επιλογές της προηγούμενης **Publication**. Υπάρχει επίσης η δυνατότητα να επιλεγθούν πάνω από μια εκδόσεις Server (Εικ. 33).



Εικόνα 33: Επιλογή εκδόσεων Server για Subscribers

Στο επόμενο παράθυρο το **Articles** επιλέγονται ποιοι πίνακες θα χρειαστούν για την εφαρμογή (Εικ. 34). Θα ήταν δυνατό να φορτωθούν όλοι οι αλλά δεν είναι καλός σχεδιασμός όταν η Βάση που έχει δημιουργηθεί είναι πολύ μεγάλη και με πολλά δεδομένα γιατί χρειάζεται πολύς χρόνος για να φορτωθεί από την εκάστοτε εφαρμογή.



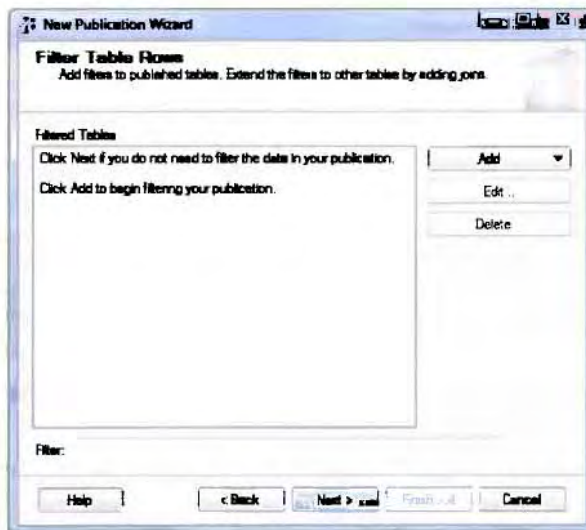
Εικόνα 34: Επιλογή Πινάκων για Publication

Στη συνέχεια δίνεται η πληροφορία ότι στη συγκεκριμένη Publication θα δημιουργηθεί μια επιπλέον στήλη σε κάθε πίνακα. Αυτή η στήλη χρησιμοποιείται για τη σύνδεση μεταξύ των δυο βάσεων (Εικ. 35)



Εικόνα 35: Ενημέρωση προσθήκης στήλης για το συγχρονισμό

Στο παράθυρο Filter Table Rows δίνεται η δυνατότητα να φιλτραριστούν ορισμένα από τα δεδομένα με εντολές SQL (Εικ. 36).



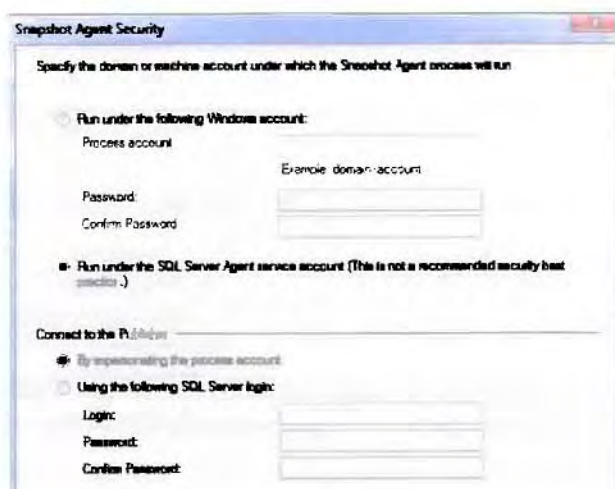
Εικόνα 36: Φιλτράρισμα δεδομένων

Στο επόμενο παράθυρο θα επιβεβαιώνεται ότι θα δημιουργηθεί ένας φάκελος, μέσα στο φάκελο που είχε δημιουργηθεί πριν, δηλαδή τον snapshot (Εικ. 37).



Εικόνα 37: Επιβεβαίωση Δημιουργίας φακέλου Snapshot

Στο παράθυρο Snapshot Agent Security θα επιλέγεται να τρέξει κάτω από τον SQL Server Agent service (Εικ. 38)



Εικόνα 38: Επιλογή λογαριασμού SQL Server Agent

Και θα εμφανιστεί στο παράθυρο όπως φαίνεται στην εικόνα (Εικ.39).



Εικόνα 39: Επιβεβαίωση επιλογής Λογαριασμού

Στο παράθυρο **Wizard Actions** επιλέγεται να δημιουργηθεί η **Publication** (Εικ. 40).



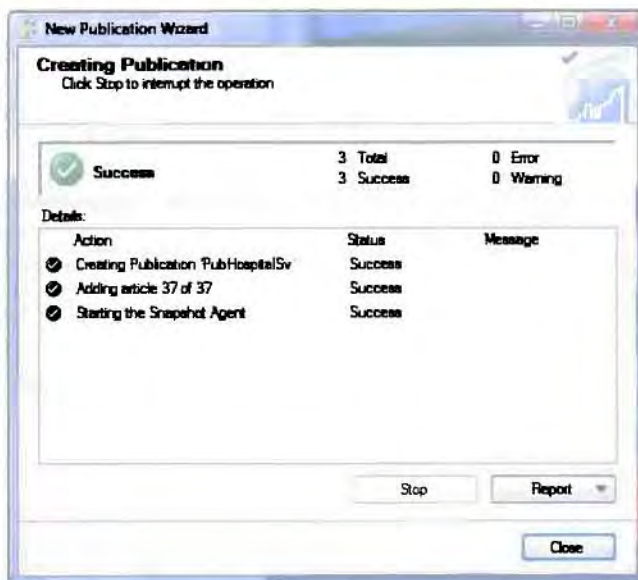
Εικόνα 40: Επιλογή δημιουργίας της Publication

Δίνεται το όνομα της **Publication** (Εικ. 41). Επίσης δίνονται πληροφορίες όσον αφορά τους πίνακες που έχουν επιλεγθεί να προστεθούν.



Εικόνα 41: Ονομασία Publication

Και τέλος αφού όλα τα βήματα έχουν γίνει σωστά, εμφανίζεται η πληροφορία για το αν έχει ολοκληρωθεί με επιτυχία η διαδικασία της **Publication** (Εικ. 42), και στον **Object Explorer** στην τοποθεσία με της **Local Publications** θα εμφανιστεί η **Publication** που έχει δημιουργηθεί.



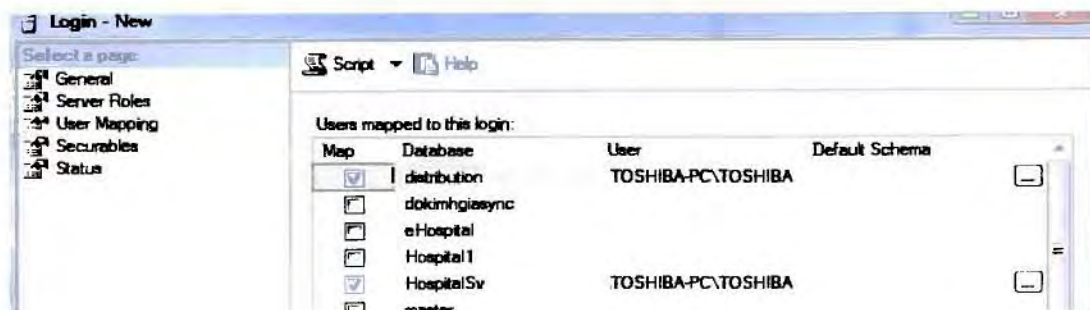
Εικόνα 42: Επιβεβαίωση δημιουργίας Publication

7.2 Logins

Πριν προχωρήσει η διαδικασία της **Subscription** θα δοθούν ορισμένα δικαιώματα σε χρήστες. Πηγαίνοντας στα **Security** -> **Login**-> **New Login** στην **General** σελίδα από search επιλέγεται ο χρήστης για τον οποίο θα αποδοθούν τα δικαιώματα (Εικ. 43) και στην **User Mapping** επιλέγονται οι δυο Βάσεις (Εικ. 44) και ακριβώς από κάτω στο **Database role** επιλέγεται το **db_owner**. Το ίδιο θα γίνει και για τον χρήστη **TOSHIBA-PC\MSR**.



Εικόνα 43: εισαγωγή χρηστών για πρόσβαση στην εφαρμογή



Εικόνα 44: Δικαιώματα σε χρήστες για Βάσεις

7.3 Synchronization

Η επόμενη διαδικασία που θα πρέπει να ακολουθήσει είναι να γίνει η συγκεκριμένη βάση διαθέσιμη ώστε οι χρήστες της εφαρμογής να έχουν πρόσβαση σε αυτή. Κάνοντας δεξί κλικ πάνω στην **Publication** επιλέγεται το **Configure Web Synchronization** και θα εμφανιστεί ένα νέο Wizard (Εικ. 45) μέσω του οποίου θα δημιουργηθεί μια virtual διεύθυνση στον IIS και παράλληλα θα δοθούν ορισμένες άδειες στη διεύθυνση αυτή. Τα βήματα που ακολουθούν έγιναν μέσω της βιβλιοθήκης αρχείων της msdn.microsoft.com [22].



Εικόνα 45: Synchronization Wizard

Η εφαρμογή θα λειτουργεί σε κινητές συσκευές και ο τύπος του Server είναι ο **SQL Server Compact** (Εικ. 46).



Εικόνα 46: Επιλογή SQL Server Compact για Subscriber Type

Στο επόμενο παράθυρο επιλέγεται ο Server που τρέχει τον IIS, επιλέγεται το **Create a**

new virtual directory, στο πλαίσιο θα εμφανιστεί το **Server node** το οποίο αν επεκταθεί θα εμφανιστεί το **Default Web Site**. Αν είχαν δημιουργηθεί και άλλα **Web Sites** θα εμφανιζόταν και αυτά και έπειτα γίνεται η επιλογή σε ποιο από αυτά θα δημιουργηθεί η εικονική διεύθυνση (Εικ. 47).



Εικόνα 47: Επιλογή Εικονικής Διεύθυνσης

Στο επόμενο παράθυρο γράφεται το ψευδώνυμο και το μονοπάτι που θα χρησιμοποιηθεί για την συγκεκριμένη εικονική διεύθυνση(Εικ. 48).



Εικόνα 48: Εισαγωγή ψευδωνύμου για πρόσβαση στην Εικονική Διεύθυνση

Θα ζητηθεί η εξουσιοδότηση για τη δημιουργία του φακέλου (Εικ. 49) που ουσιαστικά είναι η εικονική διεύθυνση.



Εικόνα 49: Εξουσιοδότηση δημιουργίας φακέλου Εικονικής Διεύθυνσης

Για την ανταλλαγή των δεδομένων μεταξύ των βάσεων απαιτείται ένα .dll αρχείο. Αυτό είναι το **sqlcesa35.dll** ο οποίος είναι ο **Server Agent** και θα προστεθεί αυτόματα στον φάκελο της εικονικής διεύθυνσης(Εικ. 50).



Εικόνα 50: Εισαγωγή του .dll αρχείου

Στο επόμενο βήμα θα ερωτηθεί για το αν θα χρησιμοποιηθεί ασφαλής επικοινωνία **SSL**, και επιλέγεται η πρώτη δυνατότητα(Εικ. 51).



Εικόνα 51: Καθορισμός καναλιού ασφαλούς λειτουργίας SSL

Στο επόμενο παράθυρο ζητείται να οριστούν συγκεκριμένοι χρήστες οι οποίοι θα έχουν πρόσβαση στον **Server**. Η επιλογή είναι να συνδεόνται ανώνυμα(Εικ. 52). Και στο αμέσως επόμενο παράθυρο μέσω ποιού χρήστη θα γίνεται η σύνδεση αυτή(Εικ. 53).



Εικόνα 52: Επιλογή σύνδεσης χρηστών



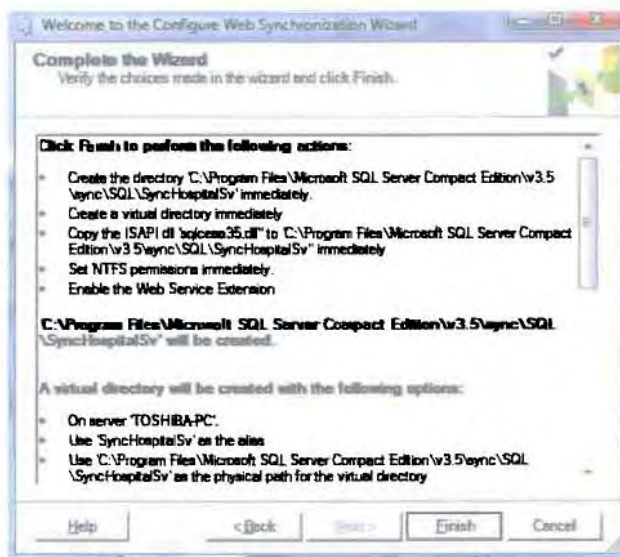
Εικόνα 53: Καθορισμός ανώνυμης πρόσβασης χρηστών

Στο επόμενο πλαίσιο εμφανίζεται η διεύθυνση του φακέλου που είχε δημιουργηθεί για να αποθηκευθεί η **Publication PubHospitalSv**(Εικ. 54).



Εικόνα 54: Καθορισμός μονοπατιού δικτύου και φακέλου

Λίγο πριν ολοκληρωθεί η διαδικασία του συγχρονισμού επαληθεύονται οι επιλογές και επιλέγεται Finish(Εικ. 55).



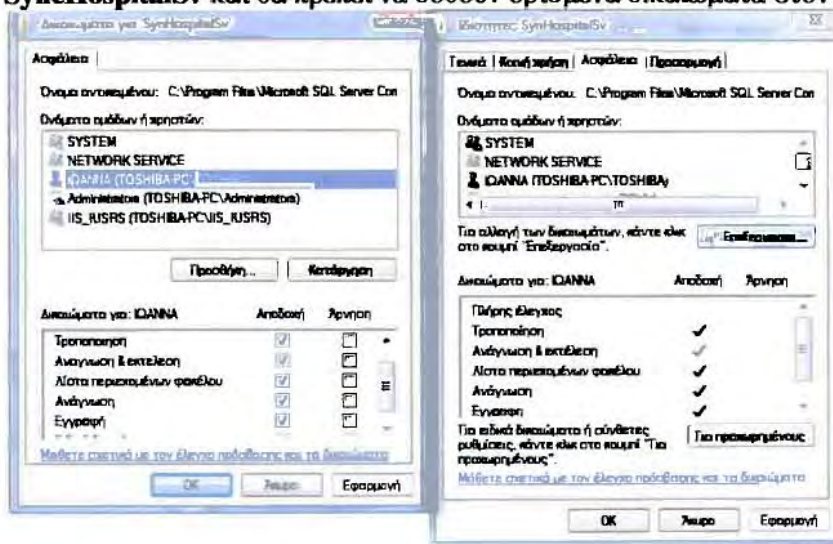
Εικόνα 55:Επαλήθευση επιλογών για Συγχρονισμό

Τέλος αν γίνει η διαδικασία σωστά εμφανίζεται το παράθυρο που επιβεβαιώνει ποιες λειτουργίες έχουν ολοκληρωθεί (Εικ. 56).



Εικόνα 56: Επιβεβαίωση δημιουργίας διαδικασίας Συγχρονισμού

Τώρα αν γίνει μετάβαση στο συγκεκριμένο μονοπάτι **C:\Program Files\Microsoft SQL Server Compact Edition\v3.5\sync\SQL** φαίνεται ότι δημιουργήθηκε ο φάκελος **SyncHospitalSv** και θα πρέπει να δοθούν ορισμένα δικαιώματα στον χρήστη (Εικ. 57).



Εικόνα 57: Απόδοση δικαιωμάτων στον φάκελο SyncHospitalSv

Πηγαίνοντας σε οποιονδήποτε Browser και γράφοντας την παρακάτω διεύθυνση <http://localhost/SyncHospitalSv/sqlcesa35.dll> θα εμφανιστεί το παρακάτω (Εικ. 58), το οποίο δείχνει ότι λειτουργεί σωστά.



Εικόνα 58: Ηλεκτρονική Διεύθυνση SyncHospitalSv

7.4 SQL server compact subscription

Το επόμενο βήμα είναι να γίνουν οι απαιτούμενες ρυθμίσεις από πλευράς του **SQL Server Compact** για τον **Subscriber**. Για να μπορεί να γίνει αυτό θα ολοκληρωθεί η διαδικασία του **Replication Wizard** πάλι μέσω του **SSMS**. Τα βήματα που ακολουθούν έγιναν μέσω της βιβλιοθήκης αρχείων της msdn.microsoft.com [23].

Ανοίγοντας το **SSMS** αυτή τη φορά επιλέγεται να γίνει σύνδεση στον **SQL Server Compact Edition** και επιλέγεται η Βάση μας η οποία αυτή τη φορά είναι η **Hospital** (Εικ. 59)



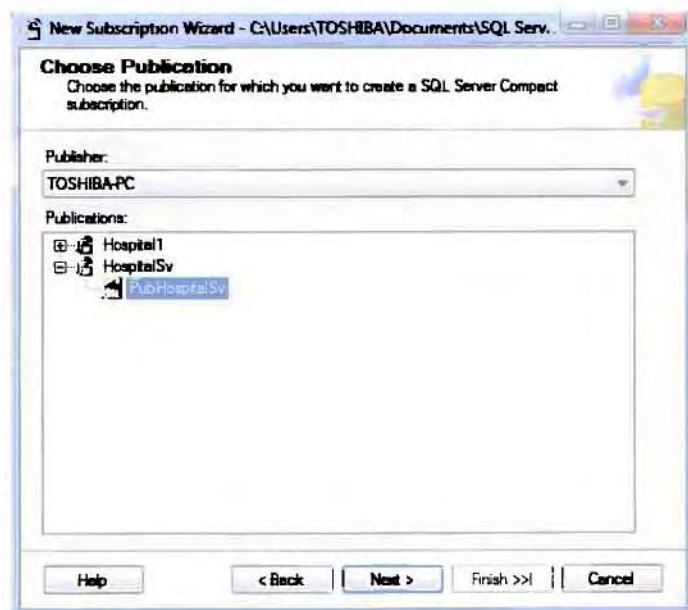
Εικόνα 59: Σύνδεση με SQL Server Compact Edition

Στον **Object Explorer** πηγαίνοντας **Replication** -> **Subscriptions** -> **New Subscription** εμφανίζεται το **Subscription Wizard**(Εικ.60)



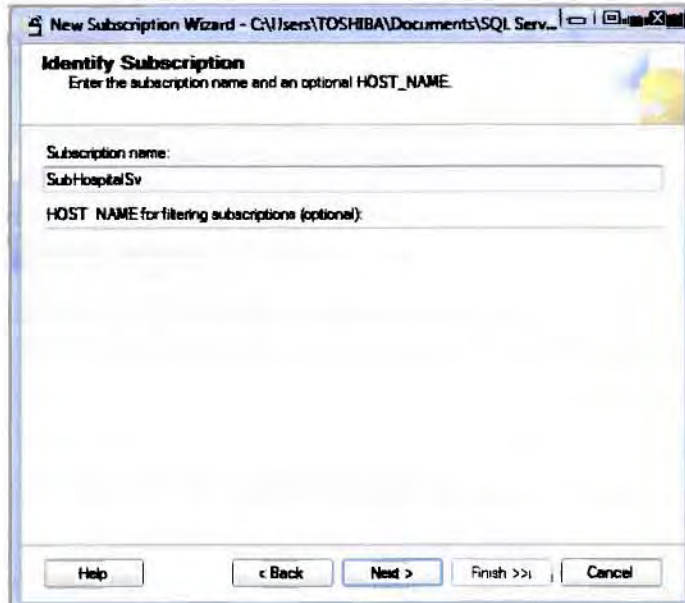
Εικόνα 60: Εμφάνιση του Subscription Wizard

Στο επόμενο παράθυρο επιλέγοντας τον Server που έχει την Publication που δημιουργήθηκε, επεκτείνεται το node και επιλέγεται η συγκεκριμένη Publication (Εικ. 61).



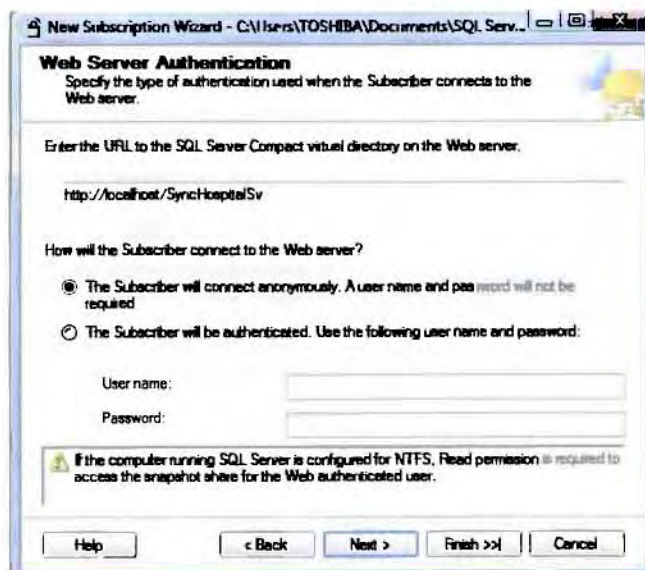
Εικόνα 61: Επιλογή Server και Publication

Στο πλαίσιο γράφεται το όνομα της **Subscription** που πρόκειται να δημιουργηθεί (Εικ. 62)



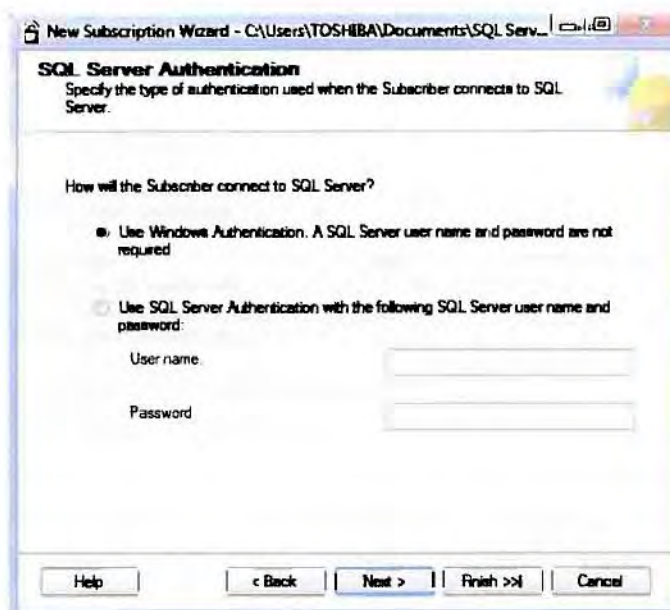
Εικόνα 62 Καθορισμός ονομασίας Subscription

Έπειτα γράφεται η διεύθυνση της εικονικής διεύθυνσης που δημιουργήθηκε πριν(Εικ. 63)



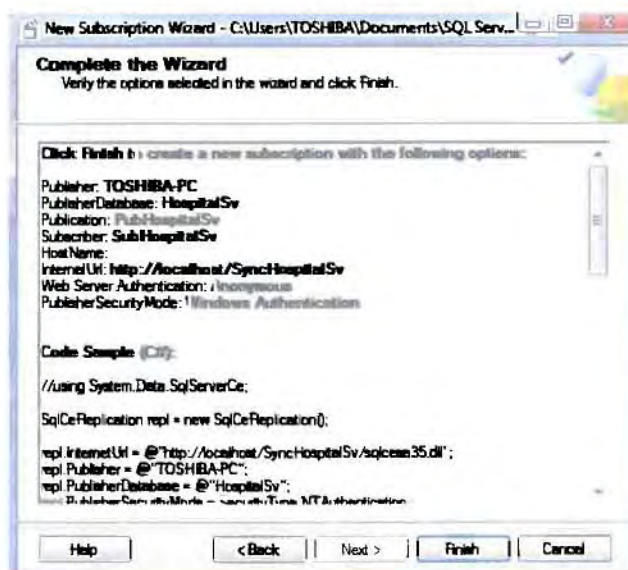
Εικόνα 63: Καθορισμός URL της Εικονικής Διεύθυνσης

Επιλέγεται η πιστοποίηση θα χρησιμοποιηθεί, η οποία θα είναι η **Windows Authentication** και μετά next (Εικ. 64)



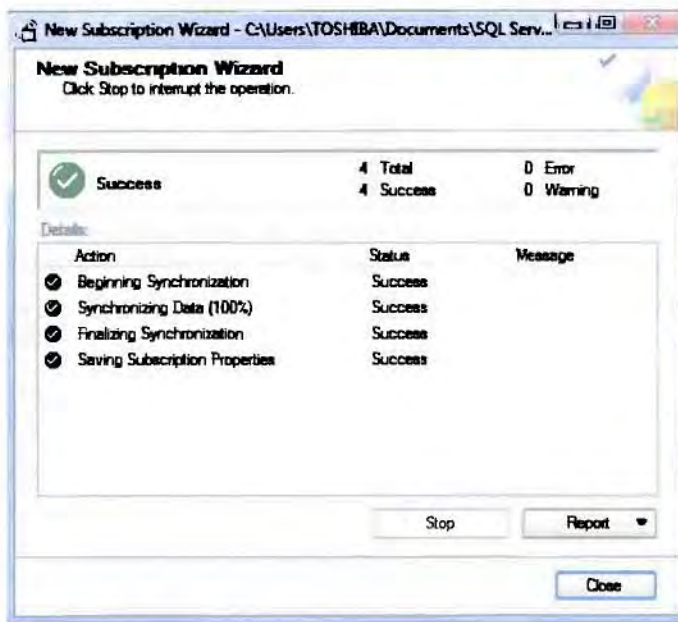
Εικόνα 64: Επιλογή πιστοποίησης για τον Subscriber

Λίγο πριν το τέλος μπορούν να επαληθευτούν οι επιλογές που έχουν γίνει κατά τη διάρκεια των βημάτων του Wizard (Εικ. 65) και μετά Finish.



Εικόνα 65: Επαλήθευση επιλογών του Subscriber

Στο τέλος εμφανίζεται το παράθυρο όπου φαίνεται αν ολοκληρώθηκε η διαδικασία με επιτυχία (Εικ. 66)



Εικόνα 66: Επιβεβαίωση επιτυχούς δημιουργίας Subscription

Πηγαίνοντας στη Βάση φαίνεται ότι έχει προστεθεί στο **Subscription** η συγκεκριμένη σύνδεση, και επεκτείνοντας το node με τους πίνακες θα εμφανιστούν οι πίνακες που υπήρχαν στη Βάση **HospitalSv** (Εικ. 65).



Εικόνα 67: SSMS, εμφάνιση σύνδεσης Subscription – Publication

Η διαδικασία που ακολουθήθηκε δημιούργησε μια σύνδεση μεταξύ της κεντρικής ΒΔ που βρίσκεται στον SQL Server και της ΒΔ που βρίσκεται στον SQL Server CE της κινητής συσκευής. Ο συγχρονισμός μεταξύ των δεδομένων θα γίνεται άμεσα, εφόσον υπάρχει βέβαια δίκτυο για να μπορεί να γίνει η σύνδεση. Σε περίπτωση που η σύνδεση δεν είναι εφικτή, λόγω έλλειψης δικτύου ή οποιοδήποτε άλλου λόγου, η ανταλλαγή των δεδομένων θα εκτελεστεί με την πρώτη ευκαιρία που θα δοθεί από πλευράς των Server. Δεν χάνονται οι αλλαγές που έχουμε κάνει απλά δεν θα γίνουν άμεσα. Παράδειγμα αν έχει φορτωθεί ωη Βάση στην κινητή συσκευή, μπορεί να γίνει οποιαδήποτε αλλαγή και τροποποίηση επιτρέπεται, και μόλις γίνει εφικτή η σύνδεση να γίνει και ο συγχρονισμός με τη ΒΔ στον SQL Server. Επίσης η σύνδεση μπορεί να ολοκληρωθεί και μέσω USB θύρας για τις περιπτώσεις πολύωρης διακοπής του δικτύου.

Έχοντας ολοκληρώσει τη διαδικασία της Merge Replication το επόμενο βήμα θα είναι να δημιουργηθεί η εφαρμογή για τις κινητές συσκευές, μέσω του Visual Studio.



8 ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΦΟΡΗΤΗΣ ΣΥΣΚΕΥΗΣ

Όπως αναφέρθηκε και πριν οι χρήστες θα είναι Γιατροί και Νοσηλευτές, οι οποίοι θα έχουν πρόσβαση στην εφαρμογή και οι λειτουργίες που θα θέλουν επιτελέσουν θα είναι η ανάγνωση της ΒΔ, εισαγωγή δεδομένων και η ανανέωση παλαιότερων πληροφοριών.

Αυτό που ενδιαφέρει όλους τους εργαζομένους είναι η λειτουργική προσπέλαση των πληροφοριών, χωρίς να περιπλανιόνται σε δυσανάγνωστα μενού, χάνοντας πολύτιμο χρόνο. Έτσι θα πρέπει να σκεφτούμε ότι η εφαρμογή αυτή θα χρησιμοποιηθεί από το ιατρικό προσωπικό μιας Νοσοκομειακής Μονάδας, και θα πρέπει να είναι έτσι σχεδιασμένη ώστε το προσωπικό να κερδίζει χρόνο και να περιορίζεται η γραφειοκρατία.

Οι λειτουργίες από την πλευρά των ιατρών είναι να έχουν πρόσβαση σε συγκεκριμένα δεδομένα που αφορούν το **Ιστορικό** του ασθενή όπως διαγνώσεις, τη φαρμακευτική αγωγή που ακολουθεί ο ασθενής, αλλεργίες που τυχόν έχει. Επίσης να ενημερωθεί για το **Θερμομετρικό Διάγραμμα**, τα **Εργαστηριακά Αποτελέσματα** και να μπορεί να στέλνει **Παραπεμπτικά** προς τα εργαστήρια.

Η τελευταία λειτουργία είναι ιδιαίτερα σημαντική γιατί έτσι μπορεί να περιοριστεί ο χρόνος που αφορά τις εξετάσεις, τα εργαστήρια θα μπορούν να ετοιμάζονται για την επόμενη εξέταση και δεν απαιτείται κάποιος ενδιάμεσος, πέραν της εφαρμογής, για να μεταφερθεί το παραπεμπτικό προς τα εργαστήρια.

Οι λειτουργίες από πλευράς των Νοσηλευτών είναι να ανανεώνουν το **Θερμομετρικό Διάγραμμα**, να μπορούν να αναφέρουν οποιαδήποτε πληροφορία προς τον Ιατρό, παράδειγμα πως έχει αντιδράσει ο ασθενής στη φαρμακευτική αγωγή και να πληροφορούνται για την φαρμακευτική αγωγή που θα πρέπει να ακολουθήσει ο ασθενής, μέσω της φόρμας **ΠΑΡΑΤΗΡΗΣΕΙΣ**.

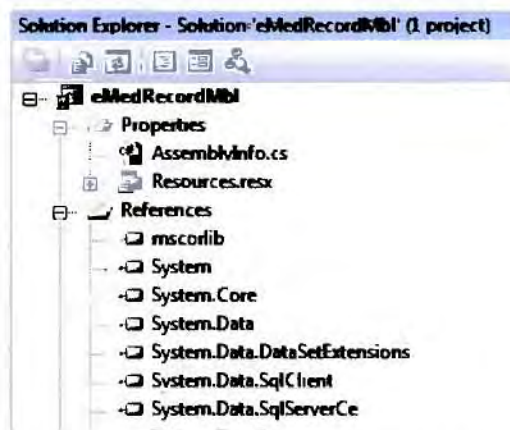
8.1 Δημιουργία Project eMedRecordMbl

Για να ξεκινήσει το πρόγραμμά θα γίνει η μετάβαση στο **Visual Studio** μέσω της έναρξης. Ακολουθώντας τη διαδρομή **File -> New -> Project** επιλέγεται το **Smart Device** και στο πλαίσιο δίνεται το όνομα του project, εδώ δημιουργήθηκε το **eMedRecordMbl**. Στο ίδιο παράθυρο πάνω αριστερά επιλέγεται το **.NET Framework 3.5** το οποίο όπως ειπώθηκε είναι ένα εργαλείο που απαιτείται για τη συγκεκριμένη εφαρμογή (Εικ. 68).



Εικόνα 68: Επιλογή .NET Framework

Στο επόμενο παράθυρο επιλέγεται σε ποια πλατφόρμα θα δουλέψει η εφαρμογή, επιλέγεται την **Windows Mobile 6 Professional SDK** και από τα **Templates** επιλέγουμε **Device Application**. Αμέσως θα δημιουργηθεί ένα νέο project με μια καινούργια φόρμα. Δεξιά στον **Solution Explorer**(Εικ. 69) εμφανίζονται οι πληροφορίες που αφορούν το συγκεκριμένο project.



Εικόνα 69: Πληροφορίες του project eMedRecordMbl

Αριστερά του παραθύρου βρίσκεται το **toolbox** όπου από εκεί θα χρησιμοποιηθούν διάφορα εργαλεία για την ανάπτυξη και σχεδίαση των φορμών, τα οποία είναι δυνατό να προστεθούν στις φόρμες με *drag and drop*. Δεξιά υπάρχουν τα *properties* του κάθε αντικείμενου που κάθε φορά επιλέγονται να χρησιμοποιηθούν.

Αυτό που πρέπει να γίνει είναι να προστεθεί στο πρόγραμμα ο Emulator. Πηγαίνοντας στο **Tools** επιλέγεται το **Device Emulator Manager** και στη συνέχεια με ποια συσκευή θα γίνει η εργασία κάθε φορά, και με δεξιά κλικ επιλέγεται η σύνδεση με το **Windows Mobile Professional Emulator** και έτσι θα εμφανιστεί ο Emulator. Έπειτα μέσω του Πίνακα Ελέγχου γίνεται μεταφορά στο Κέντρο Συσκευών των **Windows Mobile**. Στις ρυθμίσεις σύνδεσης θα επιλεγεί το **DMA(Direct Memory Access)**, έτσι επιτρέπεται στον Emulator να επικοινωνεί με τον υπολογιστή. Πηγαίνοντας ξανά στο **Device Emulator Manager** αφού έχει γίνει η σύνδεση, κάνοντας δεξιά κλικ πάλι στον Emulator αυτή τη φορά εμφανίζεται η επιλογή **Cradle**. Με αυτή την επιλογή συνδέεται ο Emulator με το **Windows Mobile Device Center**.

Εφόσον η εφαρμογή αφορά κινητές συσκευές, αυτό που ενδιαφέρει είναι να επιλεγεί το **References** και να προστεθούν οι βιβλιοθήκες **System.Data.SqlServerCe** η οποία περιέχει κλάσεις που παρέχουν πρόσβαση στον **SQL Server Compact Edition** και η **System.Data.SqlClient** η οποία περιέχει κλάσεις που αφορούν τον **SQL Server**.

Πατώντας διπλό κλικ μέσα στο πλαίσιο της φόρμας πηγαίνει στον editor όπου εκεί θα προστεθεί ο κώδικας. Στο πάνω μέρος της φόρμας εμφανίζονται οι βιβλιοθήκες που χρησιμοποιούνται. Στην αρχή της κάθε φόρμας προστίθενται οι βιβλιοθήκες ως εξής:

```
using System.Data.SqlClient;  
using System.Data.SqlServerCe;
```

Έτσι κάθε φορά που απαιτείται να προστεθεί μια κλάση θα μπορεί η εφαρμογή να την εντοπίσει. Αν εκ παραδρομής δεν αναφερθούν στην αρχή του κώδικα οι βιβλιοθήκες, δεν αναγνωρίζονται οι κλάσεις και εντοπίζεται λάθος, όπου γίνεται αντιληπτό ότι δεν έχει γίνει η αναφορά της εκάστοτε βιβλιοθήκης.

Όταν δημιουργείται ένα αντικείμενο, στη συγκεκριμένη περίπτωση η **Form1**, δημιουργείται ένας **Constructor**, ο οποίος είναι μια η βασική κλάση της φόρμας και έχει το όνομα του αντικείμενου που έχει προστεθεί. Ο **Constructor** της αρχικής φόρμας χρησιμοποιεί τη μέθοδο **-> InitializeComponent()**; ώστε να αρχικοποιεί το αντικείμενο (Εικ. 70) στο οποίο αναφέρεται.

```
using System.Text;
using System.Windows.Forms;

namespace eMedRecordMbl
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

Εικόνα 70: Constructor Φόρμας Form1

8.2 Φόρμες συσκευής

Η εφαρμογή αποτελείται από έντεκα φόρμες, οι οποίες είναι κοινές και για τους Ιατρούς και για τους Νοσηλευτές. Αυτό που ενδιαφέρει για το σχεδιασμό είναι να συμπεριληφθούν μόνο τα βασικά αντικείμενα, γιατί πρώτον ο χώρος είναι περιορισμένος και δεύτερο να είναι εύχρηστη η λειτουργία στους χρήστες. Δεν πρέπει να αγνοείται ότι η εφαρμογή απευθύνεται στο προσωπικό υγείας μιας νοσοκομειακής μονάδας, με την προϋπόθεση να χρησιμοποιείται κατά την επίσκεψη στους θαλάμους των ασθενών. Γι' αυτό δεν πρέπει οι χρήστες να είναι αντιμέτωποι με ένα περίπλοκο περιβάλλον.

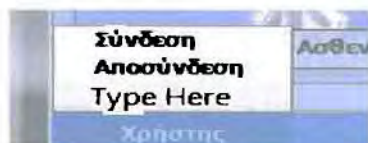
8.2.1 Αρχική φόρμα

Η **Form1** είναι η αρχική φόρμα της εφαρμογής, οποία στην αρχή δεν είναι ενεργοποιημένη. Στην φόρμα έχει προστεθεί ένα **picturebox**, ένα **button** για την επιλογή του ασθενή, όπου αυτή τη στιγμή δεν είναι ενεργοποιημένο, ένα **StatusBar** το οποίο ενημερώνει το χρήστη για το αν είναι συνδεδεμένος, ένα **button** για το συγχρονισμό με τη Βάση Δεδομένων του Server, ένα **menuItemSun** για τη σύνδεση του χρήστη, ένα **menuItemApos** για την αποσύνδεση και ένα **menuItemEx** για την έξοδο(Εικ.71).



Εικόνα 71: Αρχική φόρμα εφαρμογής

Τα **menuItem**s είναι δυνατό να διαχειριστούν κατευθείαν πάνω στη φόρμα και δεν χρειάζεται να επιλεγθούν από το **toolbox**(Εικ. 72).



mainMenu1

Εικόνα 72: MenuItem's φόρμας

Εφόσον η αρχική φόρμα δεν είναι ενεργοποιημένη θα πρέπει να δημιουργηθεί μια μέθοδος η **LogMeOut()** όπου θα καθορίζονται ποια αντικείμενα θα είναι ενεργοποιημένα, θα την καλεί το πρόγραμμα όταν φορτώνεται η φόρμα μας και αυτή η κλήση θα γίνεται μετά την αρχικοποίηση του αντικειμένου.

```
public Form1 ()  
{  
    InitializeComponent ();  
    LogMeOut ();  
}
```

8.2.1.1 Η μέθοδος LogMeOut()

Αυτή η μέθοδος θα αναφέρει ότι στο κείμενο του **statusBar1** θα αναγράφεται η πληροφορία προς τον χρήστη για το ότι δεν είναι συνδεδεμένος, το **menuItemSun** θα είναι

ενεργοποιημένο, το **MenuItemApos** θα είναι απενεργοποιημένο, το **pictureBox1** θα είναι απενεργοποιημένο και το **button** για την επιλογή του ασθενή θα είναι απενεργοποιημένο.

```
private void LogMeOut ()
{
    statusBar1.Text = "Δεν έχετε συνδεθεί";
    menuItemSun.Enabled = true;
    menuItemApos.Enabled = false;
    pictureBox1.Enabled = false;
    btnEpiIoghAsthenh.Enabled = false;
}
```

8.2.1.2 Η μέθοδος menuItemEx_Click()

Επιλέγοντας το συγκεκριμένο αντικείμενο το σύστημα θα καλέσει την έτοιμη μέθοδο **close()** για να τερματιστεί η εφαρμογή.

```
private void menuItemEx_Click(object sender, EventArgs e)
{
    this.Close();
    Application.Exit();
}
```

8.2.1.3 Η μέθοδος menuItemSun_Click()

Η συγκεκριμένη μέθοδος επιτελεί δύο διεργασίες. Στην πρώτη περίπτωση αυτό που θέλουμε απαιτείται είναι να φορτωθούν τα καινούργια δεδομένα στη Βάση της φορητής συσκευής. Το βασικό βήμα που είχε γίνει στη παράγραφο 7.4 ήταν να δημιουργηθεί το σχήμα της Βάσης στην φορητή συσκευή. Στην αρχή της δημιουργίας δεν υπάρχουν δεδομένα. Επειδή όμως τα δεδομένα στην Κεντρική Βάση αλλάζουν συνεχώς, κάθε φορά θα πρέπει να προστίθενται και στην φορητή συσκευή. Στον κώδικα που υπάρχει παρακάτω ο SQL Server CE ελέγχει αν υπάρχει η συγκεκριμένη Βάση και αν ναι να την ξαναδημιουργήσει ουσιαστικά με τα καινούργια δεδομένα τα οποία βρίσκονται στην Κεντρική Βάση. Έπειτα καλείται η συνάρτηση **Synchronize()** ώστε να συγχρονιστούν οι δύο Servers.

Επιπλέον θα καλέσει μια καινούργια φόρμα την **LogInForm**. Όμως πριν να φορτωθεί η νέα φόρμα ελέγχεται η επιστρεφόμενη τιμή ενός **dialog box** που βρίσκεται στην επόμενη φόρμα. Αν αυτή η τιμή είναι OK τότε θα καλέσει τη μέθοδο **LogMeIn()**.

```
private void menuItemSun_Click(object sender, EventArgs e)
{
    SqlCeReplication repl = new SqlCeReplication();
    try
    {
        repl.InternetUrl =
            "http://192.168.2.10/syncHospitalSv/sqlcesa35.dll";
        repl.Publisher = "[TOSHIBA-PC]";
        repl.PublisherDatabase = "[HospitalSv]";
        repl.PublisherSecurityMode =
            SecurityType.NTAuthentication;
        repl.Publication = "[PubHospitalSv]";
        repl.SubscriberConnectionString =
            @"Data Source='\\program files\emedrecordmbl\hospital.sdf'";
        repl.Subscriber = "[SubHospitalSv]";

        if (!System.IO.File.Exists(@"\program
```

```
        files\emedrecordmbl\hospital.sdf"))
    {
        repl.AddSubscription(AddOption.CreateDatabase);
    }
    repl.Synchronize();
    MessageBox.Show("Τα δεδομένα έχουν μεταφερθεί");
}

catch (System.Data.SqlServerCe.SqlCeException ex)
{
    MessageBox.Show(ex.Message);
}

finally
{
    repl.Dispose();
}
var formlog = new LogInForm();
if (formlog.ShowDialog() == DialogResult.OK)
{
    LogMeIn();
}
}
```

8.2.1.4 Η μέθοδος LogMeIn()

Δημιουργείται μια νέα μέθοδος τη **LogMeIn()** η οποία θα ενεργοποιήσει ή θα απενεργοποιήσει ορισμένες από τις λειτουργίες των αντικειμένων και θα ενημερώσει τον χρήστη μέσω του **statusBar1** για το αν έχει συνδεθεί στην εφαρμογή.

```
private void LogMeIn()
{
    statusBar1.Text = "Έχετε συνδεθεί";
    menuItemSun.Enabled = false;
    menuItemApos.Enabled = true;
    pictureBox1.Enabled = true;
    btnEpiloghAsthenh.Enabled = true;
}
```

8.2.1.5 Η μέθοδος menuItemApos_Click()

Η συγκεκριμένη μέθοδος καλεί τη **LogMeOut()** για την έξοδο από την εφαρμογή.

```
private void menuItemApos_Click(object sender, EventArgs e)
{
    LogMeOut();
}
```

8.2.1.6 Η μέθοδος btnEpiloghAsthenh_click()

Η συγκεκριμένη μέθοδος καλεί μια καινούργια φόρμα την **EpiloghAsthenhform** και μετά την εμφανίζει. Αυτό που πρέπει να προσεχθεί πριν γίνει η κλήση μιας καινούργιας φόρμας, ή οποιουδήποτε άλλου αντικείμενου, είτε να υπάρχει έτοιμο είτε να το δημιουργήσουμε εμείς. Έτσι πριν γίνει η κλήση της φόρμας ακολουθώντας το **project -> add**

-> **Windows Form** δίνεται το συγκεκριμένο όνομα στο αντικείμενο και έτσι.

```
private void btnEpiloghAsthenh_Click(object sender, EventArgs e)
{
    var EpiloghAsthenh = new EpiloghAsthenh();
    EpiloghAsthenh.Show();
}
```

8.2.1.7 Η μέθοδος `buttonApoth_Click`

Εφόσον ο χρήστης έχει τελειώσει με τις αλλαγές που θα έχει κάνει, θα επιστρέψει στην αρχική φόρμα για την επιλογή νέου ασθενή. Εκεί θα του δίνεται η δυνατότητα να ενημερώσει τη Κεντρική Βάση για τις αλλαγές που έχει κάνει, καλώντας τη `Synchronize()`.

```
private void buttonApoth_Click(object sender, EventArgs e)
{
    repl.Synchronize();
}
```

8.2.2 Φόρμα σύνδεσης

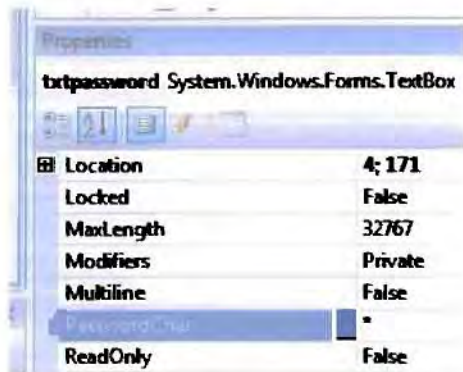
Στην φόρμα **LogInForm** ο χρήστης θα μπορεί να δώσει τα στοιχεία του για την εισαγωγή του στην εφαρμογή (Εικ. 73). Περιέχει ένα `textbox` για το `username` και ένα `textbox` για το `password` του χρήστη. Επίσης ένα `button` για να σβήσει αν θέλει ο χρήστης τα στοιχεία που έδωσε και δυο `menuItems`.



Εικόνα 73: Φόρμα LogInForm

Επειδή στο `textbox` που θα πληκτρολογείται το **Password** δεν είναι σωστό να

εμφανίζονται οι χαρακτήρες, πηγαίνοντας στα **properties** του αντικειμένου και στο **PasswordChar** προστίθεται ένας αστερίσκος (Εικ. 74).



Εικόνα 74: Properties αντικειμένου txtpassword

8.2.2.1 Η μέθοδος menuItem2_click()

Η μέθοδος **menuItem2()** αφορά το **menuItem** **Ακύρωση** το οποίο όταν θα επιλεγεί θα επιστρέψει στη φόρμα **Form1** ένα **DialogResult** του οποίου το αποτέλεσμα θα είναι **cancel**.

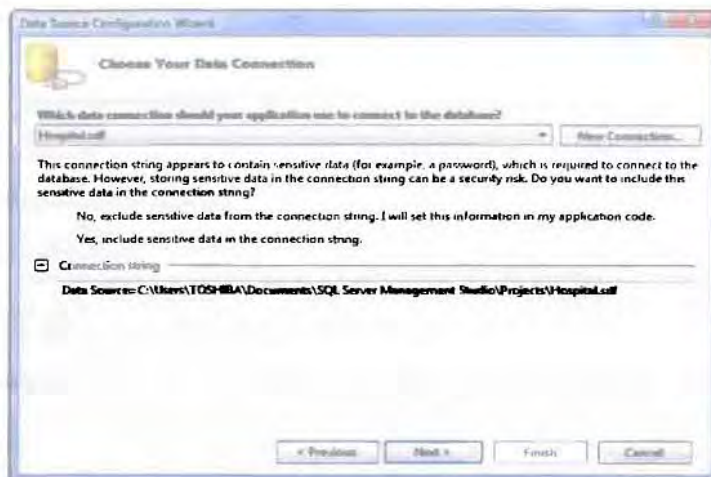
```
private void menuItem2_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.Cancel;
}
```

8.2.2.2 Η μέθοδος menuItem1_click()

Η μέθοδος **menuItem1_click()** αφορά το **menuItem** **Σύνδεση**. Αυτό που πρέπει να γίνει είναι διαβαστούν τα δεδομένα από τον πίνακα **PASSWORD** να γίνει έλεγχος και αν είναι σωστά να επιστρέψει στη φόρμα **Form1** το αποτέλεσμα **OK** του **DialogResult**. Είναι σωστά τα στοιχεία με ένα **MessageBox** ενημερώνεται ο χρήστης ότι είναι σωστά ή το αντίθετο.

Για να γίνει αυτό θα χρησιμοποιείται ένα **DataSet**, ένας **DataTable** και ένας **TableAdapter**. Ένα **DataSet** είναι ένας τύπος δεδομένων που απεικονίζει έναν πίνακα Βάσης Δεδομένων στην επιχειρησιακή λογική. Είναι ένα βασικό συστατικό της **ADO.NET** αρχιτεκτονικής και αποτελείται από μια συλλογή αντικειμένων τα **dataTables**. Το **DataTable** είναι ένας προσωρινός πίνακας, όπου αποθηκεύουμε τα δεδομένα που θέλουμε. Ο **TableAdapter** είναι αυτός που αναλαμβάνει τη σύνδεση με τη Βάση, εκτελεί **SQL queries** ή **procedures** και επιστρέφει στο **DataTable** τα δεδομένα. Ένας τρόπος είναι να αναπτυχθεί ο κώδικας τους με συγκεκριμένες εντολές και ένας άλλος τρόπος να τα δημιουργηθούν αυτόματα. Ο δεύτερος τρόπος θα αναλυθεί σε επόμενες φόρμες.

Για να δημιουργηθεί το **Dataset** του πίνακα **PASSWORD**, πηγαίνοντας στο **data Sources** -> **Add New Data Source** εμφανίζεται το **Data Source Configuration Wizard** όπου δίνονται τρεις επιλογές, **Database**, **Web Service** και **Object**. Επιλέγεται το **Database** και στο επόμενο βήμα σε ποια Βάση θα συνδεθεί (Εικ 75).



Εικόνα 75: Εισαγωγή Βάσης Δεδομένων στο project

Επεκτείνοντας το **Tables node** εμφανίζονται οι πίνακες της Βάσης (Εικ. 76), από εκεί θα επιλέγονται ποιοί πίνακες θα χρησιμοποιηθούν. Όταν ολοκληρωθεί το wizard θα πρέπει να γίνει ένα **Build** της εφαρμογής ώστε να δεχθεί τις αλλαγές. Πηγαίνοντας τώρα στο toolbox θα εμφανιστούν δύο εργαλεία. Το **PasswordDataset** και ο **PasswordAdapter**. Τώρα με **Drag and Drop** μπορούν να προστεθούν στην επιφάνεια κάτω από την φόρμα. Επιλέγοντας το **dataset** θα επικολληθεί και ο **adaptor**.



Εικόνα 76: Επιλογή πινάκων για τη δημιουργία του PasswordDataSet

Οπότε δημιουργείται ένας **PASSWORDDataTable dt** του πίνακα **PASSWORD** και ο **passwordTableAdapter1** θα εκτελέσει τη μέθοδο **Fill()** για να γεμίσει με δεδομένα το συγκεκριμένο **DataTable**. Για κάθε γραμμή **dr** του πίνακα θα ελέγξει αν οι τιμές τους είναι ίδιες με το περιεχόμενο του εκάστοτε **textbox**, και θα εμφανίσει το ανάλογο **MessageBox**.

```
private void menuItem1_Click(object sender, EventArgs e)
{
    PasswordDataSet.PASSWORDDataTable dt = new
    PasswordDataSet.PASSWORDDataTable();
    this.passwordTableAdapter1.Fill(dt);
    foreach (DataRow dr in dt.Rows)
    {
```

```
        if (dr[1].ToString() == txtusername.Text &&  
            dr[2].ToString() == txtpassword.Text)  
        {  
            txtusername.Text = "";  
            txtusername.Text = dr[1].ToString();  
            txtpassword.Text = "";  
            txtpassword.Text = dr[2].ToString();  
            MessageBox.Show("Έχετε συνδεθεί");  
            this.DialogResult = DialogResult.OK;  
        }  
        else  
        {  
            MessageBox.Show("Λάθος στοιχεία");  
        }  
    }  
}
```

8.2.2.3 Η μέθοδος button1_click()

Η συγκεκριμένη μέθοδος δεν κάνει τίποτα άλλο απ' το να αδειάζει το περιεχόμενο των **textBoxes**, επιλέγοντας το **button Καθαρισμός**

```
private void button1_Click(object sender, EventArgs e)  
{  
    txtusername.Text = "";  
    txtpassword.Text = "";  
}
```

8.2.3 Φόρμα EpiloghAsthenh

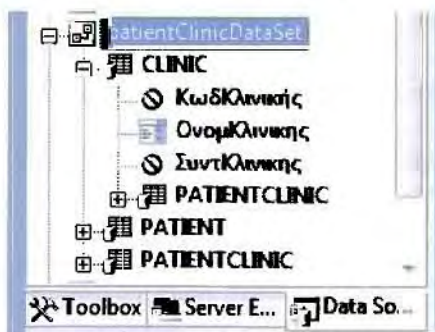
Η συγκεκριμένη φόρμα περιέχει ένα **comboBox** για την επιλογή της κλινικής (Εικ. 77), ένα **dataGrid** και δυο **textBoxes** με τα στοιχεία του ασθενή και ένα **button** για τη μετάβαση στην φόρμα με τις **ΛΕΙΤΟΥΡΓΙΕΣ**.



Εικόνα 77: Φόρμα για την επιλογή ασθενή

Για το σχεδιασμό της συγκεκριμένης φόρμας οι κλάσεις και τα αντικείμενα που χρησιμοποιήθηκαν, **DataSet** ή **TableAdapter**, έγιναν με κάποιο πιο αυτόματο τρόπο. Πρώτα από όλα θα πρέπει να δημιουργηθεί ένα **Dataset** το οποίο θα περιλαμβάνει τρεις πίνακες, τους **PATIENT**, **CLINIC** και τον **PATIENTCLINIC** και θα ονομαστεί **patientClinicDataset**.

Πηγαίνοντας στο **Data Sources** βλέπουμε ότι δημιουργήθηκε το **patientClinicDataset** με τους πίνακες που έχουν επιλεγεί (Εικ. 78). Πατώντας πάνω στον πίνακα **CLINIC** δίνονται τρεις επιλογές, **DataGrid**, **Details** ή **None** για το πώς θα φαίνεται ο πίνακας μέσα στην φόρμα, επιλέγεται το **Details**. Αυτό που θα πρέπει να εμφανίζεται στην φόρμα είναι η στήλη με το όνομα της Κλινικής μέσα σε ένα **comboBox**. Έπειτα με **Drag and Drop** μεταφέρεται στη φόρμα. Κάτω από την φόρμα αμέσως θα εμφανιστούν το **patientClinicDataSet**, το **CLINICTableAdapter** και το **CLINICBindingSource** (Εικ. 79) το οποίο αποτελεί το συνδεδετικό κρίκο με τη Βάση.

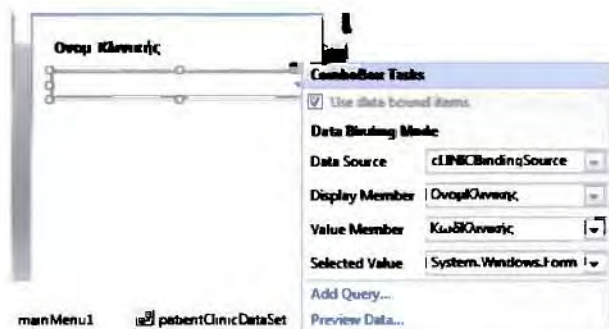


Εικόνα 78: Επιλογή τρόπου εμφάνισης αντικειμένων στη φόρμα



Εικόνα 79: Εμφάνιση αντικειμένων στη φόρμα

Πηγαίνοντας τώρα στο **comboBox** θα οριστούν ποια δεδομένα να εμφανίζει. Επιλέγεται το **cLINICBindingSource** που δημιουργήθηκε και ορίζεται να εμφανίζει τη στήλη **ΌνομαΚλινικής**(Εικ. 80).



Εικόνα 80: Εισαγωγή Bounds στο cbxClinic

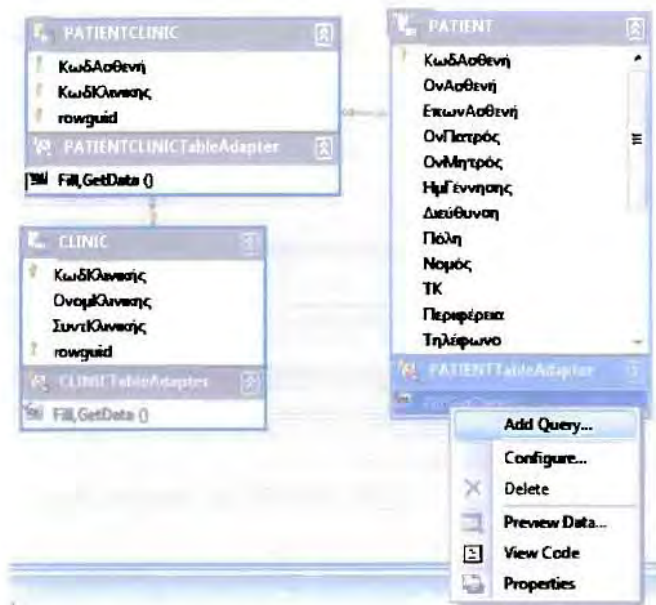
8.2.3.1 Η μέθοδος **EpiloghAsthenh_Load()**

Όταν μεταφέρθηκε το **comboBox** στην φόρμα, αυτόματα γράφτηκε η παρακάτω μέθοδος. Και έτσι αυτόματα παίρνει τα δεδομένα του πίνακα.

```
private void EpiloghAsthenh_Load(object sender, EventArgs e)
{
    if (patientClinicDataSetUtil.DesignerUtil.IsRunTime())
    {
        // TODO: Delete this line of code to remove the
        default AutoFill for 'patientClinicDataSet.CLINIC'.
        this.cLINICTableAdapter.Fill(this.patientClinicDataSet.CLINIC);
    }
}
```

8.2.3.2 Η μέθοδος **cbxClinic_SelectedIndexChanged()**

Το ίδιο γίνεται και με τον πίνακα **PATIENT** μόνο που θα επιλεγεί τα δεδομένα να εμφανίζονται σε **DataGrid**. Στον πρώτο έλεγχο γεμίζει τον πίνακα με όλα τα δεδομένα και για το δεύτερο θα δημιουργηθεί μια νέα μέθοδος η **FillByClinic()**. Πηγαίνοντας στο **DataSet** με δεξί κλικ-> **edit Dataset with Designer** θα εμφανιστεί η γραφική απεικόνισή του(Εικ. 81). Πηγαίνοντας στον πίνακα **PATIENT** επιλέγεται το **Add Query**.



Εικόνα 81: Γραφική απεικόνιση του patientClinicDataSet

```
private void cbxClinic_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        if (cbxClinic.Text == "--")
        {
            this.pATIENTTableAdapter.Fill(this.patientClinicDataSet.PATIENT);
        }
        else
        {
            this.pATIENTTableAdapter.FillByClinic(this.patientClinicDataSet.PATIENT,
            ((int)(System.Convert.ChangeType(cbxClinic.SelectedValue.ToString(),
            typeof(int), null))));
        }
    }
    catch (System.Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
}
```

Η SQL Query **FillByClinic** θα είναι η παρακάτω η οποία παίρνει ως παράμετρο τον κωδικό της Κλινικής. Επομένως όταν θα χρησιμοποιηθεί θα εμφανιστούν τα δεδομένα που έχουν ζητηθεί.

```
SELECT PATIENT.ΟνΑσθενή, PATIENT.ΕπωνΑσθενή, PATIENT.ΟνΜητρός,
PATIENT.ΗμΓέννησης, PATIENT.ΗμΕισαγωγής
FROM PATIENT INNER JOIN
PATIENTCLINIC ON PATIENT.ΚωδΑσθενή = PATIENTCLINIC.ΚωδΑσθενή
INNER JOIN CLINIC ON PATIENTCLINIC.ΚωδΚλινικής = CLINIC.ΚωδΚλινικής
WHERE (PATIENTCLINIC.ΚωδΚλινικής = @KodKlinikhs)
```

8.2.3.3 Η μέθοδος btnNextFrm_Click()

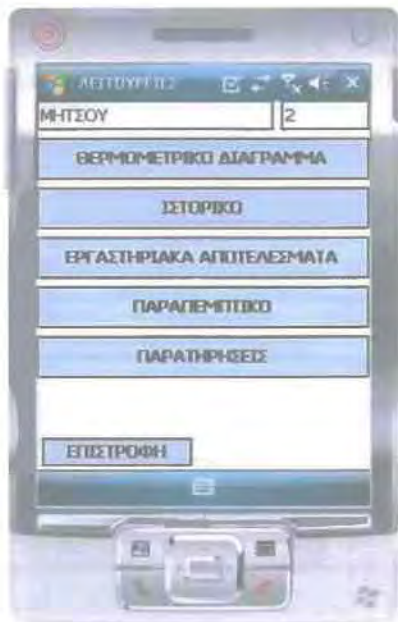
Η συγκεκριμένη μέθοδος θα καλέσει μια καινούργια φόρμα την **leitourgies** όπου όμως θα έχει δυο ορίσματα. Αυτό γίνεται για να μεταφερθούν το περιεχόμενο των **textBoxes** στην επόμενη φόρμα.

```
private void btnNextFrm_Click(object sender, EventArgs e)
{
    leitourgies frm = new
    leitourgies(txtPatient.Text,txtPatientId.Text);
    frm.Show();
}
```

8.2.4 Η φόρμα ΛΕΙΤΟΥΡΓΙΕΣ

Η φόρμα **leitourgies** ουσιαστικά είναι μια μεταβατική φόρμα για να μπορεί ο χρήστης να μεταβεί στην ανάλογη λειτουργία. Περιέχει δυο **textBoxes**, πέντε **buttons** για τη μεταφορά του χρήστη σε διαφορετική λειτουργία κάθε φορά και ένα **button** για τη μεταφορά του στην προηγούμενη φόρμα (Εικ. 82).

Είναι δυνατόν επίσης, να αλλάξει η εμφάνιση του ονόματος στη φόρμα από της ιδιότητες της φόρμας.



Εικόνα 82: Φόρμα ΛΕΙΤΟΥΡΓΙΕΣ

8.2.4.1 Η μέθοδος leitourgies()

Η μέθοδος της φόρμας θα έχει δυο ορίσματα τύπου **string**. Αυτό γίνεται για να μεταφερθούν τα δεδομένα από την προηγούμενη φόρμα και θα μπουν στα δυο αντίστοιχα **textBoxes**. Για τα **buttons** ακολουθείται η ίδια διαδικασία καλώντας κάθε φορά την ανάλογη φόρμα.

```
public leitourgies(string strTxtPatient, string strTxtPatientId)
{
    InitializeComponent();

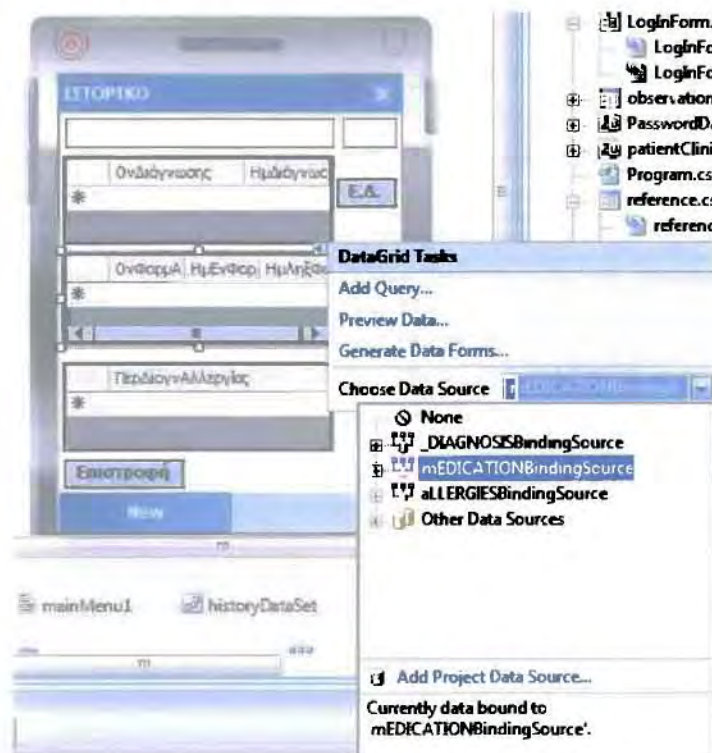
    txtPatient.Text = strTxtPatient;
    txtPatientId.Text = strTxtPatientId;
}
```

Η φόρμα περιλαμβάνει επίσης τις κλάσεις μέσω των οποίων μεταφερόμαστε στις ανάλογες φόρμες:

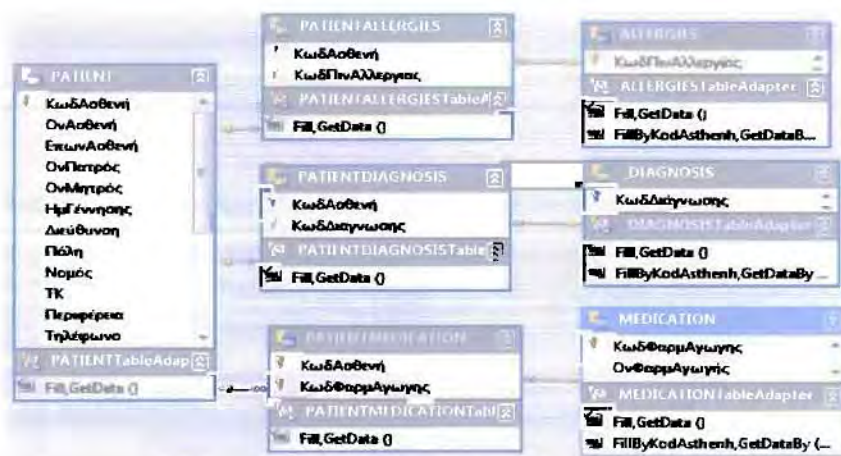
```
private void btnThDiag_Click(object sender, EventArgs e) {}
private void btnHist_Click(object sender, EventArgs e) {}
private void btnLabRes_Click(object sender, EventArgs e) {}
private void btnRef_Click(object sender, EventArgs e) {}
private void btnObser_Click(object sender, EventArgs e) {}
```

8.2.5 Η Φόρμα ΙΣΤΟΡΙΚΟ

Για το σχεδιασμό της συγκεκριμένης φόρμας(Εικ. 83) δημιουργήθηκε ένα νέο **Dataset** το **HistoryDataset** (Εικ. 84) που περιλαμβάνει τους εξής πίνακες: **PATIENTMEDICATION**, **ALLERGIES**, **PATIENT**, **PATIENTDIAGNOSIS**, **PATIENTALLERGIES**, **DIAGNOSIS**, και **MEDICATION**. Όπως και πριν μεταφέρονται στη φόρμα τα **DataGrids** και θα επιλεγεί ποιο **BindingSource** θα χρησιμοποιηθεί κάθε φορά. Πατώντας το ανάλογο **button** κάθε φορά ο πίνακας θα γεμίζει με τα δεδομένα του ασθενή. Αυτή τη φορά θα χρησιμοποιηθεί η μέθοδος **FillByKodAsthenh()** η οποία θα προστεθεί και στους τρεις βασικούς πίνακες .



Εικόνα 83: Φόρμα ΙΣΤΟΡΙΚΟ



Εικόνα 84: Γραφική απεικόνιση historyDataSet

Η SQL Query **FillByKodAsthenh** θα είναι η παρακάτω η οποία παίρνει ως παράμετρο τον ΚωδΑσθενή ο οποίος βρίσκεται στο **textbox txtPatientId**. Επομένως όταν χρησιμοποιηθεί θα εμφανιστούν τα δεδομένα που ζητήθηκαν.

```
SELECT MEDICATION.ΚωδΦαρμαγωγής, MEDICATION.ΟνΦαρμαγωγής,
MEDICATION.ΗμΕνΦαρμαγωγής, MEDICATION.ΗμΛηξΦαρμαγωγής,
MEDICATION.Δοσολογία
FROM MEDICATION INNER JOIN PATIENTMEDICATION ON
MEDICATION.ΚωδΦαρμαγωγής=PATIENTMEDICATION.ΚωδΦαρμαγωγής
INNER JOIN PATIENT ON PATIENTMEDICATION.ΚωδΑσθενή=
PATIENT.ΚωδΑσθενή
WHERE (PATIENTMEDICATION.ΚωδΑσθενή = @KodAsthenh)
```

Μέθοδοι οι οποίες δημιουργήθηκαν και οι οποίες ενεργοποιούνται με το πάτημα του ανάλογου **button** με τις οποίες φορτώνουμε τα δεδομένα στα DataGrids της φόρμας:

```
private void btnConfirmation_Click(object sender, EventArgs e){}
private void btnConfMed_Click(object sender, EventArgs e){}
private void btnConfAller_Click(object sender, EventArgs e){}
private void btnReturn_Click(object sender, EventArgs e){}
```

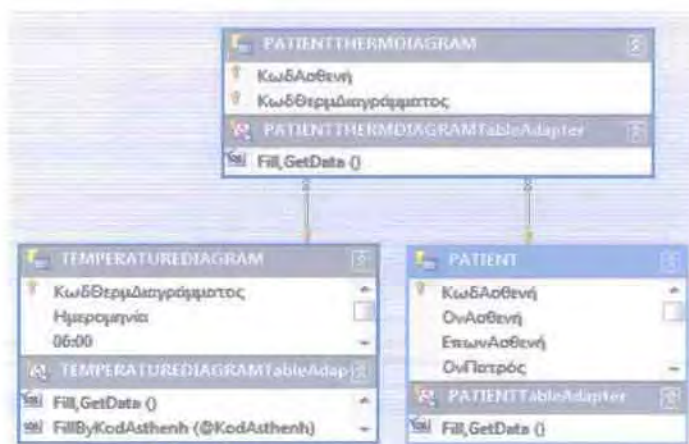
8.2.6 Η φόρμα ΘΕΡΜΟΜΕΤΡΙΚΟ ΔΙΑΓΡΑΜΜΑ

Για τη συγκεκριμένη φόρμα (Εικ. 85) θα δημιουργηθεί ένα νέο **DataSet**, το **TemperatureDataset** (Εικ. 86) το οποίο θα περιέχει τους πίνακες **PATIENT**, τον **PATIENTTHERMDIAGRAM** και τον **TEMPERATUREDIAGRAM**.

Επιλέγοντας τη φόρμα πρώτον θα ενημερώνεται το ιατρικό προσωπικό για τις τιμές της θερμοκρασίας του ασθενή και δεύτερο θα μπορεί ο Νοσηλευτής να ανανεώνει τις τιμές του Θερμομετρικού Διαγράμματος. Γι' αυτό θα χρησιμοποιηθεί η **Update** μέθοδος.



Εικόνα 85: Φόρμα ΘΕΡΜΟΜΕΤΡΙΚΟ ΔΙΑΓΡΑΜΜΑ



Εικόνα 86: Γραφική απεικόνιση temperatureDataSet

8.2.6.1 Η μέθοδος btnUpdate_Click()

Στην μέθοδο **btnUpdate_Click()** χρησιμοποιείται η SQL μέθοδος **Update()** για να ανανεωθούν τα δεδομένα στον πίνακα **TEMPERATUREDIAGRAM**. Η μέθοδος **Update** περιλαμβάνεται στις ιδιότητες του **Dataset** και μπορεί να χρησιμοποιηθεί όπως είναι ή να δημιουργηθεί μια νέα ανάλογα με τις ανάγκες κάθε φορά.

Στον κώδικα που ακολουθεί παρατηρούμε ότι προστίθεται το **SqlConnection()**. Ο **SqlConnection (String)** [24] είναι μια μέθοδος η οποία αρχικοποιεί ένα νέο **instance** της κλάσης **SqlConnection** με το συγκεκριμένο **string** που έχει οριστεί. Μέχρι στιγμής δεν το είχε προστεθεί σε καμία φόρμα, αλλά η εφαρμογή το προσέθετε μόνο του. Είναι μια κλάση η οποία συμπεριλαμβάνεται στη βιβλιοθήκη **System.Data.SqlServerCe**. Όταν δημιουργήθηκαν τα **datasets** ορίστηκαν να συνδέονται με τη Βάση μέσω του συγκεκριμένου **string**, το οποίο δίνει εντολή σε ποια Βάση θα συνδεθεί η εφαρμογή και ποιο μονοπάτι θα

ακολουθήσει.

Ο `SqlCommand(string, SqlConnection)` [25] είναι μια μέθοδος η οποία αρχικοποιεί ένα `instance` στην κλάση `SqlCommand` μέσω ενός `query` και ένα `SqlConnection string`.

Η `insertCmd.ExecuteNonQuery()` είναι η μέθοδος για να εκτελεστεί ο `SqlCommand insertCmd` [26].

```
private void btnUpdate_Click(object sender, EventArgs e)
{
    try
    {
        SqlConnection myCon = new SqlConnection("Data
Source =\\program files\\emedrecordmbl\\hospital.sdf");
        myCon.Open();

        SqlCommand updateCmd = new SqlCommand("UPDATE
TEMPERATUREDIAGRAM SET [06:00]='" + txt06.Text + "', [09:00] = '" +
txt09.Text + "', [12:00]='" + txt12.Text + "', [15:00]='" +
txt15.Text + "', [18:00]='" + txt18.Text + "', [21:00]='" +
txt21.Text + "', [00:00]='" + txt00.Text + "', [03:00] = '" + txt03.Text
+ " WHERE [ΚωδθερμΔιαγράμματος]=@KodThermDiagram", myCon);
        SqlParameter param = new SqlParameter();
        param.ParameterName = "@KodThermDiagram";
        param.Value = κωδθερμΔιαγράμματοςTextBox;
        updateCmd.Parameters.Add(param);

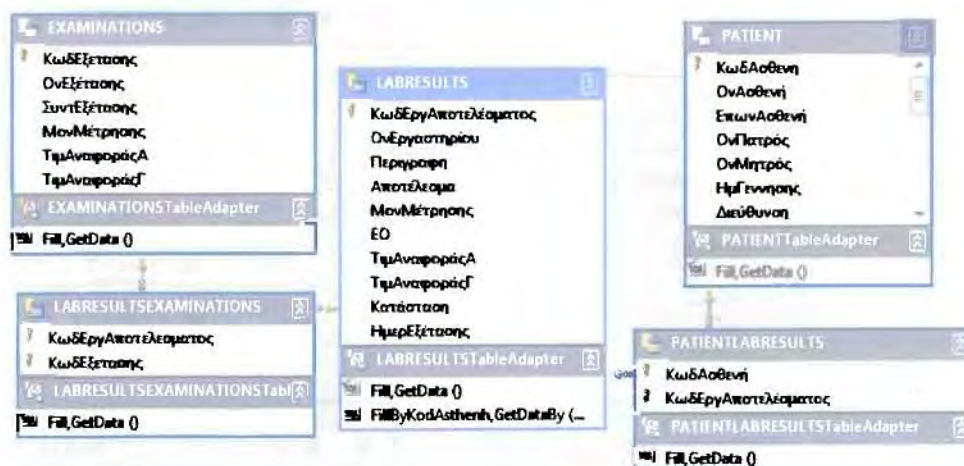
        updateCmd.ExecuteNonQuery();
        myCon.Close();
    }
    catch
    {
        MessageBox.Show("Λάθος ενημέρωσης");
    }
}
```

8.2.7 Η φόρμα LabResults

Το ζητούμενο σε αυτή τη φόρμα (Εικ. 87) είναι να μπορεί ο ιατρός να ενημερωθεί για τα εργαστηριακά αποτελέσματα του ασθενή. Περιέχει δυο `textboxes` για τα στοιχεία του ασθενή, ένα `dataGrid` για Εργαστηριακά Αποτελέσματα και δυο `buttons`. Για τη φόρμα αυτή δημιουργήθηκε ένα νέο `DataSet` το `LabResultsDataSet` (Εικ. 88). Στον `TableAdapter` του πίνακα `LABRESULTS` προσθέσαμε ένα νέο `Query`, το `FillByKodAsthenh()` με παράμετρο τον κωδικό του ασθενή.



Εικόνα 87: Φόρμα ΕΡΓΑΣΤΗΡΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ



Εικόνα 88: Γραφική απεικόνιση LabResultsDataSet

```

SELECT LABRESULTS.ΟνΕργαστηρίου, LABRESULTS.Περιγραφή,
       LABRESULTS.Αποτέλεσμα, LABRESULTS.ΜονΜέτρησης,
       LABRESULTS.ΤιμΑναφοράςΑ, LABRESULTS.ΤιμΑναφοράςΓ',
       LABRESULTS.ΗμερΕξέτασης
FROM LABRESULTS INNER JOIN PATIENTLABRESULTS ON
       LABRESULTS.ΚωδΕργΑποτελέσματος =
       PATIENTLABRESULTS.ΚωδΕργΑποτελέσματος
INNER JOIN PATIENT ON PATIENTLABRESULTS.ΚωδΑσθενή =
       PATIENT.ΚωδΑσθενή
WHERE (PATIENTLABRESULTS.ΚωδΑσθενή = @ΚοδΑσθενη)
    
```

8.2.7.1 Η μέθοδος btnConfirmation_click()

Με την μέθοδο `btnConfirmation_click()` η οποία ενεργοποιείται με το πάτημα του `button` Επιβεβαίωση γεμίζει το `DataGrid` με τα δεδομένα του συγκεκριμένου ασθενή. Οι συγκεκριμένες εντολές θα μπορούσαν να ενσωματωθούν στο `Load Form`, όταν φορτώνεται η φόρμα δηλαδή στη κλάση `public LabResults()`.

```
private void btnConfirmation_Click(object sender, EventArgs e)
{
    try
    {
        this.LABRESULTSTableAdapter.FillByKodAsthenh(this.labResultsDataSet.L
        ABRESULTS,
        ((int)(System.Convert.ChangeType(txtPatientId.Text.ToString(),
        typeof(int), null)));
    }
    catch (System.Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
    }
}
```

8.2.8 Η φόρμα `observation`

Η φόρμα `observation` (Εικ. 89) χρησιμοποιεί το `temperatureDataSet` για τη σύνδεση με τη βάση και ανανεώνει τα δεδομένα στη στήλη Παρατηρήσεις, μέσω της μεθόδου `Update` του `DataTable`. Έτσι οι Νοσηλευτές μπορούν να προσθέτουν τις παρατηρήσεις τους ώστε με τη σειρά τους να τα δουν οι Γιατροί. Για να αλλάξει το μέγεθος της στήλης όπως φαίνεται στην εικόνα, από τα `properties` του αντικειμένου επιλέγεται η επιλογή `TableStyles` και έτσι οποιαδήποτε αλλαγή.



Εικόνα 89: Φόρμα ΠΑΡΑΤΗΡΗΣΕΙΣ

8.2.9 Η φόρμα ΠΑΡΑΠΕΜΠΤΙΚΟ

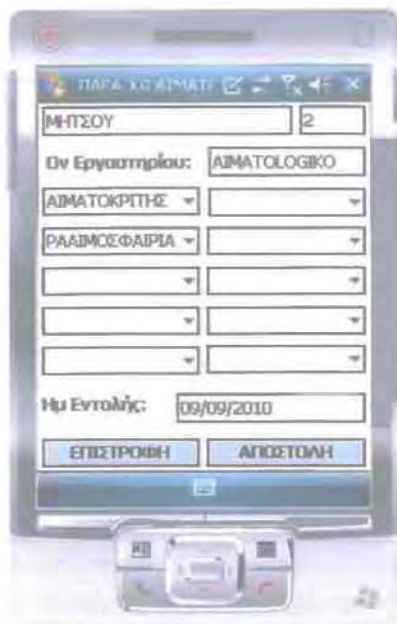
Η φόρμα **Reference** (Εικ. 90) δίνει τη δυνατότητα στον χρήστη να επιλέξει σε ποια εργαστήριο θα στείλει παραπεμπτικό, και με απλές εντολές οι οποίες έχουν αναφερθεί και πριν, μεταφέρεται στην ανάλογη φόρμα.



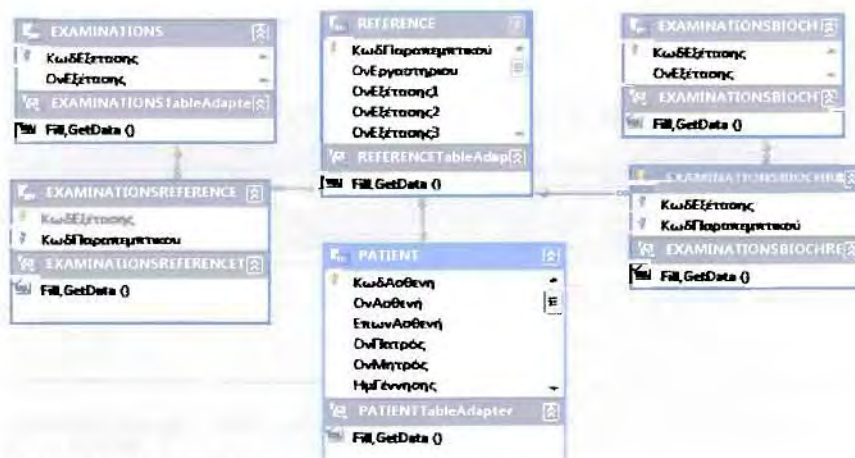
Εικόνα 90: Φόρμα ΠΑΡΑΠΕΜΠΤΙΚΟ

8.2.10 Η φόρμα ΠΑΡΑΠΕΜΠΤΙΚΟ ΑΙΜΑΤΟΛΟΓΙΚΟΥ

Στη φόρμα **referenceHematologic** (Εικ. 91) ο Γιατρός μπορεί να στείλει άμεσα εντολή για διάφορες εξετάσεις οι οποίες αφορούν το Αιματολογικό. Για το σχεδιασμό θα δημιουργηθεί ένα νέο dataset το **referenceDataSet** (Εικ. 92) το οποίο θα περιέχει τους εξής πίνακες: **EXAMINATIONSBIOC**, **EXAMINATIONSREFERENCE**, **REFERENCE**, **PATIENT**, **EXAMINATIONSBIOCHREFERENCE** και τον πίνακα **EXAMINATIONS**.



Εικόνα 91: Φόρμα ΠΑΡ/ΚΟ ΑΙΜΑΤΟΛΟΓΙΚΟ



Εικόνα 92: Γραφική απεικόνιση referenceDataSet

8.2.10.1 Αυτόματη χρήση ορίων

Σε κάθε **comboBox** της φόρμας θα πρέπει να μπορούν να εμφανίζονται οι εξετάσεις του Αιματολογικού Εργαστηρίου, ώστε να μπορεί ο Ιατρός να επιλέξει. Θα μπορούσαμε στη θέση των **comboBoxes** να χρησιμοποιήσουμε **RadioButtons**, αλλά επειδή σε μια τέτοια συσκευή πρέπει να υπολογίσουμε και το θέμα του χώρου επιλέχθηκε να σχεδιαστεί με αυτόν τον τρόπο. Για να μπορέσουμε να γίνει αυτό εφόσον έχει γίνει το **drug and drop** τον πίνακα **REFERENCE**, σε κάθε ένα **comboBox** επιλέγεται το ανάλογο **BindingSource**. Θα οριστεί να παίρνει τιμές από τον πίνακα **EXAMINATIONS** τη στήλη **ΟνΕξέτασης** (Εικ. 93).



Εικόνα 93: Εισαγωγή bounds σε comboBox εξετάσεων

8.2.10.2 Η μέθοδος btnConfirmation_Click()

Αυτή τη φορά το SQL query θα το υλοποιηθεί μέσα στον κώδικα και όχι στο DataSet όπως έγινε τις προηγούμενες φορές. Ο τύπος του SQL query είναι το INSERT και οι τιμές θα περάσουν με παράμετρο, όπου ορίζεται να παίρνουν τις τιμές από τα ανάλογα αντικείμενα στη φόρμα. Σε προηγούμενα παραδείγματα δεν χρειαζόταν να γίνει αυτό γιατί χρησιμοποιούνταν τα αντικείμενα, πχ textboxes, Datagrids, κατευθείαν από το DataSet και το οποίο αναγνώριζε από πού θα πάρουν τις τιμές οι παράμετροι που ορίζονται μέσα στα SQL queries.

```
private void btnConfirmation_Click(object sender, EventArgs e)
{
    SqlConnection myCon = new SqlConnection("Data Source
        = \\program files\\emedrecorndb1\\hospital.sdf");
    myCon.Open();

    SqlCommand insertCmd = new SqlCommand("INSERT INTO
    REFERENCE (Ονεργαστηρίου, Ονεξέτασης1, Ονεξέτασης2, Ονεξέτασης3, Ονεξέτασης
    4, Ονεξέτασης5, Ονεξέτασης6, Ονεξέτασης7, Ονεξέτασης8, Ονεξέτασης9, Ονεξέ
    τησης10, ΚωδΑσθενή) VALUES
    (@Ονεργαστηρίου, @Ονεξέτασης1, @Ονεξέτασης2, @Ονεξέτασης3, @Ονεξέτασης4,
    @Ονεξέτασης5, @Ονεξέτασης6, @Ονεξέτασης7, @Ονεξέτασης8, @Ονεξέτασης9, @Ονε
    ξέτασης10, @ΚωδΑσθενή)", myCon);
    try
    {
        insertCmd.Parameters.AddWithValue("@Ονεργαστηρίου",
            txtOnErgasthriou.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέτασης1", cbxOnExet1.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέτασης2", cbxOnExet2.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέτασης3", cbxOnExet3.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέτασης4", cbxOnExet4.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέτασης5", cbxOnExet5.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέτασης6", cbxOnExet6.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέτασης7", cbxOnExet7.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέτασης8", cbxOnExet8.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέτασης9", cbxOnExet9.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέτασης10", cbxOnExet10.Text);
        insertCmd.Parameters.AddWithValue("@ΚωδΑσθενή", txtPatientId.Text);
        Console.WriteLine(txtPatientId.Text);
        insertCmd.ExecuteNonQuery();
    }
}
```

```
    MessageBox.Show("Η εισαγωγή ολοκληρώθηκε");  
  }  
  catch  
  {  
    MessageBox.Show("Λάθος εισαγωγής");  
  }  
  myCon.Close();  
  
  txtOnErgasthriou.Text = "";  
  cbxOnExet1.Text = null;  
  cbxOnExet2.Text = null;  
  cbxOnExet3.Text = null;  
  cbxOnExet4.Text = null;  
  cbxOnExet5.Text = null;  
  cbxOnExet6.Text = null;  
  cbxOnExet7.Text = null;  
  cbxOnExet8.Text = null;  
  cbxOnExet9.Text = null;  
  cbxOnExet10.Text = null;  
  
}
```

8.2.11 Η φόρμα ΠΑΡΑΠΕΜΠΤΙΚΟ ΒΙΟΧΗΜΙΚΟΥ/ΜΙΚΡΟΒΙΟΛΟΓΙΚΟΥ

Η τελευταία φόρμα περιλαμβάνει το παραπεμπτικό για το Βιοχημικό/Μικροβιολογικό (Εικ. 94). Αυτή τη φορά θα πρέπει να εμφανίζονται οι ανάλογες εξετάσεις, όπου αυτό το ορίζεται όπως ειπώθηκε και προηγουμένως σε κάθε comboBox και παίρνει τιμές από τη στήλη ΟνΕξέτασης του πίνακα EXAMINATIONS BIOCH.



Εικόνα 94: Φόρμα ΠΑΡ/ΚΟ ΜΙΚΡΟΒ/ΒΙΟΧ

8.2.11.1 Η μέθοδος btnConfirmation_Click

Επαναλαμβάνεται ο προηγούμενος κώδικας με πριν με τη διαφορά ότι στα comboBoxes

είχε οριστεί να παίρνει τα δεδομένα από τον πίνακα EXAMINATIONSBIOCH, ορίζεται δηλαδή ανάλογα το BindingSource.

```
private void btnConfirmation_Click(object sender, EventArgs e)
{
    SqlConnection myCon = new SqlConnection("Data Source=
        \\program files\\emedrecorndb1\\hospital.sdf");
    myCon.Open();
    SqlCommand insertCmd = new SqlCommand("INSERT INTO
REFERENCE (Ονεργαστηρίου, Ονεξέταση1, Ονεξέταση2, Ονεξέταση3, Ονεξέταση
4, Ονεξέταση5, Ονεξέταση6, Ονεξέταση7, Ονεξέταση8, Ονεξέταση9, Ονεξέτ
ασης10, ΚωδΑσθενή) VALUES
(@Ονεργαστηρίου, @Ονεξέταση1, @Ονεξέταση2, @Ονεξέταση3, @Ονεξέταση4, @
Ονεξέταση5, @Ονεξέταση6, @Ονεξέταση7, @Ονεξέταση8, @Ονεξέταση9, @Ονεξ
έταση10, @ΚωδΑσθενή) ", myCon);

    Try
    {
        insertCmd.Parameters.AddWithValue("@Ονεργαστηρίου",
            txtOnErgasthriou.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέταση1", cbxOnExet1.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέταση2", cbxOnExet2.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέταση3", cbxOnExet3.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέταση4", cbxOnExet4.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέταση5", cbxOnExet5.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέταση6", cbxOnExet6.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέταση7", cbxOnExet7.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέταση8", cbxOnExet8.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέταση9", cbxOnExet9.Text);
        insertCmd.Parameters.AddWithValue("@Ονεξέταση10",
            cbxOnExet10.Text);
        insertCmd.Parameters.AddWithValue("@ΚωδΑσθενή", txtPatientId.Text);
        Console.WriteLine(txtPatientId.Text);
        insertCmd.ExecuteNonQuery();
        MessageBox.Show("Η εισαγωγή ολοκληρώθηκε");
    }
    catch
    {
        MessageBox.Show("Λάθος εισαγωγής");
    }

    myCon.Close();

    txtOnErgasthriou.Text = "";
    cbxOnExet1.Text = null;
    cbxOnExet2.Text = null;
    cbxOnExet3.Text = null;
    cbxOnExet4.Text = null;
    cbxOnExet5.Text = null;
    cbxOnExet6.Text = null;
    cbxOnExet7.Text = null;
    cbxOnExet8.Text = null;
    cbxOnExet9.Text = null;
    cbxOnExet10.Text = null;
}
```

**«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»**

Ιωάννα Ζορμπά

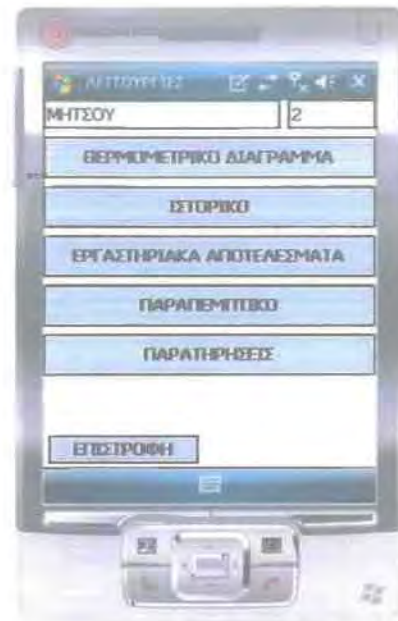
9 Παρουσίαση της τελικής Εφαρμογής και των λειτουργιών της

Ολοκληρώνοντας το σχεδιασμό και τη δημιουργία της εφαρμογής παρατίθεται ένας συνοπτικός Οδηγός Χρήσης για τους επικείμενους χρήστες της εφαρμογής με τα βήματα που θα πρέπει να ακολουθήσουν ώστε να λειτουργήσει σωστά.

Κατά την εισαγωγή στην εφαρμογή θα πρέπει να συνδεθείτε με την εφαρμογή για να μπορείτε να επιτελέσετε οποιαδήποτε λειτουργία. Επιλέγοντας το πλήκτρο **Χρήστης** -> **Σύνδεση** θα μεταφερθείτε στην φόρμα LogInForm όπου εκεί θα πρέπει να δώσετε τα στοιχεία σας, και να επιλέξετε το πλήκτρο **Σύνδεση**.



Στη Φόρμα για την επιλογή θα πρέπει να επιλέξετε σε ποια Κλινική θέλετε να συνδεθείτε για να βρείτε τον ασθενή που θέλετε. Επιλέγοντας τον ασθενή πατάτε το πλήκτρο **Μετάβαση στις ΛΕΙΤΟΥΡΓΙΕΣ** θα μεταφερθείτε στη φόρμα με τις Λειτουργίες.



Επιλέγοντας το ΙΣΤΟΡΙΚΟ θα μεταβείτε στη Φόρμα με το Ιστορικό του ασθενούς. Εκεί μπορείτε είτε να ενημερωθείτε για τις διαγνώσεις, είτε για τη φαρμακευτική αγωγή που ακολουθεί, αλλά ακόμα αν σας ενδιέφερε να ενημερωθείτε για τυχόν αλλεργίες που ίσως να έχει ο συγκεκριμένος ασθενής.

Στη φόρμα ΘΕΡΜΟΜΕΤΡΙΚΟ ΔΙΑΓΡΑΜΜΑ ο Γιατρός μπορεί να ενημερωθεί για τις μετρήσεις τις ημέρας συνολικά, και ο Νοσηλευτής μπορεί να ενημερώσει τα δεδομένα με τις νέες μετρήσεις.

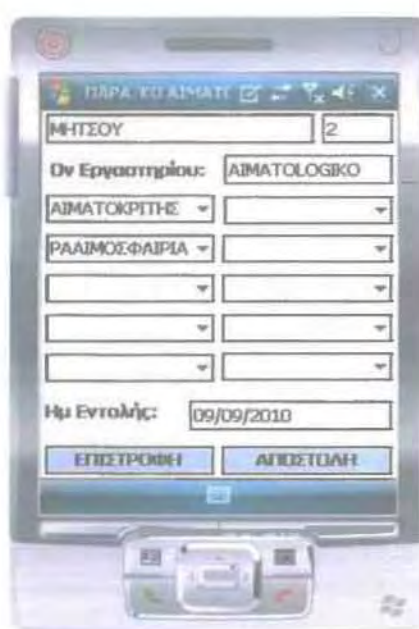


Στη φόρμα ΕΡΓΑΣΤΗΡΙΑΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ο Ιατρός μπορεί να ενημερωθεί άμεσα για τα εργαστηριακά αποτελέσματα του ασθενή.

Στη Φόρμα ΠΑΡΑΤΗΡΗΣΕΙΣ ο Γιατρός μπορεί να ενημερωθεί για τυχόν λεπτομέρειες που έχει εντοπίσει ο Νοσηλευτής, όπως πως εξελίσσεται η φαρμακευτική αγωγή του ασθενούς.



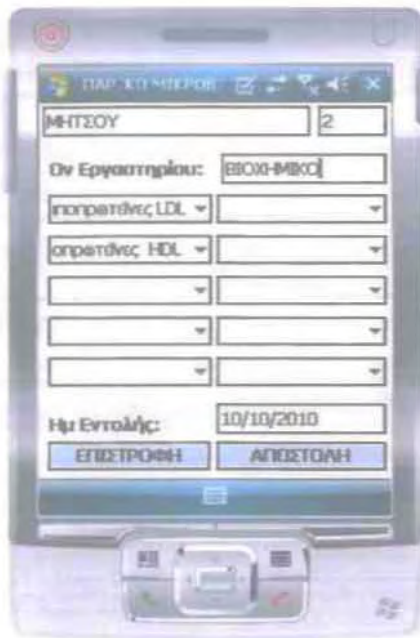
Στη φόρμα ΠΑΡΑΠΕΜΠΤΙΚΟΥ ο Γιατρός μπορεί να επιλέξει μέσα από μια λίστα εξετάσεων του ΑΙΜΑΤΟΛΟΓΙΚΟΥ ώστε να αποσταλεί το παραπεμπτικό άμεσα στο Εργαστήριο.



Το ίδιο μπορεί να γίνει και με τη φόρμα για το ΠΑΡΑΠΕΜΠΤΙΚΟ ΜΙΚΡ/ΚΟΥ-ΒΙΟΧΗΜΙΚΟΥ.

**«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»**

Ιωάννα Ζορμπά



10. Βιβλιογραφία

- [1] Ιωάννης Μαντάς, Σημειώσεις μαθήματος Πληροφοριακά συστήματα υγείας, 2007
- [2] Ιωάννης Αποστολάκης, Πληροφοριακά Συστήματα Υγείας, Παπαζήση 2007
- [3] Πληροφορίες για Merge Replication
<http://etutorials.org/Programming/building+solutions+with+the+microsoft+net+compact+framework/Part+II+Essential+Architectural+Concepts/Chapter+7.+Data+Synchronization/RDA/>
- [4] Ευάγγελος Σακκόπουλος, σημειώσεις μαθήματος Τεχνολογίες Εφαρμογών Διαδικτύου, 2007
- [5] Web Synchronization for Merge Replication
<http://msdn.microsoft.com/en-us/library/ms151763.aspx>
- [6] Merge Replication Step by Step Procedure
<http://www.codeproject.com/KB/database/sql2005-replication.aspx>
- [7] Microsoft Visual Studio 2008 Service Pack 1
<http://www.microsoft.com/download/en/details.aspx?id=10986>
- [8] System.Data.SqlServerCe Namespace
<http://msdn.microsoft.com/en-us/library/system.data.sqlserverce%28v=VS.90%29.aspx>
- [9] Windows Mobile 6 Professional and Standard Software Development Kits Refresh
<http://www.microsoft.com/download/en/details.aspx?id=6135>
- [10] Microsoft SQL Server Compact 3.5 Service Pack 2 Server Tools
<http://www.microsoft.com/download/en/details.aspx?id=9290>
- [11] Microsoft Windows Mobile Device Center 6.1 for Windows Vista
<http://www.microsoft.com/download/en/details.aspx?id=14>
- [12] System.Data.SqlClient Namespace
<http://msdn.microsoft.com/en-us/library/system.data.sqlclient%28v=VS.90%29.aspx>
- [13] John Mantas - Arie Hasman, Πληροφορική της Υγείας Νοσηλευτική Προσέγγιση, Π.Χ. Πασχαλίδης, 2007
- [14] Υπουργείο Υγείας -ICD10
<http://www.vvka.gov.gr/articles/health/domes-kai-draseis-gia-thn-ygeia/kwdikopoihseis/86-statistikh-taksinomhsh-noswn-kai-synafwn-problhmatwn-ygeias>
- [15] ΙΦΕΤ - Κατάλογος Φαρμάκων
<http://www.ifet.gr/drugs/catalogs.htm>
- [16] Ιατρικός Σύλλογος Αθηνών- Λίστα Ειδικοτήτων
<http://www.isathens.gr/grindex.html>
- [17] Ιατρική σχολή Αθηνών -Εξετάσεις
- Πανεπιστήμιο Στερεάς Ελλάδας - Τμήμα Πληροφορική με Εφαρμογές στη Βιοϊατρική

<http://med-ath-2000.tripod.com/fisiologikestimes.htm>

- [18] Για τη δημιουργία Βάσης Δεδομένων μέσω SSMS
<http://msdn.microsoft.com/en-us/library/ms186312.aspx>
- [19] R. Elmasri – S.B. Navathe, Θεμελιώδεις αρχές συστημάτων Βάσεων Δεδομένων, Α' & Β' ΤΟΜΟΙ (4^η έκδοση Αναθεωρημένη). ΔΙΑΥΛΟΣ 2005
- [20] Π.Ο.Υ. Κωδικοποίηση ATC
http://www.whocc.no/atc/structure_and_principles/
- [21] Creating, Modifying, and Deleting Publications and Articles (Replication)
[http://msdn.microsoft.com/en-us/library/ms152542\(v=SOL.100\).aspx](http://msdn.microsoft.com/en-us/library/ms152542(v=SOL.100).aspx)
- [22] Synchronizing Subscriptions (Replication)
[http://msdn.microsoft.com/en-us/library/ms152550\(v=SOL.100\).aspx](http://msdn.microsoft.com/en-us/library/ms152550(v=SOL.100).aspx)
- [23] Creating, Modifying, and Deleting Subscriptions (Replication)
[http://msdn.microsoft.com/en-us/library/ms152503\(v=SOL.100\).aspx](http://msdn.microsoft.com/en-us/library/ms152503(v=SOL.100).aspx)
- [24] SqlConnection Constructor
<http://msdn.microsoft.com/en-US/library/system.data.sqlserverce.sqlconnection.sqlconnection%28v=VS.90%29.aspx>
- [25] SqlCommand Constructor
<http://msdn.microsoft.com/en-us/library/system.data.sqlserverce.sqlcommand.sqlcommand%28v=VS.90%29.aspx>
- [26] SqlCommand.ExecuteNonQuery Method
<http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand.executenonquery.aspx>

ΠΑΡΑΡΤΗΜΑ

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΙΑΤΡΩΝ

ΚΛΙΝΙΚΗ:

ΕΙΔΙΚΟΤΗΤΑ:

ΗΛΙΚΙΑ: 30-40 40-50 50-60

1) Χρησιμοποιείται Η/Υ στο χώρο εργασίας σας;

ΝΑΙ

ΟΧΙ

2) Η χρήση Η/Υ στην εργασία σας θα μπορούσατε να πείτε ότι:

Είναι πολύ χρήσιμη αλλά χρονοβόρα διαδικασία

Είναι πολύ χρήσιμη χωρίς να απαιτείται πολύς χρόνος

Είναι πολύ χρήσιμη αλλά είναι δύσκολο το περιβάλλον του προγράμματος

Είναι χρήσιμη και το περιβάλλον του προγράμματος είναι εύκολο

Είναι απλά ένα χρήσιμο εργαλείο

Δεν χρησιμεύει ιδιαίτερα

Δεν χρησιμεύει καθόλου

3) Ποιες λειτουργίες χρησιμοποιείτε στον Η/Υ (πολλαπλή επιλογή)

Ανάγνωση Ιστορικού ασθενούς

Ενημέρωση Ιστορικού ασθενούς

Ανάγνωση Εργαστηριακών Αποτελεσμάτων

Ενημέρωση Εργαστηριακών Αποτελεσμάτων

Δημιουργία παραπεμπτικού

Ανάγνωση μετρήσεων Ζωτικών Σημείων

Συμπλήρωση Φαρμακευτικής Αγωγής

Πορεία υγείας ασθενούς

Οδηγίες προς Νοσηλευτές

Άλλο

4) Χρησιμοποιείτε κινητή συσκευή για λειτουργίες πέραν των βασικών πχ μηνύματα, τηλεφωνήματα;

ΝΑΙ

ΟΧΙ

5) Αν ΝΑΙ ποιες είναι αυτές (πολλαπλή επιλογή)

Internet

Ατζέντα

Σημειώσεις

Υπενθυμίσεις

Άλλο

6) Χρησιμοποιείτε κινητή συσκευή ως εργαλείο κατά την επίσκεψης στους θαλάμους;

ΝΑΙ

OXI

- 7) Αν ΝΑΙ ποιες είναι αυτές
- | | |
|--------------------------------------|--------------------------|
| Ανάγνωση Ιστορικού Ασθενούς | <input type="checkbox"/> |
| Ανάγνωση Εργαστηριακών Αποτελεσμάτων | <input type="checkbox"/> |
| Πορεία υγείας ασθενούς | <input type="checkbox"/> |
| Ανάγνωση μετρήσεων ζωτικών σημείων | <input type="checkbox"/> |
| Αποστολή παραπεμπτικού | <input type="checkbox"/> |
| Άλλο..... | |
| | |
| | |
- 8) Αν όχι ποιες λειτουργίες πιστεύετε ότι θα σας εξυπηρετούσαν
- | | |
|--------------------------------------|--------------------------|
| Ανάγνωση Ιστορικού Ασθενούς | <input type="checkbox"/> |
| Ανάγνωση Εργαστηριακών Αποτελεσμάτων | <input type="checkbox"/> |
| Πορεία υγείας ασθενούς | <input type="checkbox"/> |
| Ανάγνωση μετρήσεων ζωτικών σημείων | <input type="checkbox"/> |
| Αποστολή παραπεμπτικού | <input type="checkbox"/> |
| Άλλο..... | |
| | |
| | |
- 9) Δεν πιστεύω ότι χρειάζεται η χρήση PDA κατά την επίσκεψη στους θαλάμους

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΝΟΣΗΛΕΥΤΩΝ

ΚΛΙΝΙΚΗ:.....

ΕΙΔΙΚΟΤΗΤΑ:.....

ΗΛΙΚΙΑ: 30-40 40-50 50-60

1) Χρησιμοποιείται Η/Υ στο χώρο εργασίας σας;

ΝΑΙ

ΟΧΙ

2) Η χρήση Η/Υ στην εργασία σας θα μπορούσατε να πείτε ότι:

Είναι πολύ χρήσιμη αλλά χρονοβόρα διαδικασία

Είναι πολύ χρήσιμη χωρίς να απαιτείται πολύς χρόνος

Είναι πολύ χρήσιμη αλλά είναι δύσκολο το περιβάλλον του προγράμματος

Είναι χρήσιμη και το περιβάλλον του προγράμματος είναι εύκολο

Είναι απλά ένα χρήσιμο εργαλείο

Δεν χρησιμεύει ιδιαίτερα

Δεν χρησιμεύει καθόλου

3) Ποιες λειτουργίες χρησιμοποιείτε στον Η/Υ (πολλαπλή επιλογή)

Ανάγνωση Ιστορικού ασθενούς

Ενημέρωση Ιστορικού ασθενούς

Ενημέρωση Εργαστηριακών Αποτελεσμάτων

Ενημέρωση μετρήσεων Ζωτικών Σημείων

Ανάγνωση Φαρμακευτικής Αγωγής

Πορεία υγείας ασθενούς

Λογοδοσία

Άλλο.....

.....

.....

4) Χρησιμοποιείτε κινητή συσκευή για λειτουργίες πέραν των βασικών πχ μηνύματα,
τηλεφωνήματα;

ΝΑΙ

ΟΧΙ

5) Αν ΝΑΙ ποιες είναι αυτές (πολλαπλή επιλογή)

Internet

Ατζέντα

Σημειώσεις

Υπενθυμίσεις

Άλλο.....

.....

6) Χρησιμοποιείτε κινητή συσκευή ως εργαλείο κατά την επίσκεψή σας στους θαλάμους;

ΝΑΙ

ΟΧΙ

7) Αν ΝΑΙ ποιες είναι αυτές

Ανάγνωση Ιστορικού Ασθενούς

Πορεία υγείας ασθενούς

«Ανάπτυξη εφαρμογής σε κινητά και φορητές συσκευές (PDA's) για εξυπηρέτηση
ιατρών και νοσηλευτών εντός Νοσηλευτικών Ιδρυμάτων»

Ιωάννα Ζορμπά

Ενημέρωση μετρήσεων ζωτικών σημείων
Λογοδοσία
Άλλο.....
.....
.....

8) Αν όχι ποιες λειτουργίες πιστεύετε ότι θα σας εξυπηρετούσαν

Ανάγνωση Ιστορικού Ασθενούς
Πορεία υγείας ασθενούς
Ενημέρωση μετρήσεων ζωτικών σημείων
Λογοδοσία
Άλλο.....
.....
.....

9) Δεν πιστεύω ότι χρειάζεται η χρήση PDA κατά την επίσκεψη στους θαλάμους

