



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΚΡΗΤΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΠΟΛΥΜΕΣΩΝ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ασύρματο σύστημα παρακολούθησης και ενημέρωσης
ασθενών σε νοσοκομείο.

ΣΠΟΥΔΑΣΤΗΣ: ΓΙΩΡΓΟΣ ΜΑΚΡΗΣ
ΕΙΣΗΓΗΤΗΣ: ΑΘΑΝΑΣΙΟΣ ΜΑΛΑΜΟΣ

ΠΡΟΛΟΓΟΣ

Στις μέρες μας είναι επιτακτική ανάγκη να μπορούμε να έχουμε πρόσβαση σε έναν μεγάλο όγκο πληροφοριών άμεσα έτσι ώστε να αποφεύγουμε να σπαταλάμε χρόνο ψάχνοντας μέσα σε ένα σωρό από χαρτιά.

Ακόμα θέλουμε να μπορούμε να έχουμε πρόσβαση στην πληροφορία όχι μόνο από ένα σταθερό σημείο αλλά από οποιοδήποτε σημείο και αν βρισκόμαστε εντός η εκτός γραφείου.

Γι' αυτόν τον λόγο έχουν αναπτυχθεί πολλές εφαρμογές αποθήκευσης και επεξεργασίας δεδομένων σε προσωπικούς υπολογιστές. Ακόμα, με τους φορητούς υπολογιστές , υπολογιστές τσέπης , κινητά τηλέφωνα και την δικτύωση όλων αυτών ασύρματα μέσω WI-FI , 3G κα. πετυχαίνουμε να έχουμε πρόσβαση, ενώ βρισκόμαστε σε κίνηση, σχεδόν από όπου εμείς επιθυμούμε, αρκεί να υπάρχει κάλυψη σήματος και να έχουμε εξουσιοδότηση να δούμε ή να επεξεργαστούμε τα δεδομένα.

Σε ένα νοσοκομείο οι παραπάνω ανάγκες, δηλ η γρήγορη και από διάφορα σημεία, πρόσβαση στον φάκελο ενός ασθενή, είναι απαραίτητες για την σωστή παρακολούθηση αλλά και την έγκαιρη διάγνωση κάποιου επείγοντος περιστατικού.

Στη συγκεκριμένη πτυχιακή θα ενώσουμε την φορητότητα και την αρχειοθέτηση πληροφοριών που χρειάζεται ένας νοσηλευτής και ένας γιατρός, σε μια εφαρμογή.

Περιεχόμενα

1. Ευρετήριο εικόνων.....	5
2. Εισαγωγή.....	7
2.1 Το Project.....	7
2.2 Περιγραφή του project.....	7
2.3 Βασικοί ορισμοί για την κατανόηση ενός Δικτύου.....	9
2.4 Περιγραφή του δικτύου.....	11
2.5 Λογισμικά πακέτα που χρησιμοποιήθηκαν.....	14
3. Εγκατάσταση των Λογισμικών Πακέτων.....	15
3.1 Εγκατάσταση του SQL Server 2005.....	15
3.2 Εγκατάσταση του SQL Server Compact Edition.....	16
3.3 Εγκατάσταση του Microsoft Visual Studio .NET 2005.....	17
3.4 Εγκατάσταση του Windows Mobile 5.0 Pocket PC SDK....	18
4. Εισαγωγή στο Microsoft SQL Server 2005.....	19
4.1 Περιβάλλον εργασίας του SQL Server 2005.....	19
5. Εισαγωγή στην Visual Studio .NET 2005.....	25
5.1 Περιβάλλον εργασίας του Visual Studio .NET 2005.....	26
6. Υλοποίηση της εφαρμογής των σταθερών υπολογιστών.....	33
6.1 Δημιουργία της βάσης δεδομένων.....	33
6.2 Δημιουργία των φορμών αποθήκευση και αναζήτησης των σταθερών υπολογιστών.....	42
6.3 Δημιουργία Query's.....	48
6.4 Επεξήγηση του κώδικας της Win Form.....	49
7. Υλοποίηση της εφαρμογής για τις φορητές συσκευές.....	56

-----ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ-----

7.1	IIS.....	56
7.2	Κατανοώντας την αρχιτεκτονική συγχρονισμού RDA.....	56
7.3	Δημιουργία εικονικού καταλόγου στον IIS και συγχρονισμός των CE Server Agent με τον CE Client Agent.....	58
7.4	Δημιουργία τις εφαρμογής για τις φορητές συσκευές.....	59
7.5	Επεξήγηση του κώδικα των φορητών συσκευών.....	62
8.	Εγχειρίδιο χρήσης.....	77
8.1	Βασικά κουμπιά της εφαρμογής των σταθερών υπολογιστών.....	77
8.2	Επεξήγηση της λειτουργίας της εφαρμογής των σταθερών υπολογιστών για την γραμματεία.....	78
8.3	Επεξήγηση της λειτουργίας της εφαρμογής των σταθερών υπολογιστών για τους γιατρούς.....	81
8.4	Επεξήγηση της λειτουργίας της εφαρμογής των φορητών υπολογιστών τσέπης για τους γιατρούς.....	83
8.5	Επεξήγηση της λειτουργίας της εφαρμογής των φορητών υπολογιστών τσέπης για τους νοσηλευτές.....	86
9.	DeveloperGuide.....	87
9.1	Εφαρμογή σταθερού υπολογιστή.....	87
9.2	Εφαρμογή φορητού υπολογιστή PPC.....	91
10.	Πηγές.....	97

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

• Εικόνα 1.....	11
• Εικόνα 2.....	12
• Εικόνα 3.....	12
• Εικόνα 4.....	13
• Εικόνα 5.....	20
• Εικόνα 6.....	21
• Εικόνα 7.....	21
• Εικόνα 8.....	22
• Εικόνα 9.....	23
• Εικόνα 10.....	24
• Εικόνα 11.....	25
• Εικόνα 12.....	26
• Εικόνα 13.....	27
• Εικόνα 14.....	28
• Εικόνα 15.....	29
• Εικόνα 16.....	30
• Εικόνα 17.....	31
• Εικόνα 18.....	31
• Εικόνα 19.....	32
• Εικόνα 20.....	34
• Εικόνα 21.....	35
• Εικόνα 22.....	41

-----ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ-----

• Εικόνα 23.....	43
• Εικόνα 24.....	44
• Εικόνα 25.....	45
• Εικόνα 26.....	46
• Εικόνα 27.....	47
• Εικόνα 28.....	54
• Εικόνα 29.....	54
• Εικόνα 30.....	55
• Εικόνα 31.....	55
• Εικόνα 32.....	55
• Εικόνα 33.....	57
• Εικόνα 34.....	60
• Εικόνα 35.....	61
• Εικόνα 36.....	61
• Εικόνα 37.....	62
• Εικόνα 38.....	74
• Εικόνα 39.....	78
• Εικόνα 40.....	79
• Εικόνα 41.....	80
• Εικόνα 42.....	82
• Εικόνα 43.....	83
• Εικόνα 44.....	84
• Εικόνα 45.....	84
• Εικόνα 46.....	85
• Εικόνα 47.....	86

ΕΙΣΑΓΩΓΗ

2.1 To project

Καταγραφή ασθενών που εισέρχονται για νοσηλεία σε νοσοκομείο και ημερήσια παρακολούθηση αυτών μέσω προσωπικών υπολογιστών και υπολογιστών τσέπης συνδεδεμένα ενσύρματα και ασύρματα (WI-FI) με ένα κεντρικό εξυπηρετητή.

2.2 Περιγραφή του project

Ο σκοπός αυτής της πτυχιακής είναι να δημιουργήσουμε για ένα νοσοκομείο μια βάση δεδομένων όπου εκεί θα αποθηκεύονται τα στοιχεία όλων των ασθενών που νοσηλεύονται σε αυτό. Αρχικά όταν εισέρχεται ένας ασθενής για νοσηλεία ή για μια απλή εξέταση να μπορεί η γραμματεία μέσα από μια εύχρηστη φόρμα να κάνει εισαγωγή των στοιχείων του από τον υπολογιστή της στην βάση δεδομένων του κεντρικού υπολογιστή (SERVER). Από εκεί θα μπορούν τα στοιχεία αυτά να είναι διαθέσιμα για όποιον έχει δικαίωμα πρόσβασης , ενσύρματα μέσω του προσωπικού του υπολογιστή που βρίσκετε στο γραφείο του, ή ασύρματα με έναν υπολογιστή τσέπης από όποιο σημείο του νοσοκομείου βρίσκετε και έχει σήμα.

Στην συνέχεια όταν ο γιατρός δέχεται ένα ασθενή στον γραφείο του για να τον εξετάσει θα μπορεί μέσω του υπολογιστή του να βλέπει τον φάκελο του ασθενή και τα απαραίτητα στοιχεία του, όπως αν έχει εισαχθεί

ξανά στο νοσοκομείο και για ποιον λόγο έγινε η εισαγωγή. Θα έχει την δυνατότητα όμως να επεξεργάζεται, ακόμα και να συμπληρώνει, τα στοιχεία που κρίνει αυτός ότι είναι απαραίτητα έτσι ώστε ο φάκελος του ασθενή να είναι ενημερωμένος.

Ο γιατρός θα έχει την δυνατότητα να εξετάζει έναν ασθενή και εκτός γραφείου, με την βοήθεια ενός υπολογιστή τσέπης. Ο υπολογιστής τσέπης θα δίνει την δυνατότητα στον γιατρό να βλέπει τα στοιχεία του ασθενή, γιατί θα είναι συνδεδεμένο ασύρματα με τον κεντρικό υπολογιστή του νοσοκομείου. Έτσι η εξέταση θα μπορεί να γίνει ακόμα και σε έναν διάδρομο του νοσοκομείου.

Τέλος οι νοσηλευτές θα μπορούν και αυτοί με τους υπολογιστές τσέπης να επισκεφτούν τους νοσηλευόμενους που είναι στα δωμάτια τους, για τις καθημερινές εξετάσεις τους, για παράδειγμα να μετρήσουν την θερμοκρασία τους, να πάρουν τον καρδιακό τους παλμό, και να αποθηκεύσουν κατευθείαν στον ηλεκτρονικό τους φάκελο τα αποτελέσματα.

Με λίγα λόγια μέσα σε ένα νοσοκομείο θα υπάρχει ένα δίκτυο από προσωπικούς υπολογιστές και υπολογιστές τσέπης, ενσύρματο και ασύρματο, όπου σε αυτό θα μεταφέρονται και θα αποθηκεύονται πληροφορίες των νοσηλευομένων. Οι πληροφορίες θα είναι προσβάσιμες 24 ώρες το 24άωρο από κάποιον προσωπικό υπολογιστή και υπολογιστή τσέπης που βρίσκεται μέσα σε αυτό το δίκτυο.

2.3 Βασικοί ορισμοί για την κατανόηση ενός Δικτύου

Ένα **δίκτυο υπολογιστών** είναι ένα σύνολο από αυτόνομους ή μη αυτόνομους διασυνδεδεμένους υπολογιστές. Οι υπολογιστές θεωρούνται διασυνδεδεμένοι όταν είναι σε θέση να ανταλλάξουν πληροφορίες μεταξύ τους και αυτόνομοι όταν δεν είναι δυνατό κάποιος υπολογιστής να ελέγξει τη λειτουργία κάποιου άλλου (π.χ. εκκίνηση ή τερματισμό).

Τα Δίκτυα Ηλεκτρονικών Υπολογιστών φέρουν τους εξής χαρακτηρισμούς, που καθορίζουν και την κατηγορία τους :

- Ανάλογα με το φυσικό μέσο διασύνδεσης τους χαρακτηρίζονται ως Ενσύρματα ή Ασύρματα
- Ανάλογα με τον τρόπο πρόσβασης σε αυτά χαρακτηρίζονται ως Δημόσια ή Ιδιωτικά δίκτυα
- Ανάλογα με την γεωγραφική κάλυψη του δικτύου χαρακτηρίζονται ως Τοπικά (LAN και WLAN), Μητροπολιτικά (MAN και WMAN), Ευρείας κάλυψης (WAN και WWAN) και Προσωπικά (PAN και WPAN)

Υποσημείωση: οι χαρακτηρισμοί με το πρόσθετο W ανταποκρίνονται στον ασύρματο (Wireless) τρόπο σύνδεσης.

Η **ενσύρματη επικοινωνία**, περιλαμβάνει όλων των ειδών τις εναέριες, τις επίγειες ή τις υπόγειες συνδέσεις αυτού του είδους. Παραδείγματα τέτοιων δικτύων αποτελούν όλα τα χάλκινα καλωδιακά δίκτυα, όπως επίσης και τα οπτικά δίκτυα.

Ως **ασύρματο δίκτυο** χαρακτηρίζεται το τηλεπικοινωνιακό δίκτυο, συνήθως τηλεφωνικό ή δίκτυο υπολογιστών, το οποίο χρησιμοποιεί, ραδιοκύματα ως φορείς πληροφορίας. Η ασύρματη επικοινωνία, σε αντίθεση με την ενσύρματη, δεν χρησιμοποιεί ως μέσο μετάδοσης κάποιον τύπο καλωδίου.

Δημόσια χαρακτηρίζονται τα δίκτυα όπου όποιος θέλει μπορεί να γίνει μέλος σε αυτά.

Ιδιωτικά είναι τα δίκτυα που ενώνουν μια ομάδα υπολογιστών ενός σπιτιού, ενός γραφείου ή μιας εταιρίας και δεν μπορεί να εισέρθει σε αυτά κάποιος άλλος χωρίς την άδεια του διαχειριστή του δικτύου.

Τα **τοπικά δίκτυα** ή και LAN (local area networks) είναι δίκτυα που συνδέουν υπολογιστές σε κοντινές αποστάσεις, π.χ. από υπολογιστές που βρίσκονται σε ένα δωμάτιο μέχρι υπολογιστές που απέχουν μερικά χιλιόμετρα μεταξύ τους.

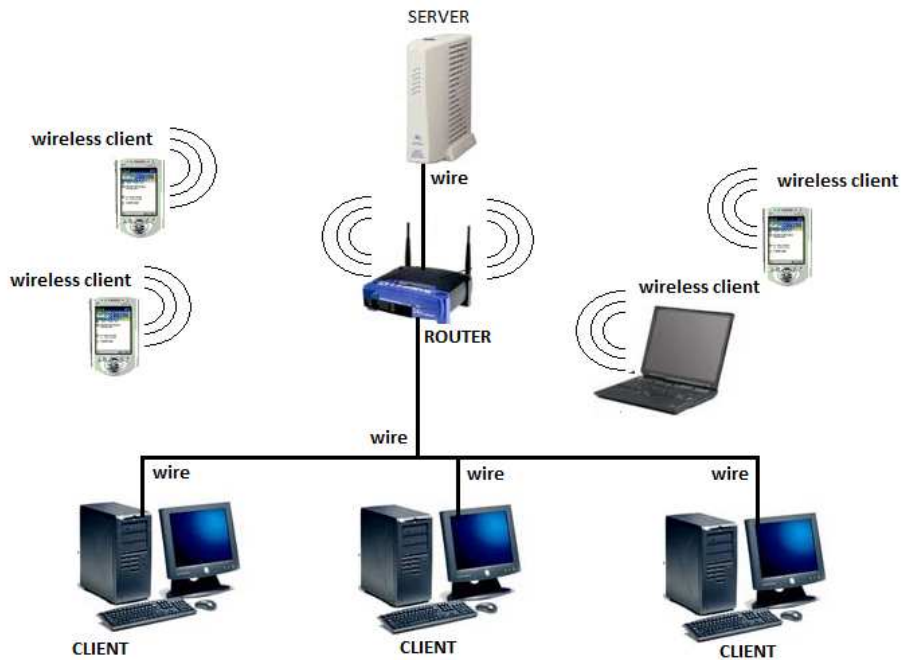
Ένα **μητροπολιτικό δίκτυο** ή και MAN (Metropolitan Area Network) είναι μια μεγαλύτερη εκδοχή ενός τοπικού δικτύου καθώς καλύπτει μεγαλύτερες αποστάσεις, π.χ. από μια ομάδα γειτονικών γραφείων μιας εταιρείας έως μια πόλη.

Τα **δίκτυα ευρείας περιοχής** ή WAN (Wide Area Network) καλύπτουν μεγάλες γεωγραφικές περιοχές, π.χ. από σύνδεση μεταξύ διαφορετικών πόλεων μέχρι μιας ολόκληρης ηπείρου και μπορούν να συνδέσουν ακόμη και περισσότερα από ένα τοπικά δίκτυα καθώς και ομάδες τοπικών δικτύων.

Το **Wi-Fi** είναι μια αποτελεσματική λύση σύνδεσης υπολογιστών σε δίκτυα δεδομένων και συγκεκριμένα με το Διαδίκτυο από ιδιωτικές κατοικίες, δημόσιες περιοχές και οποιαδήποτε άλλη τοποθεσία, χωρίς καλώδια.

Ένας υπολογιστής που είναι συνδεδεμένος με δίκτυο χρησιμοποιώντας κάρτα δικτύου ή Wi-Fi συσκευή μπορεί να είναι σε συνεχή κίνηση, ενώ την ίδια στιγμή να στέλνει και να λαμβάνει δεδομένα από ένα δίκτυο. Αυτό είναι μια λύση που κάνει τους φορητούς υπολογιστές να γίνουν μετακινούμενα δίκτυα. Χάρη στη χρήση της ασύρματης τεχνολογίας, το Wi-Fi σύστημα ελαχιστοποιεί την ανάγκη για την εσωτερική καλωδίωση.

2.4 Περιγραφή του δικτύου



Εικόνα 1. Δίκτυο του νοσοκομείου (Server-Client)

Το παραπάνω σχέδιο είναι ένα αντιπροσωπευτικό δείγμα ενός τυπικού δικτύου όπως αυτό που θα χρησιμοποιήσουμε για να υλοποιήσουμε αυτήν την πτυχιακή

Όπως βλέπουμε στο δίκτυο μας πρέπει να υπάρχει ένα SERVER (Εικόνα 2). Αυτός πρέπει να βρίσκεται σε έναν χώρο όπου θα μπορεί να παραμένει ανοιχτός όλο το 24ωρο και καθημερινά. Για να παραμένει ο server συνέχεια ανοιχτός πρέπει να του παρέχουμε σταθερό ρεύμα συνέχεια και να βρίσκεται σε ένα περιβάλλον με την κατάλληλη θερμοκρασία ώστε να μην υπερθερμαίνεται και σβήνει.



Εικόνα 2. Υπολογιστής Server

Η βάση δεδομένων με τα στοιχεία των ασθενών θα βρίσκεται σε αυτόν τον υπολογιστή, όπου αυτός θα μοιράζει τα δεδομένα σε όποιον υπολογιστή (CLIENT) τα ζητήσει. Για το λόγο ότι σε ένα νοσοκομείο τα δεδομένα πρέπει να είναι διαθέσιμα συνέχεια, πρέπει και ο SERVER να παραμένει συνεχώς ενεργός.

Μια σημαντική συσκευή που είναι απαραίτητη για το δίκτυο μας είναι ο Router (δρομολογητής). Ο Router είναι η συσκευή στην οποία θα συνδέσουμε τους ηλεκτρονικούς υπολογιστές και τους υπολογιστές τσέπης του δικτύου μας και μέσω αυτού θα γίνεται η επικοινωνία με τον SERVER. Με άλλα λόγια ο δρομολογητής αναλαμβάνει τη μεταβίβαση των δεδομένων από και προς τον κατάλληλο υπολογιστή του δικτύου.



Εικόνα 3. Wi-Fi Router

Ο Router που θα χρησιμοποιήσουμε σε αυτό το δίκτυο θα είναι ένας WirelessRouter. Ο WirelessRouter έχει την δυνατότητα να συνδέει υπολογιστές μεταξύ τους ενσύρματα αλλά και ασύρματα. Όπως βλέπουμε και στην Εικόνα1, οι προσωπικοί υπολογιστές (Clients) που θα βρίσκονται στα γραφεία των γιατρών ή στην γραμματεία, θα είναι συνδεδεμένοι ενσύρματα με τον Router για την ανταλλαγή πληροφοριών με τον Server. Οι υπολογιστές τσέπης (WirelessClient) θα συνδέονται ασύρματα με Wi-Fi .



Εικόνα 4. Υπολογιστής Τσέπης (Pocket PC)

Ακόμα και ένας προσωπικός υπολογιστής ή φορητός υπολογιστής θα μπορεί να συνδέεται ασύρματα αρκεί να έχουν εγκατεστημένη μια ασύρματη κάρτα δικτύου.

Οι Clients (πελάτες) στο δίκτυο μας είναι οι συσκευές που μας δίνουν την δυνατότητα να δούμε και να επεξεργαστούμε τα δεδομένα που βρίσκονται στον Server .

Εν κατακλείδι, ένα δίκτυο **client-server** είναι γνωστό ως δίκτυο πελάτη-διακομιστή. Ο πελάτης είναι ο υπολογιστής ενός χρήστη που αποστέλλει αιτήματα προς ένα άλλο υπολογιστή, τον server-διακομιστή. Ο server λαμβάνει τα αιτήματα-requests των πελατών clients και (συνήθως) απαντάει σε αυτά παρέχοντας κάποια δεδομένα.

2.5 Λογισμικά πακέτα που χρησιμοποιήθηκαν

Για την ολοκλήρωση του Project χρησιμοποιήθηκαν τα παρακάτω λογισμικά πακέτα :

- Microsoft SQL Server 2005 Developer Edition
- Microsoft SQL Server Compact Edition
- Microsoft Visual Studio .NET 2005
- Windows Mobile 5.0 Pocket PC SDK

Με το Microsoft SQL Server 2005 Developer Edition φτιάξαμε την βάση δεδομένων όπου σε αυτή θα αποθηκεύονται όλα τα στοιχεία των ασθενών. Αυτό θα πρέπει να είναι εγκατεστημένο στον Server, γιατί όπως αναφέραμε παραπάνω ο Server θα είναι αυτός που θα αποθηκεύει, θα διαχειρίζεται και θα μοιράζει τις πληροφορίες που θα του ζητούν οι Clients.

Το Microsoft SQL Server Compact Edition δεν είναι ένα αυτόνομο λογισμικό πακέτο αλλά ένα εργαλείο που δίνει την δυνατότητα στον SQL Server να φτιάχνει βάσεις δεδομένων για mobile συσκευές και να κάνει τον συγχρονισμό των δεδομένων με τον Server.

Με το Microsoft Visual Studio 2005 δημιουργήσαμε τις εφαρμογές που θα τρέχουν οι Client. Το Microsoft Visual Studio 2005 δεν χρειάζεται να είναι εγκατεστημένο σε κάποιον Client, αφού όταν δημιουργήσουμε της φόρμες, αυτές θα μετατραπούν σε εκτελέσιμα αρχεία με κατάληξη **.exe** και θα φορτωθούν μόνο αυτά σε κάθε Client.

Όπως το SQL Server Compact Edition έτσι και το Windows Mobile 5.0 Pocket PC SDK δεν είναι κάποιο λογισμικό πακέτο, αλλά ένα πρόσθετο εργαλείο του Microsoft Visual Studio που δίνει την δυνατότητα στο Visual Studio να φτιάχνει εφαρμογές για υπολογιστές τσέπης, κινητά τηλέφωνα και γενικά για οποιαδήποτε Mobile συσκευή.

Ούτε το Windows Mobile 5.0 Pocket PC SDK χρειάζεται να είναι εγκατεστημένο σε κάποια φορητή συσκευή, επειδή μόλις δημιουργήσουμε την εφαρμογή θα την μετατρέψουμε σε αρχεία **.CAB** που είναι το αντίστοιχο εκτελέσιμο (.exe) των σταθερών συσκευών.

3.0 Εγκατάσταση των Λογισμικών Πακέτων

3.1 Εγκατάσταση του SQL Server 2005

Όταν πήρα τη απόφαση να ασχοληθώ με την συγκεκριμένη πτυχιακή το πρώτο που έπρεπε να κάνω είναι να εγκαταστήσω όλα τα λογισμικά πακέτα που ανέφερα στην προηγούμενη ενότητα στο υπολογιστή μου. Δυστυχώς συνάντησα κάποιες δυσκολίες που ευτυχώς ξεπεραστήκαν. Οι δυσκολίες που συνάντησα έχουν να κάνουν με το λειτουργικό σύστημα που τρέχει ο υπολογιστής μου, διότι ο SQL Server 2005 Developer Edition δεν είναι απόλυτα συμβατός με τα καινούρια λειτουργικά συστήματα τις Microsoft, Windows Vista και Windows 7. Έτσι χρειάστηκε να εγκαταστήσω και τις τελευταίες ενημερώσεις τους.

Ο παρακάτω πίνακας δείχνει τα ελάχιστα αρχεία που πρέπει να εγκατασταθούν ώστε ο SQL Server να λειτουργεί σωστά ,ανάλογα με το λειτουργικό πρόγραμμα που εκτελεί ο κάθε υπολογιστής .

	Απαραίτητα	Προαιρετικά
Windows XP	<ul style="list-style-type: none"> • SQL Server 2005 Developer Edition 	<ul style="list-style-type: none"> • Service Pack 1 • Service Pack 2 • Service Pack 3
Windows Vista	<ul style="list-style-type: none"> • SQL Server 2005 Developer Edition • Service Pack 1 • Service Pack 2 • Service Pack 3 	
Window 7	<ul style="list-style-type: none"> • SQL Server 2005 Developer Edition • Service Pack 1 • Service Pack 2 • Service Pack 3 	

3.2 Εγκατάσταση του SQL Server Compact Edition

Για να μπορεί ο SQL Server να επικοινωνεί και να δημιουργεί βάσεις δεδομένων για φορητές συσκευές, πρέπει να εγκατασταθεί και ο SQL Server Compact Edition. Παρακάτω θα δούμε τα αρχεία που πρέπει να εγκαταστήσουμε ώστε να τρέχει σωστά ο SQL Server CE :

- **SQLSERVERCE-EN.MSI**

Αυτό το αρχείο εγκαθιστά τα βασικά λειτουργικά στοιχεία του SQL Server Compact Edition.

- **SSCE31SDK-ENU.MSI**

Εγκαθιστά αρχεία που είναι απαραίτητα για την ανάπτυξη και τον προγραμματισμό μιας εφαρμογής, για mobile συσκευές.

- **SSCE31VSTOOLS-ENU.exe**

Το παρών αρχείο εγκαθιστά τα απαραίτητα εργαλεία ώστε να μπορεί να γίνεται η επικοινωνία του Visual Studio με τον Sql Server CE. Χωρίς αυτό η Visual Studio θα μπορεί να επικοινωνεί μόνο με τον SQL Server.

- **SQLCE30SETUPEN.MSI**

Εγκαθιστά απαιτούμενα εργαλεία στον Web Server έτσι ώστε να μπορούν ο SQL Server με τον SQL Server Compact Edition να συγχρονίζουν τα δεδομένα τους.

- **SSCE31BOL-EN.MSI**

Εγκαθιστά το ηλεκτρονικό βιβλίο(manual) του SQL Server CE.

Σημείωση : τα αρχικά EN και ENU στα αρχεία δείχνουν ότι πρόκειται για την αγγλική έκδοση. Εάν τρέχουμε άλλες γλώσσες εκτός την αγγλική στον SQL Server ,τότε θα πρέπει να κατεβάσουμε την αντίστοιχη γλωσσά.

3.3 Εγκατάσταση του Microsoft Visual Studio .NET 2005

Για να εγκαταστήσουμε τη Visual Studio 2005 σε έναν υπολογιστή, θα πρέπει να ξέρουμε ότι στα λειτουργικά συστήματα Windows Vista και Windows 7 υπάρχουν ασυμβατότητες και εκτός από την Visual θα πρέπει να εγκαταστήσουμε και τις ενημερώσεις, έτσι ώστε να είναι συμβατά με το υπάρχον λειτουργικό που τρέχουμε στον υπολογιστή μας.

Ο επόμενος πίνακας μας δείχνει τα απαραίτητα αρχεία που πρέπει να έχει ένας υπολογιστής ανάλογα με το λειτουργικό πρόγραμμα που τρέχει :

	Απαραίτητα	Προαιρετικά
Windows XP	<ul style="list-style-type: none"> • Visual Studio2005 	<ul style="list-style-type: none"> • Service Pack 1 KB926601
Windows Vista	<ul style="list-style-type: none"> • Visual Studio2005 • Service Pack 1 KB926601 • Service Pack KB932232 	
Window 7	<ul style="list-style-type: none"> • Visual Studio2005 • Service Pack 1 KB926601 • Service Pack KB932232 	

3.4 Εγκατάσταση του Windows Mobile 5.0 Pocket PC SDK

Το Windows Mobile 5.0 Pocket PC SDK το εγκαθιστούμε για να μπορεί η Visual Studio να δημιουργεί εφαρμογές για mobile συσκευές. Δεν χρειάζεται κάτι ιδιαίτερο για την εγκατάσταση του αρκεί να είναι ήδη εγκατεστημένη η Visual Studio και το ActiveSync στον υπολογιστή που θα δημιουργήσουμε την εφαρμογή μας.

Το ActiveSynce είναι ένα πρόγραμμα της Microsoft που συνδέει μια φορητή συσκευή με ένα υπολογιστή μέσω ενός καλωδίου USB. Αφού γίνει η σύνδεση, τότε μπορεί ο χρήστης να μεταφέρει δεδομένα όπως φωτογραφίες και τραγούδια από την μια συσκευή στην άλλη. Η Visual

Studio μπορεί με το ActiveSync να τρέχει τα προγράμματα που δημιουργήθηκαν σε αυτήν κατεύθυνση στην φορητή συσκευή.

Τα Windows Vista και τα Windows 7 έχουν εγκατεστημένο ήδη την εξέλιξη του ActiveSync που το ονομάζουν Windows Mobile Device Center.

4.0 Εισαγωγή στο Microsoft SQL Server 2005

Η πλατφόρμα Microsoft SQL Server είναι μια σχεσιακή βάση δεδομένων Διαχείρισης Συστήματος (RDBMS) σχεδιασμένο να λειτουργεί σε πλατφόρμες που κυμαίνονται από φορητούς υπολογιστές έως και μεγάλους Servers πολλαπλών επεξεργαστών. Ο SQL Server χρησιμοποιείται συνήθως ως σύστημα υποστήριξης για ιστοσελίδες, ή ακόμα και για αποθήκευση δεδομένων σε τοπικά δίκτυα, όπου αυτός μπορεί να υποστηρίξει χιλιάδες ταυτόχρονους χρήστες.

4.1 Περιβάλλον εργασίας του SQL Server 2005

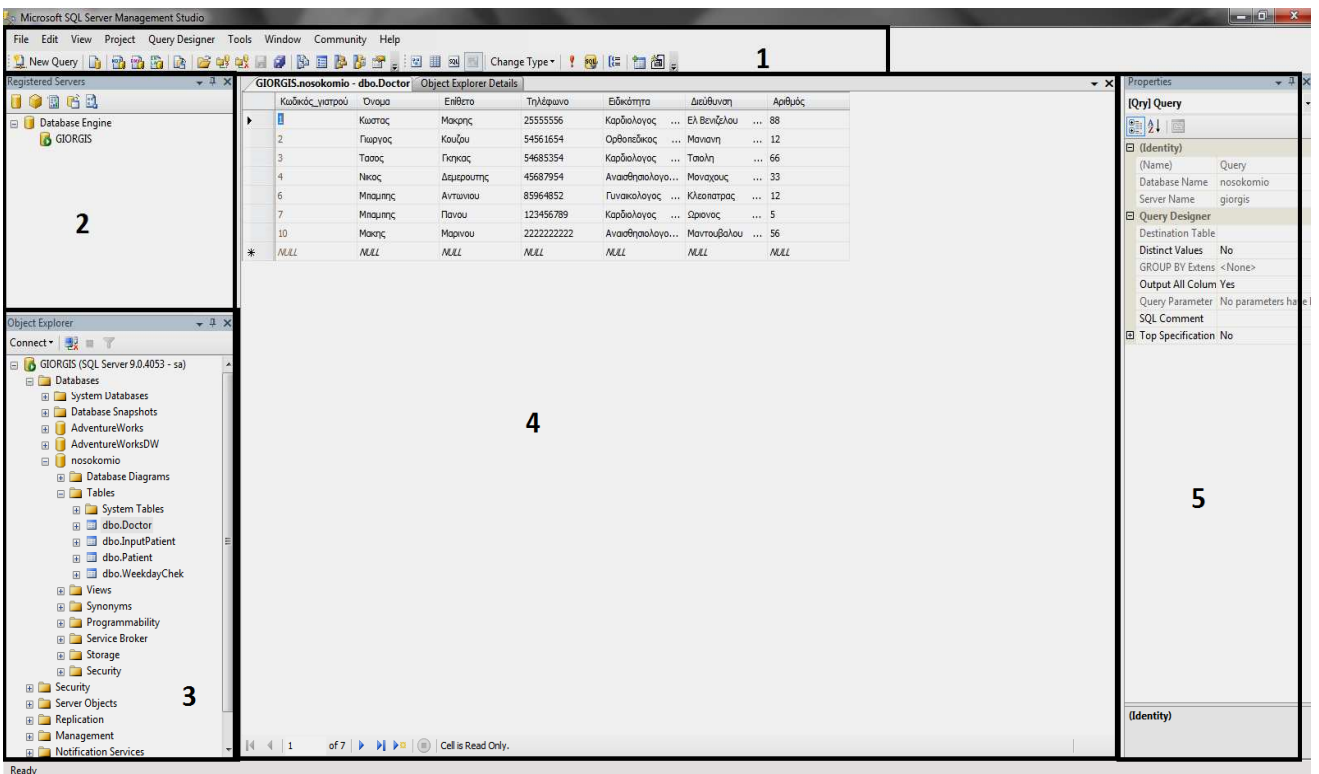
Πατώντας το SQL Server Management Studio από το menu έναρξης των Windows θα εμφανιστεί η παρακάτω εικόνα :



Εικόνα 5. Είσοδος στον SQL Server

Στο πρώτο πεδίο (Server Type) επιλέγουμε τον τύπο Server που θέλουμε να συνδεθούμε. Οι επιλογές είναι οι εξής: Database Engine, Analysis Services, Reporting Services, SQL Server Compact Edition και Integration Service. Εμείς για το συγκεκριμένο project επειδή θέλουμε να δημιουργήσουμε μια βάση δεδομένων επιλέγουμε να συνδεθούμε με το Database Engine.

Το δεύτερο πεδίο μας ζητάει να επιλέξουμε το όνομα του Server. Στο τρίτο να ορίσουμε αν η πιστοποίηση θα γίνεται από τα Windows (Window Authentication) ή από τον Sql Server (Sql Server Authentication). Πληκτρολογώντας το User Name και το Password ανοίγει το περιβάλλον εργασίας του Sql Server 2005 (Εικόνα 6). Το User Name και το Password είναι τα ίδια με αυτά που μας ζητήθηκε να ορίσουμε όταν κάναμε την εγκατάσταση τον SqlServer.



Εικόνα 6. Περιβάλλον εργασίας του SQL Server

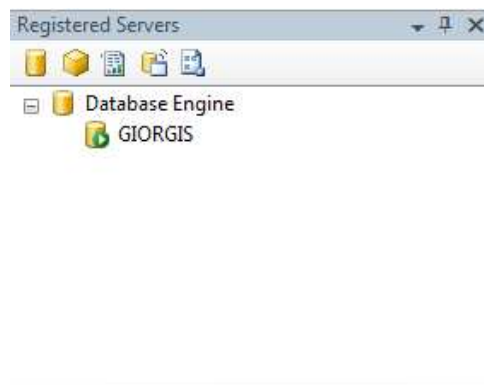
Όπως βλέπουμε (Εικόνα 6) το περιβάλλον του SQL Server Management Studio (SSMS) αποτελείται από 5 βασικές περιοχές. Αυτές έχουν αριθμηθεί και επιλεχτεί στην παραπάνω εικόνα.



Εικόνα 7. Η γραμμή του menu με τη μπάρα εργαλείων.

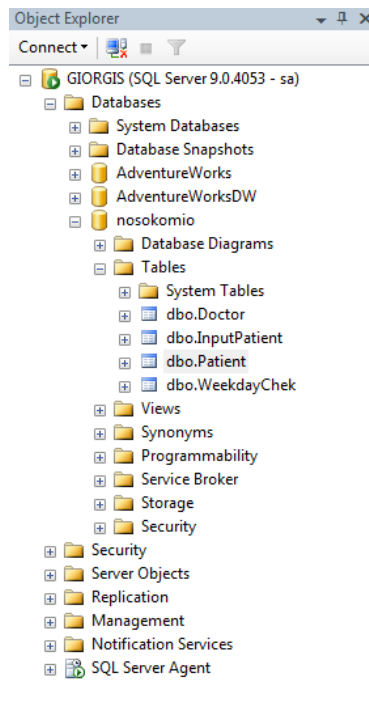
Ξεκινάμε με την πρώτη περιοχή που έχουμε επιλέξει (Εικόνα 7). Είναι η μπάρα του menu μαζί με την μπάρα εργαλείων. Στη μπάρα του menu όπως και σε κάθε πρόγραμμα, βρίσκουμε όλες τις λειτουργίες του

προγράμματος από το Save μέχρι και όλα τα εργαλεία που υποστηρίζει το SSMS. Ακριβώς από κάτω βρίσκεται η μπάρα εργαλείων που μας δίνει την δυνατότητα να εκτελούμε κάποιες ενέργειες άμεσα χωρίς να χάνουμε χρόνο ψάχνοντας μέσα στην γραμμή του menu. Ενδεικτικά μερικά από τα κουμπιά είναι η δημιουργία ερωτημάτων (Queries) για μια βάση δεδομένων που έχουμε δημιουργήσει και το κουμπί της αποθήκευσης. Εκτελώντας κάποια συγκεκριμένη εργασία προσθέτονται ή αφαιρούνται αυτόματα τα κατάλληλα κουμπιά ώστε να μας διευκολύνουν στην υλοποίηση της.



Εικόνα 8. Registered Servers

Η δεύτερη περιοχή που έχουμε επιλέξει (Εικόνα 8) μας βοηθά να αποθηκεύουμε και να διαλέγουμε τους Server που χρησιμοποιούμε. Το SSMS έχει την δυνατότητα να συνδέεται με διαφόρους Server ώστε να μπορεί να επεξεργάζεται βάσεις που βρίσκονται στον κάθε ένα από αυτούς. Στην εικόνα 8 βλέπουμε ότι υπάρχει καταχωρημένος ο Server με το όνομα Giorgis.



Εικόνα 9. Object Explorer

Η τρίτη περιοχή που έχει επιλεχτεί είναι μια από τις βασικότερες. Αυτή είναι ο Object Explorer (Εικόνα 9).

Βλέπουμε ότι ο Object Explorer μας δείχνει τα περιεχόμενα ενός Server. Στην περίπτωση αυτή ο Server μας έχει το όνομα GIORGIS, που συνήθως είναι και το όνομα του υπολογιστή που είναι εγκατεστημένος ο SqlServer. Τα περιεχόμενα που θα μας απασχολήσουν περισσότερο στο project αυτό είναι οι βάσεις δεδομένων (Databases). Εκτός από τις Deafault βάσεις που έχει μπορούμε να δημιουργήσουμε και εμείς μια δική μας με τους πίνακες και τα πεδία που εμείς επιθυμούμε. Στην παραπάνω εικόνα την βάση δεδομένων nosokomio την έχουμε δημιουργήσει εμείς για να υλοποιήσουμε την πτυχιακή άσκηση. Εκτός από τις βάσεις ο Object

Explorer μας δίνει την δυνατότητα να επεξεργαστούμε τις ιδιότητες της κάθε βάσης, όπως τις παραμέτρους σύνδεσης της κάθε βάσης με άλλες, τις παραμέτρους ασφάλειας και τον τύπο επικοινωνίας αυτής με τις άλλες βάσεις.

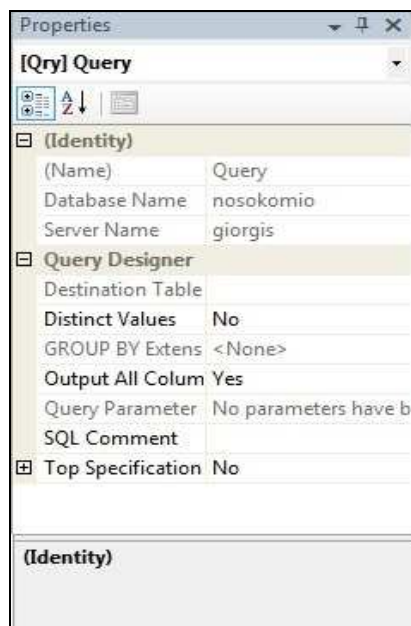
Σημαντικά είναι και τα κουμπιά του Object Explorer, που μας δίνουν την δυνατότητα να συνδεθούμε με διαφορετικούς τύπους Server, ή να διακόψουμε την σύνδεση μας με το κουμπί Disconnect.

Ο Object Explorer Details που είναι το πεδίο 4 που έχουμε επιλέξει, ουσιαστικά είναι η περιοχή που δημιουργούμε και επεξεργαζόμαστε τις βάσεις μας και τις ιδιότητες τους. Με λίγα λόγια είναι η περιοχή εργασίας του SSMS.

GIORGIS.nosokomio - dbo.Patient		Object Explorer Details								
	Κωδικός_ασθενή	Κωδικός_γιατρού	Όνομα	Επίθετο	Τηλέφωνο	Εθνικότητα	Ημερομηνία_γέ...	Διεύθυνση	Αριθμός	Ασφαλιστικός_...
▶	1	1	Μανωλης	Μαρολας	849464654	Ελληνική	21/1/2003 12:0...	Αχιλαου ...	78	ΙΚΑ
	10	1	Βίβη	Μαρκου	859641454	Ελληνική	13/3/1998 12:0...	Γκαγκαρη ...	99	ΤΕΒΕ
	13	3	Θωμας	Τσιμπος	456465454	Ελληνική	28/8/2000 12:0...	Μανωλιου ...	44	ΙΚΑ
	16	4	Κατη	Μακρη	489754324	Ελληνική	31/1/1999 12:0...	Τσαση ...	15	ΟΓΑ
	19	4	Ποπη	Νανου	896541233	Αλβανική	5/2/1993 12:00:...	Μαχαλα ...	2	ΙΚΑ
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Εικόνα 10. Object Explorer Details.

Στην εικόνα 10 βλέπουμε ότι έχουμε ανοίξει το table με το όνομα Patient της βάσης nosokomio και βλέπουμε τα δεδομένα. Αυτά μπορούμε να τα επεξεργαστούμε προσθέτοντας ή αφαιρώντας μια εγγραφή και ρυθμίζοντας ανάλογα τις ιδιότητες της.



Εικόνα 11. Properties Explorer

Τέλος η περιοχή 5 (Εικόνα 11) που ονομάζεται Property Explorer είναι αυτή που μας δίνει την δυνατότητα να βλέπουμε και να επεξεργαζόμαστε τις ιδιότητες του κάθε αντικειμένου που έχει επιλεγθεί.

5.0 Εισαγωγή στην Visual Studio .NET 2005

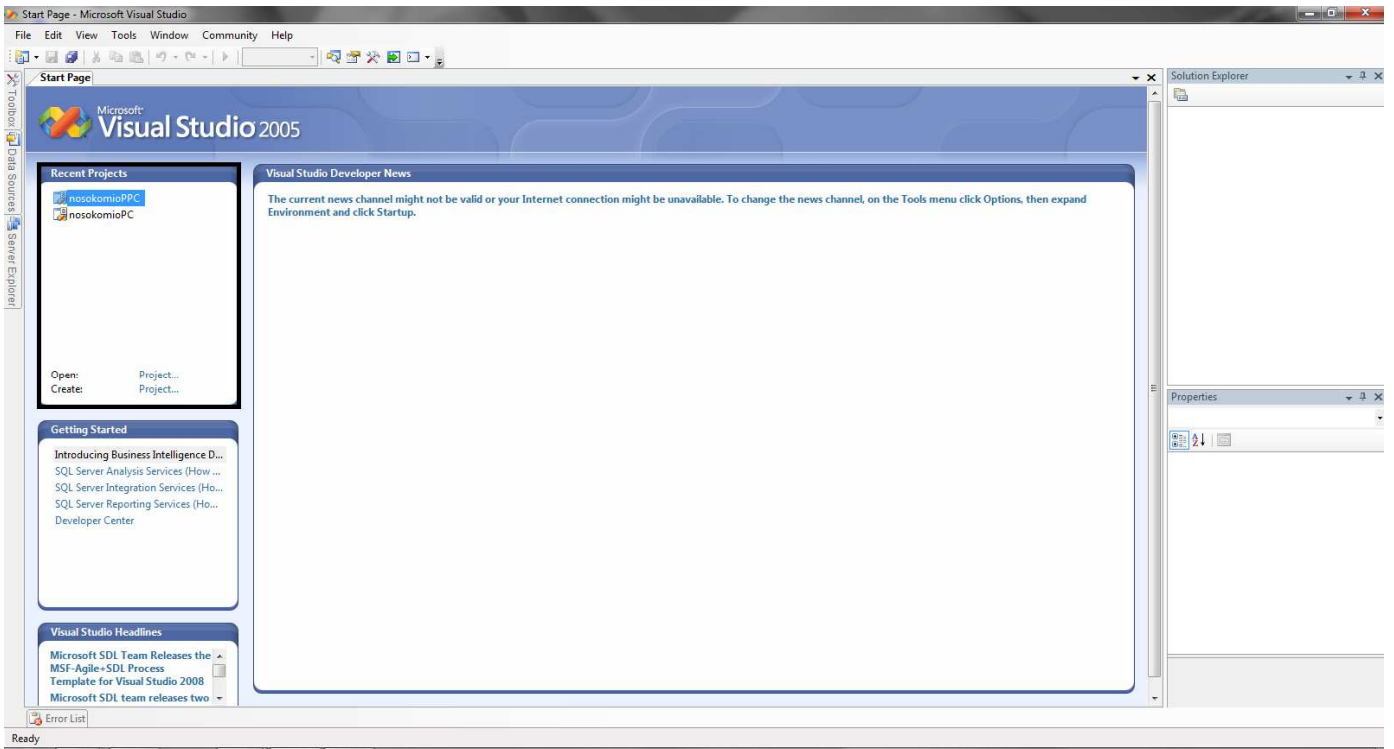
Η .NET είναι μια πλατφόρμα ανάπτυξης εφαρμογών σε περιβάλλον Windows. Σύμφωνα με την Microsoft, έχει ως σκοπό την απλοποίηση της ανάπτυξης εφαρμογών «κρύβοντας» τις τεχνικές λεπτομέρειες υλοποίησης πολλών λειτουργιών, όπως διαχείριση μνήμης, επικοινωνία μέσω δικτύου, είσοδο/έξοδο από συσκευές, αφήνοντας έτσι τον προγραμματιστή ελεύθερο να επικεντρωθεί στην «λογική» του προγράμματος.

Η .NET υποστηρίζει πληθώρα γλωσσών προγραμματισμού οι οποίες είναι ειδικά σχεδιασμένες για αυτό, όπως C#, Visual Basic.NET, J++ και managed C++. Στην

πραγματικότητα, η .NET καταλαβαίνει μόνο μια γλώσσα προγραμματισμού, την Microsoft Intermediate Language (MSIL). Συνεπώς, οποιαδήποτε γλώσσα προγραμματισμού μπορεί να μεταγλωττιστεί σε MSIL μπορεί να τρέξει στην πλατφόρμα .NET . Ο χρήστης μπορεί ακόμα να γράψει απευθείας ένα πρόγραμμα σε MSIL στο Notepad να το κάνει compile και να το τρέξει στο .NET!

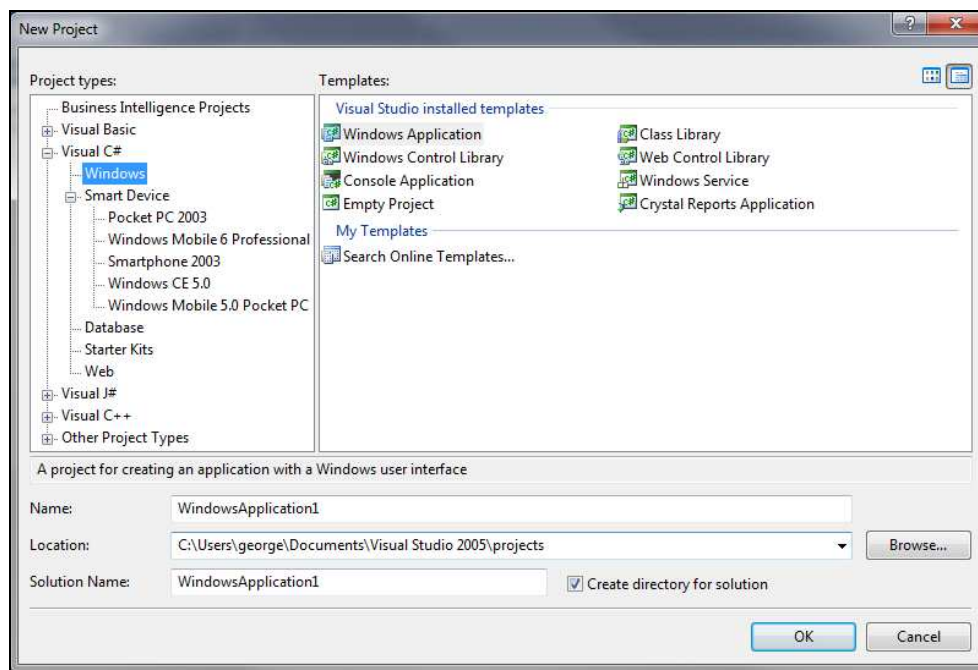
Ακόμα η .NET είναι και ένα εργαλείο προγραμματισμού για εφαρμογές που τρέχουν σε φορητές συσκευές όπως , Pocket PC, PDA , Smartphone. Με αυτή θα δημιουργήσουμε τις φόρμες για την εγγραφή των στοιχείων στους προσωπικούς υπολογιστές και υπολογιστές τσέπης.

5.1 Περιβάλλον εργασίας του Visual Studio .NET 2005



Εικόνα 12. Η αρχική σελίδα της .NET

Ανοίγοντας την .NET από το menu έναρξης των Windows εμφανίζεται η αρχική σελίδα. Στο σημείο που αξίζει να δώσουμε ιδιαίτερη προσοχή είναι το σημείο που έχουμε επιλέξει με μαύρο χρώμα που γράφει Recent Projects (Εικόνα 12). Εκεί βρίσκουμε παλιότερα project που έχουμε δημιουργήσει (πχ nosokomioPC) ή μπορούμε να δημιουργήσουμε ένα καινούριο από την επιλογή Create Project. Επιλέγοντας Create Project θα εμφανιστεί ένα παράθυρο (Εικόνα 13) όπου εκεί θα διαλέξουμε το είδος του project που θέλουμε να δημιουργήσουμε, την γλώσσα προγραμματισμού που επιθυμούμε και το όνομα που θα το ονομάσουμε.

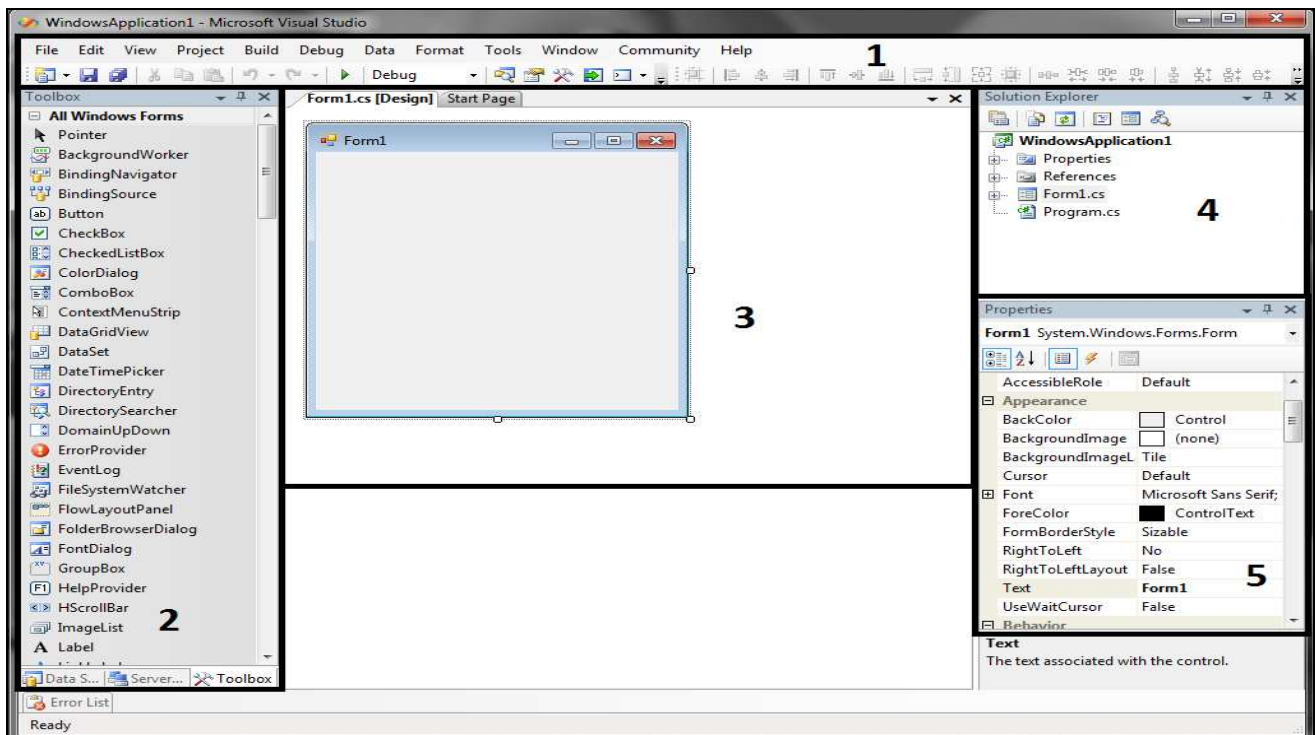


Εικόνα 13. New Project

Όπως βλέπουμε στην περιοχή που ονομάζεται, Project Type μπορούμε να διαλέξουμε την γλώσσα προγραμματισμού που θέλουμε να δουλέψουμε πχ. Visual Basic

,C#,J# C++. Επιλέγοντας μια από αυτές, στην συγκεκριμένη περίπτωση έμεις διαλέξαμε την C#, μας δίνεται η δυνατότητα να διαλέξουμε αν θα δημιουργήσουμε μια εφαρμογή για Windows που θα τρέχει σε σταθερό υπολογιστή ή μια mobile εφαρμογή που θα τρέχει σε φορητές συσκευές όπως Pocket PC και smart phone.

Στο δίπλα περιθώριο (Templates) διαλέγουμε πιο συγκεκριμένα την εφαρμογή που θέλουμε να δημιουργήσουμε, αν θα είναι ένα κενό Project ,μια Class Library κτλ. Συμπληρώνοντας το όνομα και την θέση που θέλουμε να αποθηκεύσουμε το Project, ανοίγει το βασικό περιβάλλον της .NET (Εικόνα 14). Στην συγκεκριμένη περίπτωση εμείς έχουμε διαλέξει μια Windows Form με την γλώσσα C#.



Εικόνα 14. Περιβάλλον του .NET

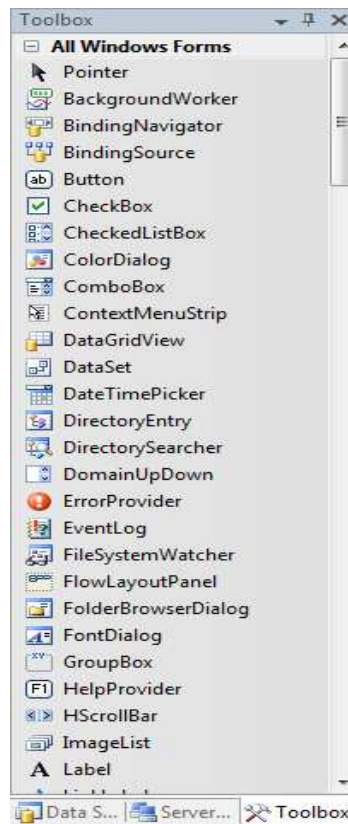
Όπως σε κάθε πλατφόρμα υπάρχουν οι περιοχές εργασίας, σαν αυτή που έχουμε επιλέξει και στην παραπάνω φωτογραφία. Αρχίζοντας με την περιοχή 1 βλέπουμε την μπάρα του μενυ ,με το τις επιλογές file ,open και γενικά όλα τα εργαλεία που υποστηρίζει η πλατφόρμα.(Εικόνα 15).



Εικόνα 15. Μπάρα menu και μπάρα εργαλείων του .NET

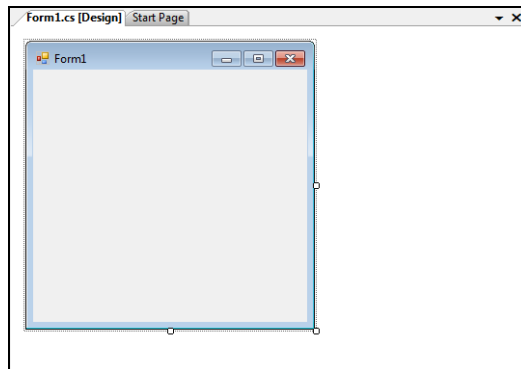
Από κάτω βρίσκεται η μπάρα εργαλείων, που από τα πιο χρήσιμα κουμπιά είναι αυτό του Debug. Το Debug κάνει Compile το πρόγραμμα που έχουμε δημιουργήσει και το τρέχει ώστε να δει ο προγραμματιστής αν είναι σωστή η εφαρμογή του.

Η δεύτερη περιοχή που έχουμε επιλέξει με το νούμερο 2 έχει όλα τα εργαλεία που χρειαζόμαστε για να υλοποιήσουμε μια Windows Form (Εικόνα 16).



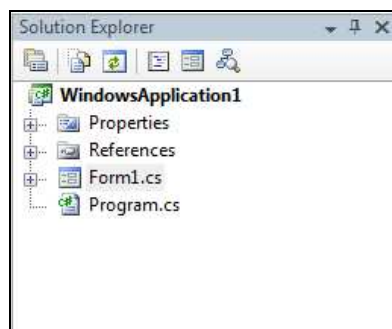
Εικόνα 16. Toolbox Explorer

Επιλέγοντας ένα από τα εργαλεία που θέλουμε όπως ένα button , ή ένα label και ό,τι μπορεί να υπάρχει σε μια Windows form, με drag and drop μπορούμε να το προσθέσουμε στο δικό μας Form. Συνεχίζοντας στην περιοχή 3 , η .NET έχει δημιουργήσει μια κενή φόρμα όπου σε αυτήν εμείς θα προσθέσουμε τα κουμπιά τα label και γενικώς όποιο εργαλείο θέλουμε από το Toolbox Explorer. Ο σκοπός μας είναι να δημιουργήσουμε την δική μας win Form ανάλογα με τις απαιτήσεις μας. (Εικόνα 17)



Εικόνα 17. Κενή winForm.

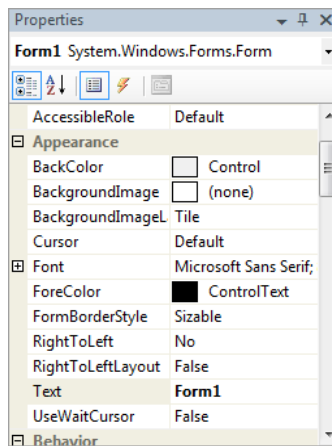
Μόλις τοποθετήσουμε ένα εργαλείο μέσα στην φόρμα, η .NET μας δίνει την δυνατότητα πατώντας διπλό κλικ πάνω σε αυτό, να βλέπουμε τον κώδικα που βρίσκεται από πίσω του , έτσι ώστε να έχουμε το έλεγχο και να προγραμματίζουμε το κάθε ένα εργαλείο ξεχωριστά. Πατώντας δυο φορές πάνω στην φόρμα, εμφανίζεται όλος ο κώδικας αυτής.



Εικόνα 18. Το Solution Explorer του .NET

Δημιουργώντας ένα καινούριο project , η .NET δημιουργεί ένα δέντρο με τις βιβλιοθήκες , τις κλάσεις και τις ιδιότητες του Project, περιοχή 4. Με λίγα λόγια το Solution Explorer (Εικόνα 18) μας διευκολύνει στην

αναζήτηση του κώδικα, γιατί ταχτοποιεί το project ανά κλάση ,winForm και dataset.



Εικόνα 19. Property Explorer του .NET

Τέλος στην περιοχή 5 βρίσκεται το Properties Explorer (Εικόνα 19) όπου εκεί ανάλογα με το αντικείμενο που έχουμε επιλέξει με το mouse ,μπορούμε να ρυθμίσουμε τις ιδιότητές του, το χρώμα το όνομα και πολλά άλλα ανάλογα με το είδος του αντικειμένου.

Σε αυτό το σημείο τελειώσαμε με την περιγραφή των δυο προγραμμάτων που χρησιμοποιήσαμε ώστε να υλοποιήσουμε το project. Και τα δύο είναι προγράμματα που χρησιμοποιούνται ευρέως για ένα σύνολο από εφαρμογές όπως η δική μας. Αυτά μπορούν να χρησιμοποιηθούν για την κατασκευή μικρών προγραμμάτων αλλά και μεγάλων εμπορικών εφαρμογών.

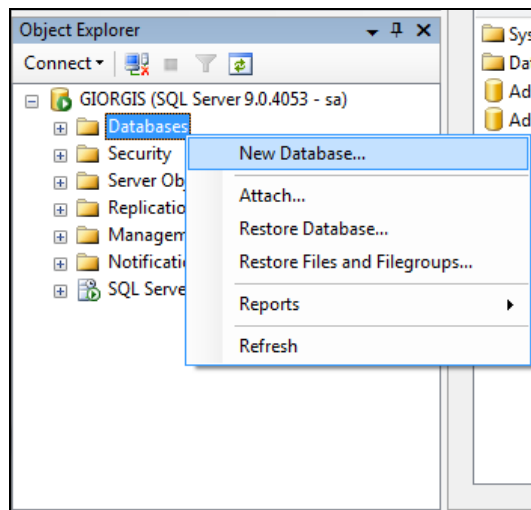
Στο επόμενο κεφάλαιο θα περιγράψουμε πώς με τα παραπάνω εργαλεία θα δημιουργήσουμε την εφαρμογή ανταλλαγής και αποθήκευσης δεδομένων του δικτύου του νοσοκομείου.

6.0 Υλοποίηση της εφαρμογής των σταθερών υπολογιστών

6.1 Δημιουργία της βάσης δεδομένων

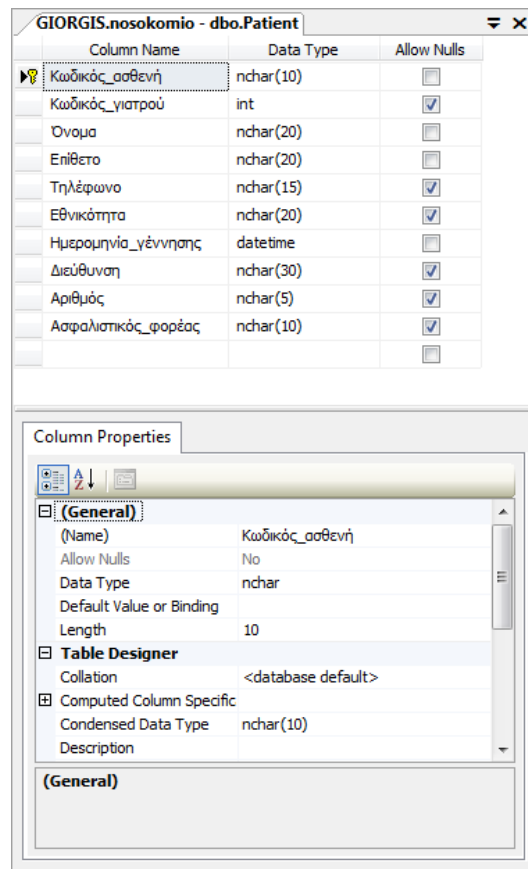
Όπως αναφέραμε σε προηγούμενο κεφάλαιο η δημιουργία της βάσης δεδομένων γίνεται με την βοήθεια του προγράμματος Microsoft SQL Server 2005. Είναι σημαντικό πριν ξεκινήσουμε την υλοποίηση της βάσης, αλλά και ολόκληρης της εφαρμογής, να κάνουμε μια μελέτη που θα μας δώσει τις πληροφορίες που χρειαζόμαστε ώστε η εφαρμογή να είναι λειτουργική. Πχ Θα πρέπει να ξέρουμε ποια στοιχεία είναι απαραίτητα για τον γιατρό ή τον νοσηλευτή, ώστε η εργασία τους να γίνεται ευκολότερα.

Αφού καταλάβουμε ποια πεδία θα πρέπει να έχει η βάση μας συνεχίζουμε στην υλοποίηση της. Ανοίγουμε και μπαίνουμε μέσα στο SSMS από το menu έναρξης των Windows πηγαίνουμε στο Object Explorer, κάνουμε δεξί κλικ στο φάκελο Database και διαλέγουμε New Database. (Εικόνα 20)



Εικόνα 20. Δημιουργία νέας βάσης δεδομένων.

Δίνοντας το όνομα που εμείς θέλουμε στο Wizard που θα εμφανιστεί και πατώντας OK δημιουργείται μια κενή βάση με το όνομα που ορίσαμε. Τώρα θα πρέπει να γεμίσουμε την βάση με τους πίνακες και τα πεδία που εμείς επιθυμούμε. Πηγαίνοντας πάλι στο Object Explorer και ανοίγοντας τον φάκελο με τις βάσεις βλέπουμε αυτήν που μόλις δημιουργήσαμε. Ανοίγοντας και τον φάκελο αυτής θα δούμε ένα φάκελο που ονομάζεται Table. Κάνοντας δεξί κλικ πάνω σε αυτόν επιλέγουμε New Table και θα εμφανιστεί το ακόλουθο παράθυρο. (Εικόνα 21)



Εικόνα 21. Συμπλήρωση ενός πίνακα.

Στην εικόνα 21 βλέπουμε πώς δημιουργούμε έναν πίνακα. Στο πάνω μισό μέρος της εικόνας βλέπουμε ότι έχουμε 3 στήλες. Η πρώτη είναι η Column Name. Σε αυτήν ορίζουμε το όνομα των στηλών από τις οποίες θα αποτελείται ο πίνακας. Στην περίπτωση του συγκεκριμένου Project βλέπουμε ότι έχουμε βάλει τις τιμές Κωδικός Ασθενή ,Κωδικός Γιατρού κτλ. Η διπλανή στήλη μας δίνει την δυνατότητα να ορίσουμε τον τύπο των δεδομένων που θα αποθηκεύονται σε κάθε στήλη που δημιουργούμε. Πχ στο πεδίο 'Κωδικός Γιατρού' εμείς θέλουμε τα περιεχόμενα της στήλης να αποτελούνται αποκλειστικά από αριθμούς .Για τον λόγο αυτό εμείς

δηλώνουμε τον τύπο int ώστε να χρησιμοποιούνται όσο τον δυνατό λιγότερα Byte. Με τον τρόπο αυτό μπορεί να γίνει έλεγχος των δεδομένων που αποθηκεύονται σε κάθε πεδίο. Στην τρίτη στήλη Allow Nulls,,επιλέγουμε αν θα μπορούμε να αφήσουμε κενή μια εγγραφή δηλ να μένει NULL, η αν θα πρέπει αναγκάστηκε να συμπληρώνεται από τον χρήστη. Στο κάτω μισό της εικόνας21 είναι οι ιδιότητες της κάθε στήλης που δημιουργούμε. Μπορούμε από εκεί να ορίσουμε πιο συγκεκριμένες ιδιότητες. Αυτός είναι ο ένας τρόπος να δημιουργήσουμε μια βάση δεδομένων. Ο SSMS μας δίνει την δυνατότητα μέσα από ένα Query να δημιουργήσουμε μια βάση προγραμματιστικά.

Αφού είδαμε πώς να δημιουργούμε μια βάση δεδομένων με τα πεδία της, ήρθε η ώρα να δούμε ποια πεδία επιλέξαμε για το συγκεκριμένο Project.

Το όνομα της Βάση που έχουμε δημιουργήσει είναι 'nosokomio'. Αυτή αποτελείται από 4 πίνακες που τα ονόματα τους είναι :

- Doctor
- Patient
- Input Patient
- Weekday Check

Όπως καταλαβαίνουμε ο πίνακα **Doctor** αποθηκεύει τα στοιχεία των γιατρών που εργάζονται στο νοσοκομείο. **Patient** είναι ο πίνακας που αποθηκεύει τα στοιχεία των ασθενών, ενώ **Input Patient** ο πίνακας που αποθηκεύονται τα στοιχεία του ασθενή για το πότε και τους λόγους που έχει εισαχθεί για νοσηλεία. Ενώ ο τελευταίος πίνακας **Weekday** Check είναι ο πίνακας που αποθηκεύει την ημερήσια κατάσταση του ασθενούς,

εφόσον έχει γίνει εισαγωγή και έχει νοσηλευτεί έστω και μια ημέρα στο νοσοκομείο.

Οι παρακάτω πίνακες δείχνουν τα πεδία, που βρίσκονται μέσα σε κάθε πίνακα καθώς και τις ιδιότητες αυτών.

Πίνακας 1. Doctor

Όνομα Στήλης	Τύπος Δεδομένων	Επιτρέπεται κενή εγγραφή	Περιγραφή
Κωδικός γιατρού	int	Not Null	Πρωτεύον κλειδί. Είναι ο κωδικός του γιατρού μοναδικός για κάθε γιατρό σε αυτό το νοσοκομείο. *Αυτόματη εισαγωγή αριθμού.
Όνομα	nchar(20)	Not Null	Όνομα γιατρού
Επίθετο	nchar(20)	Not Null	Επίθετο γιατρού
Τηλέφωνο	nchar(20)	Null	Τηλέφωνο γιατρού
Ειδικότητα	nchar(30)	Null	Ειδικότητα γιατρού
Διεύθυνση	nchar(30)	Null	Η διεύθυνση γιατρού
Αριθμός	nchar(5)	Null	Αριθμός της διεύθυνση του γιατρού

Πίνακας 2. Patient

Όνομα Στήλης	Τύπος Δεδομένων	Επιτρέπεται κενή εγγραφή	Περιγραφή
Κωδικός ασθενή	nchar(10)	Not Null	Πρωτεύον κλειδί. Κωδικός ΑΜΚΑ του ασθενή, μοναδικός για κάθε ασθενή
Κωδικός γιατρού	int	Null	Δευτερεύον κλειδί. συσχετιζόμενο με τον κωδικό γιατρού από τον πίνακα Doctor
Όνομα	nchar(20)	Not Null	Όνομα ασθενή
Επίθετο	nchar(20)	Not Null	Επίθετο ασθενή
Τηλέφωνο	nchar(15)	Null	Τηλέφωνο ασθενή
Εθνικότητα	nchar(20)	Null	Εθνικότητα ασθενή
Ημερομηνία γέννησης	datetime	Not Null	Ημερομηνία γέννησης ασθενή
Διεύθυνση	nchar(30)	Null	Διεύθυνση ασθενή
Αριθμός	nchar(5)	Null	Αριθμός διεύθυνσης ασθενή
Ασφαλιστικός φορέας	nchar(10)	Null	Ασφαλιστικός φορέας ασθενή

Πίνακας 3. Input Patient

Όνομα Στήλης	Τύπος Δεδομένων	Επιτρέπεται κενή εγγραφή	Περιγραφή
Κωδικός εισαγωγής	int	Not Null	Πρωτεύον κλειδί. Κωδικός του ασθενή μοναδικός για κάθε φορά που κάνει εισαγωγή στο νοσοκομείο *Αυτόματη εισαγωγή αριθμού
Κωδικός ασθενή	nchar(10)	Not Null	Δευτερεύον κλειδί. Κωδικός ΑΜΚΑ του ασθενή συσχετιζόμενο με το κωδικό εισαγωγής από τον πίνακα Patient
Κωδικός γιατρού	int	Null	Είναι ο κωδικός του γιατρού μοναδικός για κάθε γιατρό σε αυτό το νοσοκομείο

Πίνακας 3. Input Patient

Όνομα Στήλης	Τύπος Δεδομένων	Επιτρέπεται κενή εγγραφή	Περιγραφή
Αιτία εισαγωγής	nvarchar(200)	Null	Η αιτία που έκανε τον ασθενή να επισκεφτεί το νοσοκομείο
Ημέρα εισαγωγής	datetime	Not Null	Ημερομηνία επίσκεψης ασθενή
Θεραπευτική αγωγή	nvarchar(200)	Null	Θεραπευτική αγωγή που δόθηκε από τον γιατρό στον ασθενή

Πίνακας 4. Weekday Check

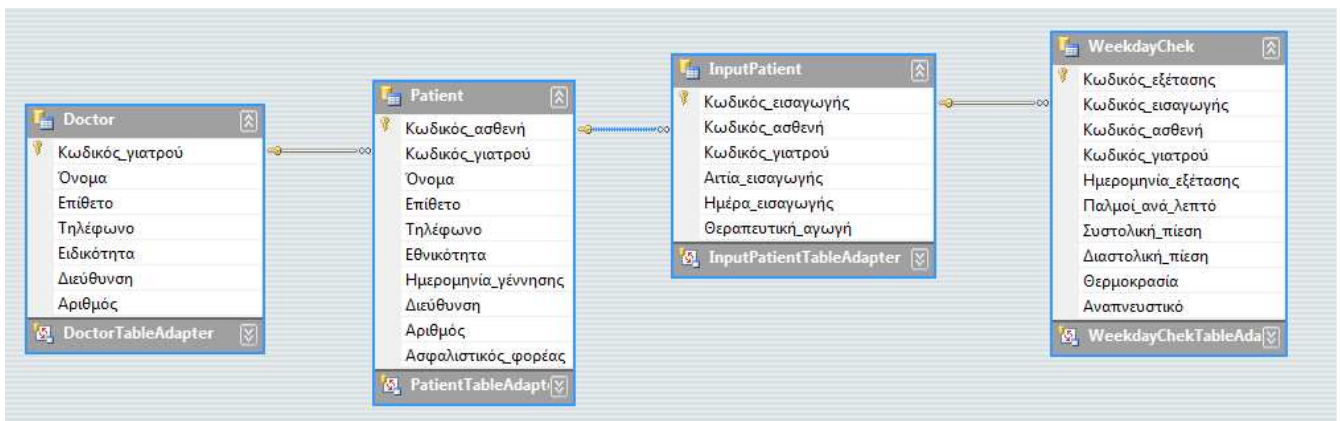
Όνομα Στήλης	Τύπος Δεδομένων	Επιτρέπεται κενή εγγραφή	Περιγραφή
Κωδικός εξέτασης	int	Not Null	Πρωτεύον κλειδί. Μοναδικός κωδικός για κάθε εξέταση που γίνεται από τον νοσηλευτή στον ασθενή εφόσον αυτός βρίσκεται για νοσηλεία πάνω από μια μέρα στο νοσοκομείο
Κωδικός εισαγωγής	int	Not Null	Δευτερεύον κλειδί. Κωδικός εισαγωγής ασθενή στο νοσοκομείο συσχετιζόμενος με τον κωδικό εισαγωγής του πίνακα Input Patient
Κωδικός ασθενή	nchar(10)	Not Null	Κωδικός ΑΜΚΑ ασθενή
Κωδικός γιατρού	int	Not Null	Κωδικός γιατρού που είχε εξετάσει αρχικά τον ασθενή
Ημερομηνία εξέτασης	datetime	Not Null	Ημερομηνία που έγινε η εξέταση του ασθενή από τον νοσηλευτή
Παλμοί ανά λεπτό	nchar(5)	Null	Παλμοί ανά λεπτό που είχε ο ασθενής την στιγμή της εξέτασης από τον νοσηλευτή
Συστολική πίεση	nchar(10)	Null	Συστολική Πίεση που είχε ο ασθενής την στιγμή της εξέτασης από τον νοσηλευτή

Πίνακας 4. Weekday Check

Όνομα Στήλης	Τύπος Δεδομένων	Επιτρέπεται κενή εγγραφή	Περιγραφή
Διαστολική πίεση	nchar(5)	Null	Διαστολική πίεση που είχε ο ασθενής την στιγμή της εξέτασης από τον νοσηλευτή
Θερμοκρασία	nchar(5)	Null	Η θερμοκρασία που είχε ο ασθενής την στιγμή της εξέτασης από τον νοσηλευτή

*Αυτόματη εισαγωγή αριθμού εννοούμε ότι στο Column Property έχουμε ρυθμίσει την επιλογή Identity Specification να είναι Yes και να έχει Identity Seed 1. Δηλαδή σε κάθε νέα εγγραφή που θα γίνεται στην στήλη που έχουμε επιλέξει, να καταχωρείτε αυτόματα ένα αριθμός και σε κάθε επομένη καταχώριση να προστίθεται συν 1 από τον προηγούμενο.

Τώρα θα πρέπει να ορίσουμε την έννοια του συσχετισμού μεταξύ των πινάκων (Relation Ships). Στους πίνακες μεταξύ τους δημιουργούμε έναν συσχετισμό έτσι ώστε τα κοινά τους πεδία να μην μπορούν να έχουν διαφορετικές τιμές. Αυτό γίνεται για τον λόγο ότι αν δυο πεδία σε δυο διαφορετικούς πίνακες έχουν διαφορετικές τιμές, δεν θα έχουμε την δυνατότητα να κάνουμε αναζήτηση των εγγραφών που έγιναν για ένα συγκεκριμένο πεδίο μιας άλλης εγγραφής. Πάντα κάθε πίνακας πρέπει να έχει ένα πρωτεύον κλειδί όπου αυτό δηλώνει ότι οι τιμές του πεδίου αυτού είναι μοναδικές. Το πρωτεύον κλειδί προσδιορίζει μονοσήμαντα μια οντότητα, ενώ ένα δευτερεύον κλειδί προσδιορίζει ένα υποσύνολο όλων των εγγραφών. Στην επόμενη εικόνα θα δούμε τις σχέσεις αυτών των πινάκων.(Εικόνα 22)



Εικόνα 22. Relation Ships των πινάκων.

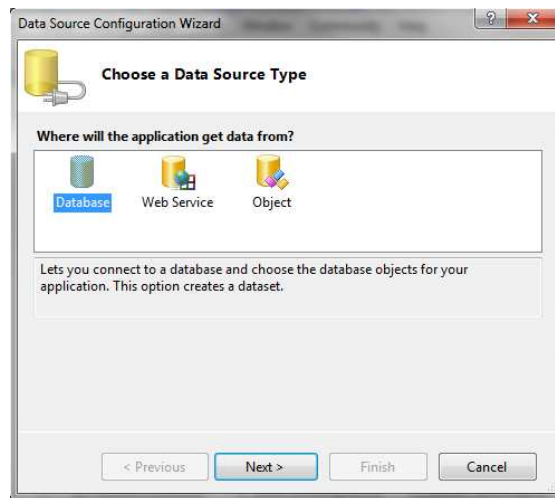
Η εικόνα μας εξηγεί διαγραμματικά την σχέση που έχει ο ένας πίνακας με τον άλλον. Το κλειδί δείχνει ότι η εγγραφή είναι μοναδική ενώ το ∞ ότι οι εγγραφές είναι περισσότερες από μία. Για παράδειγμα, για μία εγγραφή του πεδίου "Κωδικός Γιατρού" του πίνακα Doctor μπορούμε να έχουμε άπειρες εγγραφές του πεδίου "Κωδικός Γιατρού" για τον πίνακα Patient και ούτω κάθε εξής. Έχοντας τελειώσει με την κατασκευή της βάσης δεδομένων θα μπορούσαμε να κάνουμε ερωτήματα (Query) για να κάνουμε αναζητήσεις συγκεκριμένων στοιχείων σε κάθε πίνακα. Επειδή αυτό μπορούμε να το κάνουμε και με την Visual Studio .Net ,θα το δούμε στο επόμενο κεφάλαιο. Στην παράγραφο 6.2 που ακολουθεί αναλύουμε το πώς δημιουργήσαμε τις φόρμες που τρέχουν οι σταθεροί υπολογιστές και οι φορητές συσκευές, καθώς και τον προγραμματισμό που κάναμε σε αυτές.

6.2 Δημιουργία των winForm των σταθερών υπολογιστών.

Σε αυτό το κεφάλαιο θα εξετάσουμε πώς με την Visual Studio .Net θα δημιουργήσουμε τις φόρμες που θα χρησιμοποιούν οι γιατροί και οι νοσηλευτές από κάθε σταθερό υπολογιστή. Με αυτές θα μπορεί ένας χρήστης να αποθηκεύει ή να κάνει αναζήτηση των στοιχείων που τους είναι απαραίτητα.

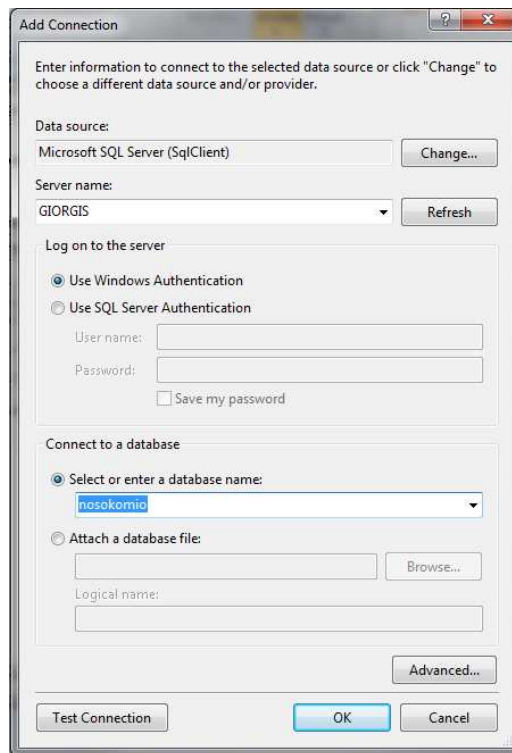
Αρχικά ανοίγουμε την Visual Studio .Net και επιλέγουμε να φτιάξουμε ένα νέο project. Στο Project type επιλέγουμε ότι η γλώσσα που θα χρησιμοποιήσουμε είναι η C# και το Template που θα δημιουργήσουμε είναι ένα Windows application. Το όνομα που του έχουμε δώσει είναι "nosokomioPC". Πατώντας OK η .NET θα δημιουργήσει ένα κενό Windows application (Εικόνα 15) που συναντήσαμε στο κεφάλαιο 5.1

Το επόμενο βήμα και πολύ σημαντικό είναι να συνδέσουμε την βάση δεδομένων που έχουμε δημιουργήσει, με την .NET και την εφαρμογή. Από την menu bar επιλέγουμε Data -> Add new Data Sources και εμφανίζεται το παρακάτω Wizard (Εικόνα 23).



Εικόνα 23. Σύνδεση της βάσης δεδομένων με τη .NET

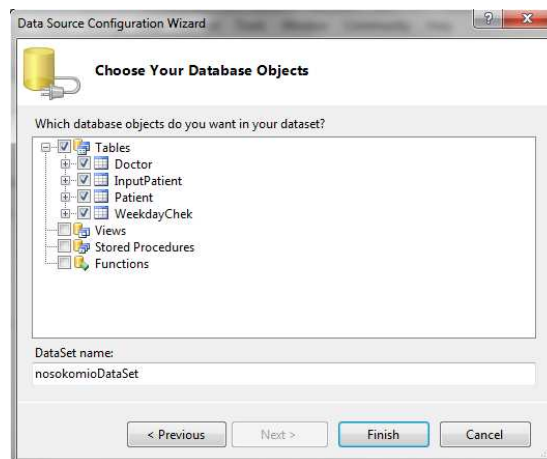
Μας δίνετε η δυνατότητα να επιλέξουμε τι θέλουμε να συνδέσουμε, Database ,Web Service ή Object. Εμείς επιλέγουμε να συνδέσουμε την Database και πατάμε Next. Στην συνέχεια επιλέγουμε να κάνουμε νέα σύνδεση. Πατώντας το κουμπί New Connection εμφανίζεται ένα Wizard (εικόνα 24) όπου από εκεί θα διαλέξουμε τον Server που επιθυμούμε να συνδέσουμε.



Εικόνα 24. Add Connection

Στο Wizard αυτό μας δίνονται οι πληροφορίες της σύνδεσης που θα έχει το project με τον Server. Στο πεδίο Data Source επιλέγουμε τον τύπο της βάσης με την οποία θα συνδεθεί η εφαρμογή. Επιλέγουμε Microsoft SQL Server. Πατώντας το κουμπί Change βλέπουμε τις άλλες επιλογές που έχουμε να διαλέξουμε. Στο Server Name βάζουμε το όνομα 'giogris' δηλαδή το όνομα που δώσαμε στον Server κατά την εγκατάσταση του SQL Server.

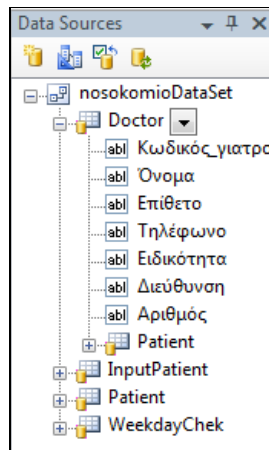
Από κάτω επιλέγουμε να χρησιμοποιήσετε Windows Authentication ενώ στο Select database Name τοποθετούμε το όνομα της βάσης που δημιουργήσαμε. Το όνομα της είναι 'νοσοκομιο' . Πατώντας το κουμπί Test Connection μας δίνεται η δυνατότητα να ελέγξουμε αν η σύνδεση με την συγκεκριμένη βάση είναι εφικτή η όχι. Πατάμε OK για να προχωρήσουμε. Στο επόμενο παράθυρο αφήνουμε τις ρυθμίσεις που υπάρχουν ήδη και πατάμε next. Κάνουμε το ίδιο αφήνοντας τις υπάρχοντες ρυθμίσεις και πατάμε next. Τέλος επιλέγουμε το "table" στο παράθυρο (εικόνα 25) αν θέλουμε να επιλεχτούν όλοι οι πίνακες και τα πεδία που έχουμε δημιουργήσει. Έχουμε την δυνατότητα να μην επιλέξουμε όλους τους πίνακες αλλά κάποιους από αυτούς. Πατάμε Finish και ολοκληρώνουμε την διαδικασία πρόσθεσης της βάσης στο project.



Εικόνα 25.

Εφόσον έχουμε ολοκληρώσει όλη την διαδικασία επιτυχώς στο Project έχουν προστεθεί τα απαραίτητα στοιχεία για την σύνδεση με τον Server και τη βάση ώστε να μπορούμε να επεξεργαζόμαστε τα δεδομένα του.

Ήρθε η ώρα να φτιάξουμε το interface της εφαρμογής προσθέτοντας τα κατάλληλα κουμπιά και Text Box στην Windows φόρμα μας. Αρχικά από το menu bar επιλέγουμε Data -> Show Data Source για εμφανιστεί το παράθυρο Data Source όπου μέσα βρίσκονται οι πίνακες και τα περιεχόμενα αυτών. (Εικόνα 26)



Εικόνα 26. Εμφάνιση του Dataset.

Τώρα επιλέγουμε στο βελάκι δίπλα στον πίνακα Doctor αν τα δεδομένα αυτού του πίνακα θα εμφανίζονται σαν DataGridView ή σαν Details. Εμείς διαλέγουμε την δεύτερη. Το ίδιο κάνουμε και στους άλλους πίνακες, στην συνέχεια παίρνουμε τον πίνακα Patient και με drag and drop το τοποθετούμε μέσα στην φόρμα. Βλέπουμε ότι η .NET δημιουργεί τα textbox, τα label αλλά και την Navigator Bar όπου με αυτήν μπορούμε να προσθέσουμε, να διαγράψουμε και να κινηθούμε σε όλες τις εγγραφές

του πίνακα Patient. Εκτός από τον πίνακα Patient χρειάζεται να προσθέσουμε και άλλα εργαλεία στην WinForm, όπως κουμπιά, ώστε να γίνει λειτουργική. Όλα τα εργαλεία τα βρίσκουμε στο Toolbox Explorer (εικόνα 16) όπου και αυτά με drag and drop προσθέτονται με τον κώδικα που τους πλαισιώνει. Η πρώτη φόρμα που θα δημιουργήσουμε φαίνεται στην εικόνα 27.

Εικόνα 27. WinForm Στοιχεία ασθενή.

Οι ιδιότητες του κάθε εργαλείου υπάρχουν στον Property Explorer αφού πρώτα έχουμε επιλέξει ένα από αυτά. Άρα τα ονόματα των κουμπιών και της φόρμας τα επιλέγουμε από εκεί. Για να δούμε ή να επεξεργαστούμε τον ήδη υπάρχων κώδικα που δημιούργησε η .NET με την προσθήκη των εργαλείων στην φόρμα, αρκεί να κάνουμε διπλό κλικ σε ένα από αυτά τα εργαλεία.

Ή διαφορετικά πηγαίνουμε στο Solution Explorer κάνουμε δεξί κλικ στο όνομα της φόρμας και επιλέγουμε View Code. Και με τους δύο αυτούς τρόπους εμφανίζεται όχι μόνο ο κώδικας της φόρμας που έχουμε επιλέξει, αλλά και όλων των εργαλείων που έχουμε προσθέσει στην WinForm

6.3 Δημιουργία Query's

Αφού έχουμε προσθέσει την βάση δεδομένων και έχουμε δημιουργήσει την πρώτη WinForm στο project, ήρθε η ώρα να φτιάξουμε τα Querys των table. Όπως έχουμε αναφέρει και σε προηγούμενο κεφάλαιο μπορούμε να τα δημιουργήσουμε και στον SSMS, αλλά εμείς θα το κάνουμε με την .NET. Όταν προσθέσαμε την βάση δεδομένων στο project αυτόματα στο Solution Explorer προστέθηκε το 'nosokomioDataSet.xsd' όπου αυτό το αρχείο μας δείχνει τα table και τις συνδέσεις (Relations) αυτών. Κάνοντας δεξί κλικ Open στο αρχείο 'nosokomioDataSet.xsd' εμφανίζονται σε γραφικό περιβάλλον (εικόνα 22) οι πίνακες και τα relations αυτών. Κάνουμε δεξί κλικ στο table adapter του πίνακα που επιθυμούμε επιλέγουμε Add -> Query . Στην συνέχεια επιλέγουμε Use SQL statements -> next -> Select which return rows και στο περιθώριο που μας δίνετε γράφουμε τα Query που εμείς επιθυμούμε.

Ο βασικός τύπος Query που χρησιμοποιούμε σε αυτό το project είναι ο ακόλουθος:

```
«SELECT Κωδικός_ασθενή, Κωδικός_γιατρού, Όνομα, Επίθετο, Τηλέφωνο, Εθνικότητα, Ημερομηνία_γέννησης, Διεύθυνση, Αριθμός,
```


Ασφαλιστικός_φορέας FROM Patient WHERE (Κωδικός_ασθενή = @Κωδικός_ασθενή) »

Στο παραπάνω Query χρησιμοποιούμε τρεις βασικές εντολές SELECT, FROM και WHERE. Μετά το SELECT δηλώνουμε τα πεδία από τα οποία θα πάρουμε εγγραφές. Μετά το FROM δηλώνουμε τον όνομα του πίνακα που βρίσκονται τα πεδία αυτά και μετά το WHERE δηλώνουμε για ποια τιμή ποιου πεδίου θα επιστραφούν οι εγγραφές . Με λίγα λόγια το παραπάνω Query λέει να πάρουμε τις εγγραφές των πεδίων Κωδικός_ασθενή, Κωδικός_γιατρού, Όνομα, Επίθετο, Τηλέφωνο, Εθνικότητα, Ημερομηνία_γέννησης, Διεύθυνση, Αριθμός, Ασφαλιστικός_φορέας από τον πίνακα Patient όπου η εγγραφή του πεδίου «Κωδικός_ασθενή» έχει την τιμή «@Κωδικός_ασθενή». Την τιμή την παίρνουμε από ένα Textbox της φόρμας. Δίνοντας ένα όνομα στο Query αυτό, μπορούμε να το καλούμε όποτε εμείς επιθυμούμε γράφοντας το όνομα του στον κώδικα μας. Ανάλογα δημιουργούμε και άλλα Query με διαφορετικά πεδία για τα άλλα Table.

6.4 Επεξήγηση του κώδικας της Win Form

Σε αυτό το κεφάλαιο θα εξηγήσουμε τον κώδικα της WinForm καθώς και αυτόν που προσθέσαμε για να γίνει η WinForm πιο λειτουργική ανάλογα με το τι θέλουμε να κάνει ο χρήστης. Κάνοντας διπλό κλικ στην WinForm θα εμφανιστεί ο κώδικας . Στην αρχή της σελίδας απαραίτητο είναι να προσθέσουμε τις βιβλιοθήκες που χρειάζεται η εφαρμογή ώστε να

μπορούμε να χρησιμοποιούμε τις εντολές που υπάρχουν σε αυτές. Οι βιβλιοθήκες φαίνονται στον κώδικα που ακλουθεί :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
```

Στο κουμπι Save έχουμε προσθέσει τον παρακάτω κώδικα:

```
if (d)
{
    if (κωδικός_ασθενήTextBox.Text == "" || όνομαTextBox.Text == "" ||
        επίθετοTextBox.Text == "")
    {
        MessageBox.Show("Συμπληρωστε υποχρεωτικά τα πεδια με *", "Message",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        κωδικός_ασθενήTextBox.Focus();
        return;
    }
    if (b)
    {
        MessageBox.Show("Συμπληρωστε την σωστη μερομηνία γέννησης", "Message",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        ημερομηνία_γέννησηςDateTimePicker.Focus();
        return;
    }

    try
    {
        this.Validate();
        this.patientBindingSource.EndEdit();
        this.patientTableAdapter.Update(this.nosokomioDataSet.Patient);
        MessageBox.Show("Οιαλλαγες αποθηκευτηκαν", "Message", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
    }
    catch (System.Exception ex)
    {
        MessageBox.Show("Update failed", "Message", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
}
```

Στον κώδικα αυτό οι μεταβλητές b, d είναι δηλωμένες να είναι τύπου Bool. Στο πρώτο If γίνεται ο έλεγχος αν έχει πατηθεί το κουμπί delete. Γιατί αν έχει πατηθεί πηγαίνουμε κατευθείαν στο try όπου εκεί γίνεται και η αποθήκευση των τιμών από την φόρμα, στην βάση δεδομένων. Στο επόμενο if γίνεται ο έλεγχος αν έχουν συμπληρωθεί τα textbox που απαιτούνται. Αυτά είναι: κωδικός ασθενή, όνομα και επίθετο. Αν όχι τότε ο κώδικας μέσα στο if εμφανίζει ένα message box που μας παροτρύνει να συμπληρώσουμε τα απαιτούμενα πεδία και τοποθετεί τον κέρσορα στο textbox κωδικός ασθενή. Στην συνέχεια με το return ξαναγίνεται ο έλεγχος. Στο τρίτο if γίνεται ο έλεγχος αν έχει αλλάξει η ημερομηνία γέννησης και έχει μπει η σωστή. Εφόσον έχουν συμπληρωθεί τα απαραίτητα πεδία στο try catch βρίσκονται και εκτελούνται οι εντολές αποθήκευσης των δεδομένων. Αν δεν γίνει για κάποιο λόγο η αποθήκευση τότε εμφανίζεται το μήνυμα Update Failed. Με τον τρόπο αυτό μπορούμε να κάνουμε έλεγχο αν τα δεδομένα αποθηκεύτηκαν σωστά η όχι. Το κουμπί "εμφάνιση όλων των εγγραφών" μας εμφανίζει όλες τις εγγραφές που έχουμε αποθηκεύσει και με τα βελάκια αριστερά δεξιά μεταβαίνουμε σε αυτήν που επιθυμούμε.

Ο κώδικας :

```
this.patientTableAdapter.Fill(this.nosokomioDataSet.Patient);
```

που βρίσκετε στο κουμπί εμφάνιση όλων των εγγραφών, λέει στον table adapter να συμπληρώσει (Fill) με τα στοιχεία που έχουμε επιλέξει στα query (κεφάλαιο 6.3) από το nosokomiodataset και το table patient.

Συνεχίζουμε με ένα κομμάτι του κώδικα που βρίσκετε στην πρώτη winForm :

```
Form4 form4 = new Form4();
Form4.form1inForm4 = this;
Form4.Show();
```

Συγκεκριμένα αυτός ο κώδικας βρίσκεται στο κουμπί «Άνοιγμα Φακέλου» του Group Box «Στοιχεία Εισαγωγής Ασθενή». Με αυτόν τον κώδικα καταφέρνουμε να ανοίξουμε μια νέα winForm την «Στοιχεία Εισαγωγής Ασθενή» και να μπορούμε να μεταφέρουμε μια public μεταβλητή από τη μια φόρμα στην άλλη. Έτσι ανοίγοντας την νέα φόρμα μεταφέρουμε το «ΑΜΚΑ » που είναι ο κωδικός του ΑΜΚΑ του ασθενή που έχουμε επιλέξει, από την αρχική φόρμα στην νέα πατώντας το κουμπί.

Στην νέα φόρμα πρέπει να εμφανίσουμε την μεταβλητή από την προηγούμενη φόρμα. Ο κώδικας :

```
this.κωδικός_ασθενήTextBox.Text =
    ((Form1)this.form1inForm4).κωδικός_ασθενήTextBox.Text;
```

τοποθετεί την τιμή που είχε το textbox κωδικός_ασθενή της winForm «Στοιχεία ασθενή» στο textbox κωδικός_ασθενή της winForm «Στοιχεία Εισαγωγής Ασθενή». Έτσι με τον κώδικα :

```
this.inputPatientTableAdapter.FillBykodastheni(this.nosokomioDataSet.InputPatient,
    κωδικός_ασθενήTextBox.Text);
```

όπως έχουμε εξηγήσει προηγουμένως καταφέρνουμε και συμπληρώνουμε τα στοιχεία σύμφωνα με τον κωδικό του ασθενή που μεταφέρθηκε από τη προηγούμενη winForm.

Ένα σημαντικό σημείο του κώδικα που πρέπει να διευκρινίσουμε είναι ότι σε κάθε textbox η μεταβλητή που αποθηκεύετε είναι τύπου String. Ανατρέχοντας στο κεφάλαιο 6.1 θα δούμε ότι η μεταβλητή κωδικός γιατρού και κάποιες άλλες τις έχουμε δηλώσει στην βάση δεδομένων ότι είναι τύπου int. Τότε ο παραπάνω κώδικα δεν θα μπορεί να δουλέψει σωστά και θα πρέπει να βρούμε ένα τρόπο να μετατρέψουμε το περιεχόμενο του textbox από String σε Int. Αυτό το καταφέρνουμε με τον παρακάτω κώδικα :

```
int a = int.Parse("0" + this.κωδικός_γιατρούTextBox.Text);
```

όπου a είναι η μεταβλητή του textbox. Άρα ο κώδικας για την συμπλήρωση των δεδομένων σύμφωνα με τον κωδικό του γιατρού θα είναι ο εξής:

```
this.doctorTableAdapter.FillBykod(this.nosokomioDataSet.Doctor, a);
```

Ακόμα ένα σημείο του κώδικα που πρέπει να δώσουμε προσοχή είναι ο κώδικας που κάνει το textbox «Συμπλήρωσε το ΑΜΚΑ του ασθενή» να αλληλεπιδρά με το Enter του πληκτρολογίου. Όπως φαίνεται παρακάτω:

```
if (e.KeyCode == Keys.Enter)
{
    e.SuppressKeyPress = true;

    patientTableAdapter.FillByAMKA(nosokomioDataSet.Patient,txtEvresiAstheni.Text);
    this.txtEvresiAstheni.Text = "";

    txtEvresiAstheni.Focus();

    if(κωδικός_ασθενήTextBox.Text == "")
        MessageBox.Show("Δεν υπάρχει εγγραφή!Ελεγξτε τον ΑΜΚΑ", "Message",
            MessageBoxButtons .OK,MessageBoxIcon.Information);
}
```

Στο if γίνεται ο έλεγχος αν έχει πατηθεί το Enter η όχι. Αν η σχέση είναι αληθής, δηλαδή έχει πατηθεί το Enter, συνεχίζει στην επομένη γραμμή που λέει στον compiler να μην ακουστεί ο ήχος των Windows, που ακούγετε όταν πατάμε το Enter σε ένα κενό textbox. Συνεχίζοντας ο Table adapter εκτελεί το Query FillByAMKA που έχουμε δημιουργήσει και εξηγήσει στο κεφάλαιο 6.3. Αυτό συμπληρώνει την winForm σύμφωνα με την τιμή που έχουμε εισάγει στο textbox «Συμπλήρωσε το ΑΜΚΑ του ασθενή». Στην συνέχεια καθαρίζει το textbox από την τιμή που είχε ενώ ο κέρσορας εξακολουθεί να μένει στο συγκεκριμένο textbox. Στο επόμενο if γίνεται ο έλεγχος αν εμφανίζετε εγγραφή

στο πεδίο ΑΜΚΑ ασθενή. Αν δεν υπάρχει εγγραφή τότε εμφανίζεται ένα μήνυμα που μας ειδοποιεί ότι δεν υπάρχει εγγραφή.

Ο κώδικας : `this.Close();` που βρίσκεται στο κουμπί ΕΞΟΔΟΣ κλείνει την φόρμα.

Οι παραπάνω εντολές είναι οι εντολές που προσθέσαμε στην εφαρμογή των σταθερών υπολογιστών, όπου οι χρήστες θα αποθηκεύουν και θα αναζητούν τα δεδομένα.

Ολοκληρώνοντας αυτό το κεφάλαιο παραθέτουμε τις φόρμες που δημιουργήσαμε για την υλοποίηση της εφαρμογής.

Εικόνα 28. Αρχική φόρμα εισαγωγή ασθενή

Εικόνα 29. Φόρμα στοιχεία εισαγωγής ασθενή

Εικόνα 30. Φόρμα Στοιχεία Γιατρών

Εικόνα 31. Φόρμα Καθημερινή εξέταση ασθενή

Εικόνα 32. Φόρμες Στοιχεία γιατρού και Στοιχεία Ασθενή

7.0 Υλοποίηση της εφαρμογής για τις φορητές συσκευές

7.1 IIS

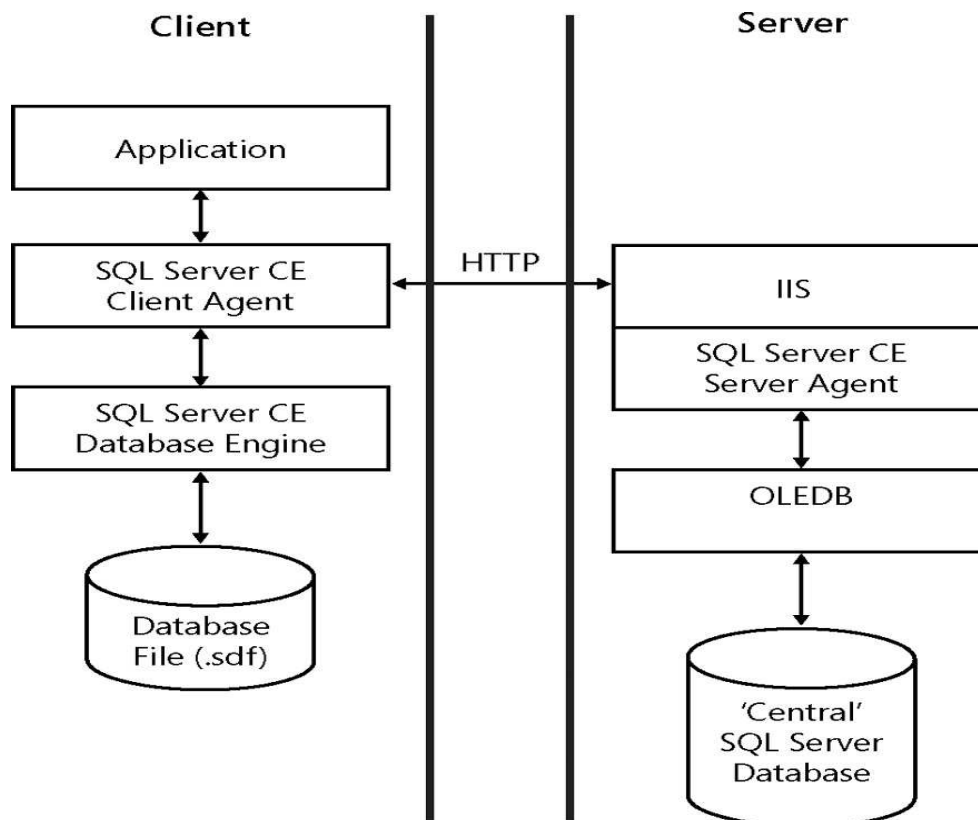
Είναι πολύ σημαντικό να έχουμε εγκαταστήσει στον Server τον IIS. Ο IIS είναι ένας web server για την πλατφόρμα των windows. Δηλαδή είναι ένα πρόγραμμα το οποίο δέχεται αιτήσεις για sites και εφαρμογές που είναι αποθηκευμένες στο μηχάνημα και τις στέλνει σε αυτόν που τις ζήτησε. Επειδή στην εφαρμογή που δημιουργούμε κάθε φορητή συσκευή θα στέλνει και θα δέχεται αιτήσεις, γίνεται κατανοητό ότι η εγκατάσταση του IIS είναι απαραίτητη για να τρέξει η εφαρμογή μας. Συνήθως τρέχει σε λειτουργικό για server αλλά μπορούμε να το εγκαταστήσουμε και σε όλα τα λειτουργικά των Windows.

7.2 Κατανοώντας την αρχιτεκτονική συγχρονισμού RDA

Η εφαρμογή που πρέπει να υλοποιήσουμε στις φορητές συσκευές πρέπει να μπορεί να φορτώνει δεδομένων από μία κεντρική βάση δεδομένων, να κάνει αλλαγές ή προσθήκες τοπικά, και στη συνέχεια να αποστέλλει τις αλλαγές στην κεντρική βάση δεδομένων SQL Server, έτσι ώστε οι άλλοι χρήστες να μπορούν να τις δουν. Αυτό μπορούμε να το υλοποιήσουμε με 3 τεχνικές :

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

Σε αυτό το Project η τεχνική που θα χρησιμοποιήσουμε για την υλοποίηση της εφαρμογής των φορητών συσκευών είναι η πρώτη. Η RDA χρησιμοποιεί τη μηχανή SQL Server CE Database Engine και το SQL Server CE Client Agent στην φορητή συσκευή Client. Όταν ζητούνται δεδομένα από την κεντρική βάση δεδομένων του SQLServer, ο SQLServer CE Client Agent δημιουργεί αίτημα στον SQL Server CE Server Agent μέσω HTTP. Ο server agent τρέχει ως επέκταση Internet Server API (ISAPI) στο IIS. Εκτελείται στο φάκελο Sscesa30.dll, ο οποίος βρίσκεται σε ένα Web site, το οποίο είναι σχεδιασμένο για τη χρήση της RDA. Η αρχιτεκτονική της τεχνικής RDA φαίνεται στην επόμενη εικόνα.



Εικόνα 33. Η αρχιτεκτονική της τεχνικής RDA.

7.3 Δημιουργία εικονικού καταλόγου στον IIS και συγχρονισμός των CE Server Agent με τον CE Client Agent

Για να γίνεται συγχρονισμός των δεδομένων του CE Server Agent και του CE Client Agent χρησιμοποιούμε το εργαλείο Configure Web Synchronization Wizard. Με αυτό δημιουργούμε ένα εικονικό κατάλογο στον IIS .Ορίζουμε τον IIS να έχει την δυνατότητα εισαγωγής δεδομένων στην βάση, όπως επίσης και να έχει άδεια πρόσβασης να εκτελεί οποιαδήποτε απαιτούμενη αλλαγή στα αντικείμενα της βάσης δεδομένων.

Από το menu "Εναρξη" των Windows, ->όλα τα προγράμματα,-> Microsoft Sql Server Compact επιλέγουμε Configure Web Synchronization Wizard. Αφού ανοίξει το Wizard πατάμε next και στην σελίδα που εμφανίζεται επιλέγουμε τον τύπο του συνδρομητή. Εμείς σε αυτό το Project θέλουμε να είναι Sql Server Compact. Πατάμε Next και στην επόμενη σελίδα, επιλέγουμε τον υπολογιστή στον οποίο τρέχει ο IIS και επιλέγουμε την εγκατάσταση ενός νέου εικονικού καταλόγου ή τη χρήση ενός ήδη υπάρχοντος . Στην επόμενη σελίδα, Virtual Directory Information, εισάγουμε ένα όνομα στο Alias box (π.χ. NosokomioPC) και επιλέγουμε Next. Ο wizard δημιουργεί έναν πραγματικό φάκελο και μας ρωτά αν πρέπει να τον αντιγράψει στο SQL Server CE Agent και να το καταχωρίσει, πρέπει να απαντήσουμε Yes. Στη σελίδα Secure Communications, ορίζουμε αν πρέπει να χρησιμοποιηθεί ένα ασφαλές κανάλι επικοινωνίας. Αν ναι πρέπει να λάβουμε και να εγκαταστήσουμε ένα πιστοποιητικό Server, όπως επίσης και να ορίσουμε ότι τα πιστοποιητικά πελάτη απαιτούνται για όλους τους πελάτες. Στην συνέχεια επιλέγουμε αν ο IIS πρέπει να δέχεται την ταυτότητα των χρηστών η όχι. Εμείς επιλέγουμε την πρώτη επιλογή Client will connect anonymously. Την επόμενη την αφήνουμε όπως έχει και πατάμε next. Στην σελίδα Snapshot Share Access

πληκτρολογούμε την διαδρομή που έχουμε κάνει share. Πατάμε Finish και έχουμε ολοκληρώσει την διαδικασία. Για να ελέγξουμε αν η διαδικασία έγινε με επιτυχία ανοίγουμε τον Browser και πατάμε \\ το όνομα του υπολογιστή που είναι εγκατεστημένος ο IIS \ το όνομα του φακέλου που δημιουργήσαμε \ sqlcesa35.dll και βλέπουμε αν τρέχει σωστά ο Microsoft SQL Server Compact Server Agent. Για παράδειγμα αν το όνομα του υπολογιστή είναι giorgos, το όνομα του φακέλου nosokomioPC, το path που θα γράψουμε για να κάνουμε τον έλεγχο θα είναι : \\giorgos\nosokomioPC\ sqlcesa35.dll.

7.4 Δημιουργία της εφαρμογής για τις φορητές συσκευές

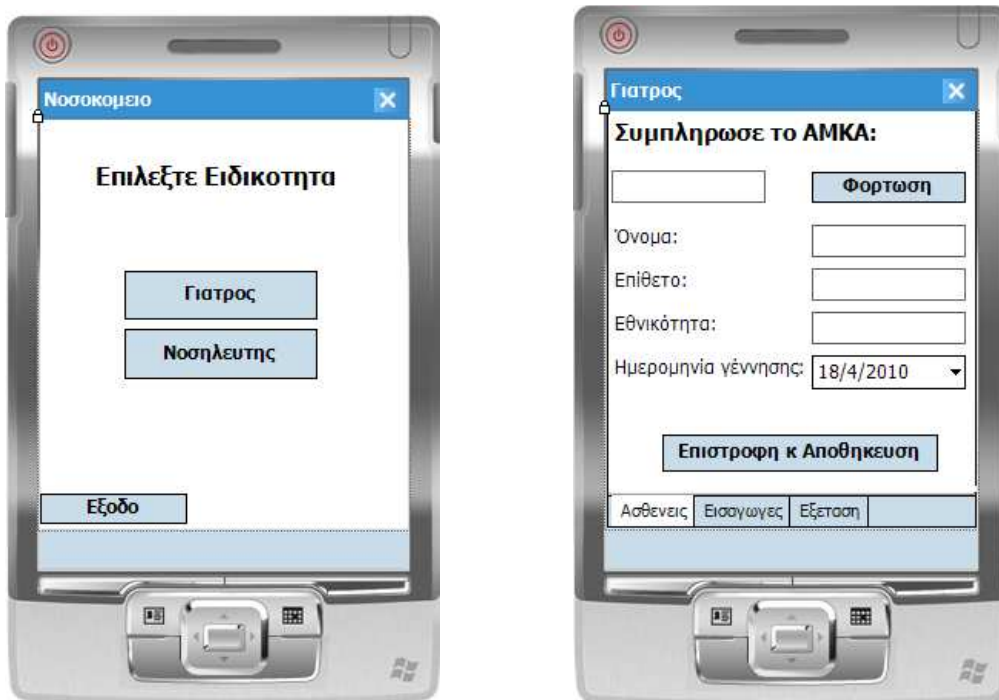
Όπως έχει ειπωθεί, η δημιουργία της εφαρμογής των φορητών συσκευών θα υλοποιηθεί με τη Visual Studio .Net 2005. Ανοίγουμε την .NET και επιλέγουμε New Project όπου στο Project Type επιλέγουμε την γλώσσα προγραμματισμού C# -> Smart Device -> Windows Mobile 5.0 Pocket PC ενώ στο Template επιλέγουμε Device Application. Στην συνέχεια δίνουμε ένα όνομα στην εφαρμογή και πατάμε OK για να συνεχίσουμε. Όπως βλέπουμε στην εικόνα 34 η .NET δημιουργεί μια κενή εφαρμογή όπου εμείς θα δημιουργήσουμε την φόρμα σύμφωνα με τις απαιτήσεις μας.



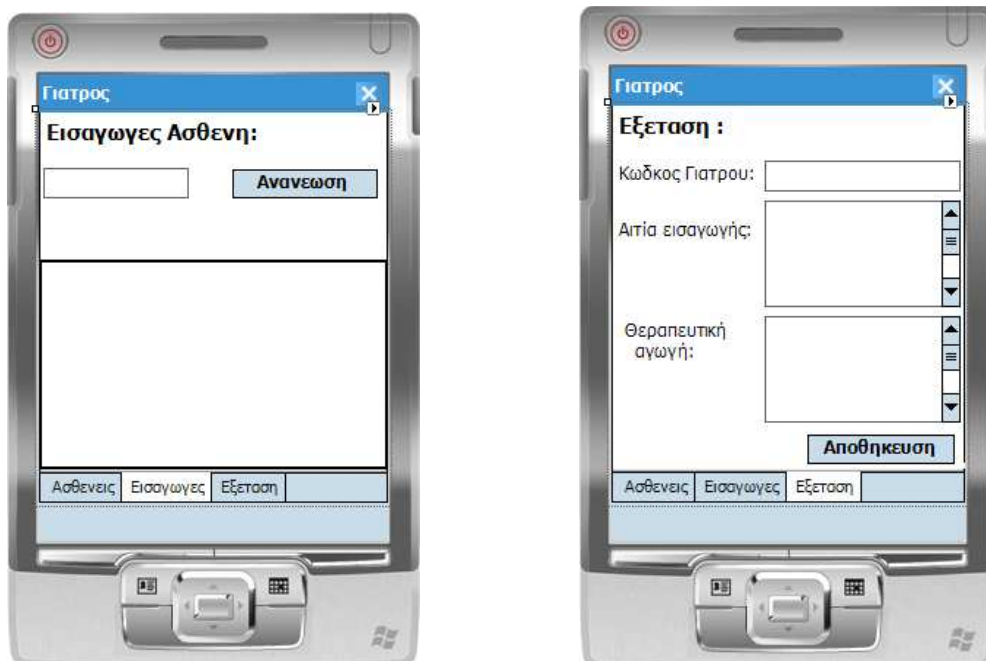
Εικόνα 34 . Κενή φόρμα φορητής συσκευής.

Όπως και στις winForm έτσι και δω από το toolbox με drag and drop προσθέτουμε ό,τι εργαλείο εμείς επιθυμούμε μέσα στην φόρμα όπου έχει ονομαστεί Form1 από την .NET. Παρόλο που δουλεύουμε για φορητή συσκευή στην συνέχεια θα εργαζόμαστε με τον ίδιο ακριβώς τρόπο που δουλεύαμε και για τις winForm.

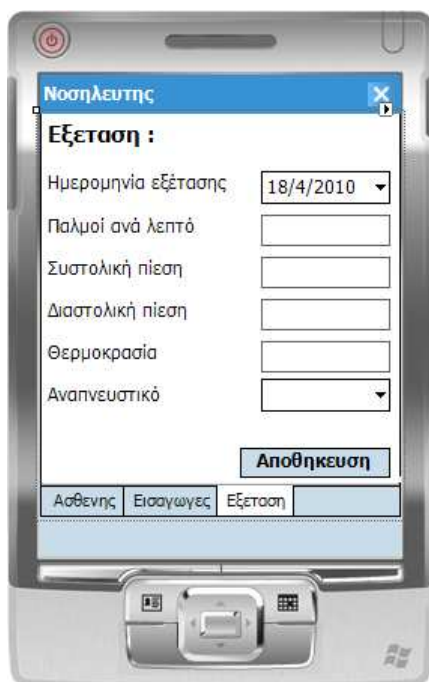
Οι φόρμες που θα δημιουργήσουμε είναι οι παρακάτω :



Εικόνα 35. Αρχική Φόρμα και Φόρμα Ασθενή



Εικόνα 36. Φόρμες Εισαγωγές και Εξέταση



Εικόνα 37. Φόρμα Εξέταση για τον χρήστη Νοσηλεύτης

Της φόρμες Ασθενής και Εισαγωγές τις δημιουργούμε από δυο φορές, μία για τον χρήστη Γιατρό και μια για τον χρήστη Νοσηλεύτης. Δηλαδή εκτελώντας την εφαρμογή η πρώτη φόρμα μας ζητά να επιλέξουμε έναν από τους δυο χρήστες, Γιατρός η Νοσηλεύτης. Όποιον από τους δυο και να διαλέξουμε οι καρτέλες ασθενείς και εισαγωγές είναι ίδιες για τον λόγο αυτό τις δημιουργούμε από δυο φορές. Αφού δημιουργήσουμε τις φόρμες ήρθε η στιγμή να δούμε τον κώδικα αυτών.

7.5 Επεξήγηση του κώδικα των φορητών συσκευών

Ξεκινώντας να γράφουμε τον κώδικα το πρώτο πράγμα που πρέπει να κάνουμε είναι να προσθέσουμε τις βιβλιοθήκες που είναι απαραίτητες έτσι ώστε να μπορεί να γίνεται με τις κατάλληλες εντολές η επικοινωνία της συσκευής με τον Server. Οι βιβλιοθήκες που θα πρέπει να προστεθούν είναι οι παρακάτω :

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

Συνεχίζουμε και παραθέτουμε το μέρος του κώδικα που κάνει τη σύνδεση της φορητής συσκευής με τον Server :

```
try
{
    string sqlServerConnection = @"Provider =SQLOLEDB;Data Source
        =192.168.1.4';Initial Catalog = nosokomio;User Id=sa;Password=123";

    SqlCeRemoteDataAccess myRDA = new SqlCeRemoteDataAccess(
        "http://192.168.1.4/nosokomioPDA/sqlcesa35.dll",
        "DataSource=\\Program Files\\nosokomioPPC\\nosokomio.sdf");
    selectkodastheni = txtamka.Text;

    if (File.Exists(@"\\Program Files\\nosokomioPPC\\nosokomio.sdf"))
        File.Delete(@"\\Program Files\\nosokomioPPC\\nosokomio.sdf");

    SqlCeEngine engine = new SqlCeEngine();
    engine.LocalConnectionString =
        "DataSource=\\Program Files\\nosokomioPPC\\nosokomio.sdf";
    engine.CreateDatabase();

    string strCmd = "Select * From Patient where Κωδικός_ασθενή= " + selectkodastheni;
    myRDA.Pull("Patient",strCmd,sqlServerConnection,RdaTrackOption.TrackingOnWithIndexes,
        "ErrorTable");

    string strCmd2 = @"Select * From InputPatient where Κωδικός_ασθενή="+selectkodastheni;
    myRDA.Pull("InputPatient",strCmd2,sqlServerConnection,RdaTrackOption.TrackingOnWithIndexes,
        "ErrorInputTable");

        myRDA.Dispose();
        FillTextBox();
}
catch (Exception ex)
{
    MessageBox.Show(ex.ToString());
}
}
```

Παίρνοντας τον κώδικα από την αρχή βλέπουμε ότι έχουμε βάλει μέσα σε try τον κώδικα που γίνεται η σύνδεση και η μεταφορά στην φορητή συσκευή των δεδομένων,

ώστε αν κάτι δεν πάει καλά το catch να μας εμφανίσει ένα message box ότι η μεταφορά δεν έχει γίνει σωστά. Έτσι κάνουμε έλεγχο αν γίνετε σωστά η μεταφορά των δεδομένων.

Μετά το try η πρώτη γραμμή είναι το connection string με τον server. Ο provider είναι τύπου OLEDB. Στο Data Source δηλώνουμε το όνομα του υπολογιστή που έχουμε φορτώσει τον SQL Server ή μπορούμε να βάλουμε το IP του, μπορούμε να χρησιμοποιήσουμε και τους δύο τρόπους. Initial Catalog είναι το όνομα της βάσης δεδομένων που έχουμε δημιουργήσει στον SQLServer ενώ στο User Id και στο Password βάζουμε αυτά που μας ζητεί ο SQLServer όταν συνδεόμαστε σαν Server Authentication. Από κάτω δημιουργούμε ένα αντικείμενο SqlCeRemoteDataAccess που το ονομάζουμε myRDA. Μέσα στις παρενθέσεις το πρώτο όρισμα είναι το url του εικονικού καταλόγου που δημιουργήσαμε στην ενότητα 7.3 ενώ δίπλα είναι η διαδρομή που είναι αποθηκευμένη η βάση .sdf στην φορητή συσκευή.

Στην συνέχεια, επειδή ο SQL Server 2005 Compact Edition, δεν μπορεί να εκτελέσει την εντολή Pull αν υπάρχει ήδη ένα table αποθηκευμένο στην ίδια διαδρομή με το ίδιο όνομα, εκτελούμε τις παρακάτω εντολές :

```

if (File.Exists(@"\Program Files\nosokomioPPC\nosokomio.sdf"))
    File.Delete(@"\Program Files\nosokomioPPC\nosokomio.sdf");

SqlCeEngine engine = new SqlCeEngine();
engine.LocalConnectionString = "DataSource=\\Program
                               Files\nosokomioPPC\nosokomio.sdf";
engine.CreateDatabase();
    
```

όπου στο If ελέγχουμε αν υπάρχει ήδη η βάση nosokomio.sdf και αν υπάρχει την διαγράφουμε. Στην συνέχεια ο κώδικας δημιουργεί μια κενή βάση με το όνομα nosokomio.sdf όπου εκεί θα μεταφερθούν τα δεδομένα από τον Server με την εντολή Pull.

Παρακάτω βλέπουμε τον κώδικα που κάνει την μεταφορά των δεδομένων από τον Server στην φορητή συσκευή.

```
string strCmd = "Select * From Patient where Κωδικός_ασθενή= " + selectkodastheni;
myRDA.Pull("Patient",strCmd,sqlServerConnection, RdaTrackOption.TrackingOnWithIndexes,
"ErrorTable");
```

Το strCmd είναι το Query που δηλώνει να παρθούν όλα τα δεδομένα από τον πίνακα Patient όπου ο κωδικός ασθενή είναι ίσος με την τιμή selectkodikosastheni. Αμέσως μετά εκτελείτε η Pull που έχει για ορίσματα τα εξής:

- "Patient" είναι το όνομα του πίνακα που θα τοποθετηθούν τα αρχεία από τον Server.
- strCmd δηλώνουμε ποια και από ποιον πίνακα θα πάρουμε τα δεδομένα από τον server.
- SqlConnection δηλώνει το string που θα συνδέεται με τον Server.
- RdaTrackingOption υποδεικνύει αν το SQLServer CE θα πρέπει να καταγράφει τις αλλαγές που γίνονται στον πίνακα που μεταφέρθηκε (για να σας επιτρέψει να τις επιστρέψετε στο SQLServer). Οι επιλογές είναι TrackingOn, TrackingOff, TrackingWithIndexesOn και TrackingWithIndexesOff.
- "Error Table" είναι το όνομα ενός πίνακα στην τοπική βάση δεδομένων SQL Server CE που χρησιμοποιείται για σφάλματα. Κάθε τοπικός πίνακας που δημιουργούμε θα πρέπει να χρησιμοποιεί ξεχωριστό τοπικό πίνακα σφαλμάτων. Αν προσπαθήσουμε να επαναχρησιμοποιήσουμε ένα όνομα πίνακα σφαλμάτων, η μεταφορά θα αποτύχει.
-

Όταν τελειώνουμε με τα Pull πρέπει πάντα να κάνουμε Dispose() έτσι ώστε να απελευθερώνουμε το RDA ώστε να μπορεί να κάνει το Push στην συνέχεια. Κάτω από το RDA.Dispose(); καλείτε η κλάση FillTextBox():

```

void FillTextBox()
{
    txttonoma.Text = "";
    txtepitheto.Text = "";
    txtethnikotita.Text = "";

    myConn = new SqlConnection("DataSource=\\Program Files
\\nosokomioPPC\\nosokomio.sdf");
    myConn.Open();

    SqlDataAdapter myAdapter = new SqlDataAdapter("Select
Όνομα,Επίθετο,Εθνικότητα,Ημερομηνία_γέννησης From Patient Where
Κωδικός_ασθενή =" + selectkodastheni, myConn);

    DataTable Table = new DataTable("Patient");
    myAdapter.Fill(Table);

    txttonoma.DataBindings.Add("Text", Table, "Όνομα");
    txttonoma.DataBindings.Clear();
    txtepitheto.DataBindings.Add("Text", Table, "Επίθετο");
    txtepitheto.DataBindings.Clear();
    txtethnikotita.DataBindings.Add("Text", Table, "Εθνικότητα");
    txtethnikotita.DataBindings.Clear();

    dateTimePicker2.DataBindings.Add("Value", Table,
"Ημερομηνία_γέννησης");
    dateTimePicker2.DataBindings.Clear();

    myAdapter.Dispose();
    myConn.Close();

    if (txttonoma.Text == "")
        MessageBox.Show("Δεν υπάρχει εγγραφή!Ελεγξτε τον ΑΜΚΑ");
}

```

Ο παραπάνω κώδικας συμπληρώνει τα textbox με τιμές, ανάλογα με το ΑΜΚΑ του ασθενή, στην φόρμα «Ασθενείς» (Εικόνα 32). Συγκεκριμένα οι πρώτες τρεις γραμμές καθαρίζουν τα textbox, όνομα ,επίθετο ,εθνικότητα της φόρμας «ασθενείς» . Στην συνέχεια δημιουργούμαι ένα connection string όπου αυτό θα δείχνει την διαδρομή που είναι αποθηκευμένη η βάση δεδομένων "nosokomio.sdf". Ανοίγουμε την σύνδεση με το connection string και δημιουργούμε ένα data adapter όπου αυτός θα επικοινωνεί με την βάση που κάναμε pull στην φορητή συσκευή. Στον data adapter κάνουμε την επιλογή των στοιχείων που θέλουμε να μεταφέρουμε και δέχεται δυο ορίσματα .Το πρώτο είναι το query με τα στοιχεία που θέλουμε να μεταφερθούν και το δεύτερο είναι το

connection string. Συνεχίζουμε και φτιάχνουμε ένα πίνακα που τον ονομάζουμε "Table" που να είναι ίδιος με τον "Patient" που έχουμε φορτώσει στην συσκευή . Ο κώδικας myAdapter.Fill(Table); γεμίζει το table με τα στοιχεία που έχουμε ορίσει πριν στον adapter. Στις από κάτω σειρές γίνεται το γέμισμα των textbox με τα στοιχεία που επιθυμούμε. Ο κώδικας :

```
txtonoma.DataBindings.Add("Text", Table, "Όνομα");
txtonoma.DataBindings.Clear();
```

δηλώνει ότι στο textbox που το έχουμε ονομάσει txtonoma, όπου στην φόρμα μας θα συμπληρώνεται το όνομα του ασθενή , να προστεθεί με χρήση του "Add" η τιμή του πίνακα "Table" και του πεδίου "Όνομα". Είναι απαραίτητο να εκτελέσουμε την αμέσως επόμενη εντολή γιατί αν θέλουμε να συμπληρώσουμε εκ νέου το textbox με μια νέα τιμή θα δημιουργείται πρόβλημα επειδή το DataBindings θα έχει την προηγούμενη τιμή και θα εμφανίζει αυτή αντί την νέα. Απαραίτητο μετά από κάθε connection open να κάνουμε Dispose τον Adapter για μην καταναλώνει μνήμη.

Τελειώνοντας με αυτό το κομμάτι κώδικα το τελευταίο If ελέγχει αν στο textbox έχει τοποθετηθεί μια τιμή ,αν όχι τότε μας ειδοποιεί η εφαρμογή με ένα Message Box να ελέγξουμε ξανά την τιμή ΑΜΚΑ που δώσαμε.

Θα συνεχίσουμε δείχνοντας και εξηγώντας τον κώδικα της δεύτερης φόρμας "Εισαγωγές" (Εικόνα 33). Ο κώδικας που την συνοδεύει είναι ο παρακάτω:

```
private void btnananeosi_Click(object sender, EventArgs e)
{
    txtamkaisagogi.Text = txtamka.Text;
    selectkodisagogisastheni = txtamkaisagogi.Text;

    myConn = new SqlConnection("DataSource=\\Program
                                Files\\nosokomioPPC\\nosokomio.sdf");
    myConn.Open();
```

```

        SqlConnection myAdapter = new SqlConnection("Select
Κωδικός_εισαγωγής,Κωδικός_ασθενή,Ημέρα_εισαγωγής,Κωδικός_γιατρού,Αιτία_ε
ισαγωγής,Θεραπευτική_αγωγή From InputPatient where Κωδικός_ασθενή = " +
selectkodisagogisasstheni, myConn);

        DataSet myDataSet = new DataSet();
        myAdapter.Fill(myDataSet, "Result");

        dataGrid1.DataSource =
            myDataSet.Tables["Result"].DefaultView;

        myConn.Close();
    }

```

Στην φόρμα αυτή έχουμε προσθέσει ένα textbox ένα Data Grid ένα Button. Ο παραπάνω κώδικας αρχίζει όταν πατηθεί το κουμπί "Ανανέωση". Έτσι έχουμε το textbox αυτής της φόρμας να παίρνει την τιμή από το textbox ΑΜΚΑ της προηγούμενης φόρμας "Ασθενείς". Την ίδια τιμή την δίνουμε και σε μια εξωτερική μεταβλητή τύπου string που έχουμε δηλώσει στη αρχή του κώδικα. Μετά φτιάχνουμε το connection string και ανοίγουμε την σύνδεση. Δημιουργούμε ένα Data Adapter και στην συνέχεια ένα Dataset. Στο Dataset τοποθετούμε τα αποτελέσματα από το Data Adapter. Τελικά για να εμφανίσουμε τα αποτελέσματα στο Data Grid χρησιμοποιούμε την εντολή:

```
dataGrid1.DataSource = myDataSet.Tables["Result"].DefaultView;
```

και τελειώνουμε κλείνοντας τη σύνδεση.

Στην φόρμα "Εξέταση" (Εικόνα 34) θα γράφουμε μέσα στα textbox την αιτία εισαγωγής του ασθενή και ποια θεραπευτική αγωγή του πρότεινε ο γιατρός που τον εξέτασε. Με λίγα λόγια στην βάση που αποθηκεύσαμε με το Pull θα περάσουμε νέα δεδομένα στις εγγραφές εκτός αν υπάρχουν ήδη και απλώς θέλουν συμπλήρωμα. Έτσι επιλέγοντας με την πένα του PDA μια σειρά από το Data Grid περνάμε στα textbox της φόρμας "Εξέταση" αλλά και σε μια εξωτερική μεταβλητή "cellValue0" τις τιμές που

έχουμε επιλέξει, ώστε να τις βλέπουμε καλύτερα και να μπορούμε να τις συμπληρώσουμε. Με τον παρακάτω κώδικα κάνουμε ακριβώς αυτό το πράγμα:

```
private void dataGrid1_Click(object sender, EventArgs e)
{
    cellValue0 =
        dataGrid1[int.Parse(dataGrid1.CurrentRowIndex.ToString
            ()), 0].ToString();

    textBox4.Text =
        dataGrid1[int.Parse(dataGrid1.CurrentRowIndex.ToString
            ()), 3].ToString();
    textBox5.Text =
        dataGrid1[int.Parse(dataGrid1.CurrentRowIndex.ToString
            ()), 4].ToString();
    textBox6.Text =
        dataGrid1[int.Parse(dataGrid1.CurrentRowIndex.ToString
            ()), 5].ToString();
}
```

Ο κώδικας εκτελείται όταν επιλέξουμε από το datagrid μια γραμμή. Έτσι στην μεταβλητή cellValue0 αποθηκεύονται τα δεδομένα της γραμμής που έχουμε επιλέξει και της στήλης 0. Στο textBox4 αποθηκεύετε πάλι η γραμμή που έχουμε επιλέξει με τα περιεχόμενα της στήλης 3 και πάει λέγοντας.

Αφού εμφανιστούν στην φόρμα "Εξέταση" τα στοιχεία που έχουμε επιλέξει από την προηγούμενη φόρμα, μπορούμε αν θέλουμε να διορθώσουμε η να προσθέσουμε νέα στοιχεία. Πατώντας αποθήκευση εκτελείτε ο επόμενος κώδικας:

```
private void btnapothikevsi_Click(object sender, EventArgs e)
{
    if (textBox4.Text != "" & textBox5.Text != "")
    {
        myConn = new SqlConnection("DataSource=\\Program
            Files\\nosokomioPPC\\nosokomio.sdf");
        myConn.Open();

        myCmd = myConn.CreateCommand();
        myCmd.CommandType = CommandType.Text;

        myCmd.CommandText = "UPDATE InputPatient SET
            Αιτία_εισαγωγής='" + textBox5.Text + "',Κωδικός_γιατρού='" +
            textBox4.Text + "',Θεραπευτική_αγωγή='" + textBox6.Text + "'
            WHERE Κωδικός_εισαγωγής =" + cellValue0;
```

```

        myCmd.ExecuteNonQuery();
        myConn.Close();

        textBox4.Text = "";
        textBox5.Text = "";
        textBox6.Text = "";

    }
    else
    {
        MessageBox.Show("Συμπληρώσε Κωδικός Γιατρού και Αιτία
                        εισαγωγής!");
        textBox4.Focus();
    }
}

```

Το If κάνει τον έλεγχο αν ο «Κωδικός Γιατρού» και «Αιτία Εισαγωγής» είναι συμπληρωμένα. Αν όχι εκτελείται το else και εμφανίζεται μήνυμα που μας παροτρύνει να συμπληρώσουμε τα πεδία αυτά και πηγαίνει τον κέρσορα στον «Κωδικό Γιατρού». Αν η συνάρτηση του If είναι αληθής συνεχίζει ο κώδικας που βρίσκετε μέσα σε αυτό. Συνδεόμαστε με την βάση "nosokomio.sdf" και δημιουργούμε ένα Command με το όνομα myCmd. Στην συνέχεια δηλώνουμε ότι το myCmd θα είναι τύπου Text και γράφουμε το query που θα κάνει update την βάση, με τα δεδομένα που έχουμε συμπληρώσει στην φόρμα. Συγκεκριμένα του λέμε να κάνει update το table "InputPaien" και να βάλει στα πεδία «Αιτία Εισαγωγής», «Κωδικός Γιατρού», «θεραπευτική Αγωγή» τις τιμές που έχει το textbox5, textbox4, textbox6 ,όπου ο «Κωδικός Εισαγωγής» είναι ίσος με την μεταβλητή CellValue0. Με το myCmd.ExecuteNonQuery(); γίνεται η μεταφορά στο πίνακα. Στην συνέχεια κλείνουμε την σύνδεση με την βάση και παρακάτω καθαρίζουμε τα textbox από τις τιμές που είχαμε δώσει πριν.

Μέχρι στιγμή έχουμε δει το πώς συνδέσαμε την φορητή συσκευή με την κεντρική βάση δεδομένων του Server. Στην συνέχεια πως από την κεντρική βάση δεδομένων μεταφέραμε τους πίνακες με τα δεδομένα που εμείς επιθυμούσαμε και τα αποθηκεύσαμε

στην φορητή συσκευή. Ακόμα πως εμφανίστηκαν τα δεδομένα στις φόρμες όπου κάναμε αλλαγές σε αυτά και αποθηκεύσαμε τις αλλαγές στην βάση που έχουμε μεταφέρει στην φορητή συσκευή. Ενώ παρακάτω θα δούμε πως πατώντας το κουμπί "Επιστροφή κ Αποθήκευση" θα μεταφερθούμε στην αρχική φόρμα και θα ενημερώσουμε την βάση δεδομένων του server με τις αλλαγές που κάναμε. Ενώ κάνουμε Pull για να φέρουμε δεδομένα από τον server τώρα θα κάνουμε Push και θα στείλουμε τα νέα δεδομένα πίσω σε αυτόν. Ο κώδικας μας δείχνει το πώς :

```
private void button1_Click(object sender, EventArgs e)
{
    this.Close();

    string sqlServerConnection = @"Provider =SQLOLEDB;DataSource
='192.168.1.4';Initial Catalog = nosokomio;User Id=sa;Password=123";

    SqlCeRemoteDataAccess myRDA =
newSqlCeRemoteDataAccess ("http://192.168.1.4/nosokomioPDA/sqlcesa35.dll"
, "DataSource=\\Program Files\\nosokomioPPC\\nosokomio.sdf");

    myRDA.Push("InputPatient", sqlServerConnection,
RdaBatchOption.BatchingOff);

    myRDA.Dispose();
    MessageBox.Show("Τα Δεδομενα Αποθηκευτηκαν στον Server");
}
```

Όπως βλέπουμε ο κώδικας εκτελείται όταν πατάμε το κουμπί "Επιστροφή κ Αποθήκευση" με το this.close(); κλείνει η φόρμα που είναι ανοιχτή και επιστρέφουμε στην κεντρική φόρμα. Όλος ο παρακάτω κώδικας είναι ίδιος με το Pull εκτός από τη γραμμή :

```
myRDA.Push("InputPatient",
sqlServerConnection,RdaBatchOption.BatchingOff);
```

όπου εκεί :

- "InputPatient" Το όνομα του τοπικού πίνακα που περιέχει τις καταγεγραμμένες αλλαγές.
- SqlConnection δηλώνει το string που θα συνδέεται με τον Server.
- RdaBatchOption η παράμετρος αυτή καθορίζει αν το SQL Server CE θα πρέπει να εφαρμόζει κάθε ενημέρωση σε ξεχωριστό transaction (RdaBatchingOff) ή όλες τις ενημερώσεις μαζί σε ένα και μόνο transaction (RdaBatchingOn).

Αν στην αρχική φόρμα επιλέξουμε την εφαρμογή για τον νοσηλευτή τότε στο tab control οι δυο από τις τρεις φόρμες είναι ίδιες εκτός από την φόρμα "Εξέταση" όπου εκεί ο νοσηλευτής αποθηκεύει στοιχεία της εξέτασης που έκανε στον ασθενή, όπως θερμοκρασία, πίεση κτλ (Εικόνα 34). Ακόμα το μεγαλύτερο μέρος του κώδικα είναι ίδιο με αυτό που έχουμε εξηγήσει πιο πάνω. Ο κώδικας που θα εξηγήσουμε στην συνέχεια είναι διαφορετικός γιατί ενώ πριν κάναμε Update σε μια εγγραφή ενός πίνακα τώρα θα πρέπει να δημιουργήσουμε καινούρια εγγραφή. Για τον λόγο αυτό υπάρχουν οι παρακάτω αλλαγές. Αρχίζουμε όπως και πριν κάνοντας Pull τα δεδομένα που χρειαζόμαστε από το Server και τα μεταφορτώνουμε τοπικά στην φορητή συσκευή . Στο τέλος της μεταφοράς αυτής εκτελούμε την κλάση LastID();

```
private void LastID()
{
    SqlConnection myConn = new SqlConnection("DataSource=
        \\ProgramFiles\\nosokomioPPC\\nosokomio.sdf");
    myConn.Open();

    SqlCeDataAdapter da =
        new SqlCeDataAdapter("SELECT * FROM WeekdayChek1", myConn);

    DataTable dt = new DataTable();
    da.Fill(dt);
}
```



```

for (int i = 0; i < dt.Rows.Count; i++)
    {
        m = Int32.Parse(dt.Rows[i][0].ToString());
    }
m++;
myConn.Close();
}

```

Αυτό που κάνει η παραπάνω κλάση είναι να κρατάει σε μια εξωτερική μεταβλητή την τιμή της τελευταίας εγγραφής της στήλης «Κωδικός_Εξέτασης» έτσι ώστε να χρησιμοποιηθεί στην κλάση AlterWeekdayChek().

Η κλάση αρχικά συνδέετε με την βάση "nosokomio.sdf" και δημιουργεί ένα Data Adapter που το ονομάζουμε da. Στην συνέχεια δημιουργούμε ένα DataTable και το συμπληρώνουμε με τα δεδομένα από το da. Μέσα στο for ο μετρητής ελέγχει πόσες γραμμές έχει ο πίνακας και πηγαίνει στη τελευταία γραμμή, βλέπει τι τιμή έχει και την αποθηκεύει στην μεταβλητή "m". Στην μεταβλητή προσθέτουμε συν ένα και κλείνουμε την σύνδεσή μας με την βάση.

Αφού εκτελέσουμε την κλάση συνεχίζουμε στην επόμενη κλάση όπου αυτή θα δημιουργήσει μια νέα σειρά στον πίνακα WeekdayChek με τιμή στο πεδίο «Κωδικός_Εξέτασης» την τιμή της μεταβλητής "m". Η κλάση αφού συνδεθεί και αυτή με την τοπική βάση "nosokomio.sdf" δημιουργεί μια νέα γραμμή στον πίνακα WeekdayChek με τις παρακάτω εντολές :

```

myCmd = myConn.CreateCommand();
myCmd.CommandText = @"ALTER TABLE WeekdayChek ALTER
                        COLUMN Κωδικός_εξέτασης IDENTITY (" + m + " ,1)";
myCmd.ExecuteNonQuery();

```

Η πρώτη γραμμή δημιουργεί ένα Command και το ονομάζει myCmd. Το Command Text λέει ότι στον υπάρχον πίνακα WeekdayChek και στην στήλη «Κωδικός Εξέτασης» να προστεθεί μια νέα τιμή "m", ενώ κάθε φορά που θα εκτελείται το myCmd θα

προστίθεται συν ένα στην τιμή "m". Με το ExecuteNonQuery εκτελείται το myCmd. Μέχρι στιγμής αυτό που έχουμε κάνει είναι να παίρνουμε πίνακες από τον Server και να καλούμε τις κλάσεις να δημιουργούν στον πίνακα WeekdayChek καινούριες γραμμές, όπου είναι συμπληρωμένο μόνο το πεδίο «Κωδικός Εξέτασης». Τα άλλα πεδία προς το παρόν θα είναι κενά αλλά θα συμπληρώνονται από την φόρμα «Εξέταση» (Εικόνα 34). Η φόρμα αποτελείται από textbox που εκεί ο νοσηλευτής θα συμπληρώνει τις μετρήσεις που πήρε όταν εξέτασε τον ασθενή και ένα Button που θα κάνει την αποθήκευση στην τοπική βάση της φορητής συσκευής. Αφού συμπληρώσει ο χρήστης τα στοιχεία και πατήσει το κουμπί για την αποθήκευση θα εκτελεστεί ο παρακάτω κώδικας:

```

if (textBox1.Text!="" & textBox2.Text !="" & textBox3.Text !="" &
    textBox4.Text !="" & comboBox1.Text != "")
{
    myConn = new SqlConnection("DataSource=\\Program Files
        \\nosokomioPPC\\nosokomio.sdf");
    myConn.Open();
    myCmd = myConn.CreateCommand();

    myCmd.CommandText = @"INSERT INTO
WeekdayChek (Κωδικός_εισαγωγής,Κωδικός_ασθενή,Κωδικός_γιατρού,Ημερομηνία_
εξέτασης,Παλμοί_ανά_λεπτό,Συστολική_πίεση,Διαστολική_πίεση,Θερμοκρασία,Α
ναπνευστικό)
VALUES
(@Κωδικός_εισαγωγής,@Κωδικός_ασθενή,@Κωδικός_γιατρού,@Ημερομηνία_εξέταση
ς,@Παλμοί_ανά_λεπτό,@Συστολική_πίεση,@Διαστολική_πίεση,@Θερμοκρασία,@Ανα
πνευστικό)";

    myCmd.Parameters.Add("@Κωδικός_εισαγωγής", SqlDbType.NChar, 10);
    myCmd.Parameters.Add("@Κωδικός_ασθενή", SqlDbType.NChar, 10);
    myCmd.Parameters.Add("@Κωδικός_γιατρού", SqlDbType.Int);
    myCmd.Parameters.Add("@Ημερομηνία_εξέτασης", SqlDbType.DateTime);
    myCmd.Parameters.Add("@Παλμοί_ανά_λεπτό", SqlDbType.NChar, 5);
    myCmd.Parameters.Add("@Συστολική_πίεση", SqlDbType.NChar, 10);

    myCmd.Parameters.Add("@Διαστολική_πίεση", SqlDbType.NChar, 5);
    myCmd.Parameters.Add("@Θερμοκρασία", SqlDbType.NChar, 5);
    myCmd.Parameters.Add("@Αναπνευστικό", SqlDbType.NChar, 10);

```

```

myCmd.Parameters["@Κωδικός_εισαγωγής"].Value = cellValue0;
myCmd.Parameters["@Κωδικός_ασθενή"].Value = txtamkaisagogi.Text;
myCmd.Parameters["@Κωδικός_γιατρού"].Value = cellValue3;
myCmd.Parameters["@Ημερομηνία_εξέτασης"].Value =
    dateTimePicker1.Value;
myCmd.Parameters["@Παλμοί_ανά_λεπτό"].Value = textBox1.Text;
myCmd.Parameters["@Συστολική_πίεση"].Value = textBox2.Text;
myCmd.Parameters["@Διαστολική_πίεση"].Value = textBox3.Text;
myCmd.Parameters["@Θερμοκρασία"].Value = textBox4.Text;
myCmd.Parameters["@Αναπνευστικό"].Value = comboBox1.Text;

myCmd.ExecuteNonQuery();
myConn.Close();

textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
textBox4.Text = "";
comboBox1.Text = null;
}
else
{
    MessageBox.Show("Συμπληρωσε ολα τα πεδια!");
    textBox1.Focus();
}

```

Στο if γίνεται ο έλεγχος αν είναι όλα τα πεδία τις φόρμας συμπληρωμένα γιατί αν κάποιο από αυτά είναι κενό θα εμφανίζετε messageBox με μήνυμα που θα προτρέπει τον χρήστη να συμπληρώσει όλα τα πεδία. Αφού κάνουμε την σύνδεση με την βάση δημιουργούμε ένα Command. Στο command Text δηλώνουμε να γίνει εισαγωγή στον πίνακα WeekdayChek στις στήλες «Κωδικός Εισαγωγής» , «Κωδικός ασθενή» κτλ των τιμών από τις μεταβλητές «@Κωδικός Εισαγωγής» , «@Κωδικός ασθενή» κτλ. Όταν δημιουργήσαμε μια νέα σειρά στον πίνακα WeekdayChek τα κελιά αυτά τα είχαμε αφήσει κενά. Τώρα θα πρέπει αρχικά να δηλώσουμε τι τύπου δεδομένα θα αποθηκευτούν μέσα στα κελιά και έπειτα να αντιστοιχήσουμε την μεταβλητή με τα δεδομένα. Έτσι :

```

myCmd.Parameters.Add("@Κωδικός_εισαγωγής", SqlDbType.NChar, 10);
myCmd.Parameters["@Κωδικός_εισαγωγής"].Value = cellValue0;

```

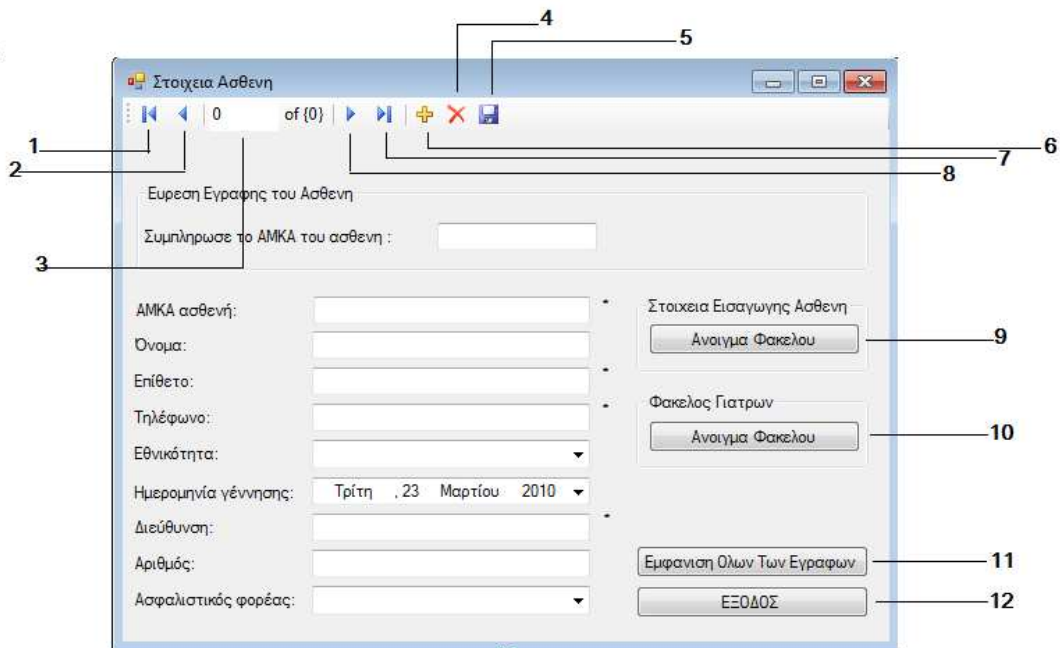
στην πρώτη γραμμή δηλώνουμε στο command που δημιουργήσαμε πριν, ότι η μεταβλητή «@Κωδικός_εισαγωγής» ,θα είναι τυπου NChar μεγεθους 10 bit. Αφού δηλώσουμε όλες τις μεταβλητές η δεύτερη γραμμή λέει στην παρέμετρος «@Κωδικός_εισαγωγής» να γίνει ίση με την τιμή τις μεταβλητής cellValue0. Όταν εκτελεστεί η εντολή ExecuteNonQuery τα δεδομένα θα έχουν αποθηκευτεί στο πίνακα WeekdayChek όπου αργότερα τα νέα δεδομένα θα επιστρέψουν πίσω στον Server.

Τελειώνοντας με την εξήγηση του κώδικα έχουμε υλοποιήσει ένα δίκτυο δεδομένων, ασύρματο και ενσύρματο , όπου σε αυτό γίνεται ανταλλαγή και αποθήκευση δεδομένων ανάμεσα στον Server, τους σταθερούς υπολογιστές και τις φορητές συσκευές.

8.0 Εγχειρίδιο χρήσης

8.1 Βασικά κουμπιά της εφαρμογής των σταθερών υπολογιστών

Τα βασικά κουμπιά της εφαρμογής φαίνονται στην επόμενη εικόνα.



Εικόνα 38. Αρίθμηση βασικών κουμπιών

- 1 Πατώντας το κουμπί πηγαίνουμε στην αρχική εγγραφή.
- 2 Πηγαίνουμε στη αμέσως προηγούμενη εγγραφή από αυτήν που βρισκόμαστε.
- 3 Το πεδίο αυτό μας δείχνει κατά σειρά σε πια εγγραφή βρισκόμαστε από το σύνολο των εγγραφών.
- 4 Διαγράφουμε μια εγγραφή.
- 5 Γίνετε αποθήκευση τις καινούριες εγγραφής ή αποθηκεύουμε τυχών αλλαγές που κάναμε σε μια εγγραφή.
- 6 Δημιουργούμε μια καινούρια εγγραφή.
- 7 Πηγαίνουμε στην τελευταία εγγραφή.
- 8 Πηγαίνουμε στην αμέσως επόμενη εγγραφή από αυτήν που βρισκόμαστε.
- 9 Ανοίγει την φόρμα "Στοιχεία Εισαγωγής ασθενή".

- 10 Ανοίγει την φόρμα "Φάκελος Γιατρών"
- 11 Όταν γίνετε μια αναζήτηση εγγραφής τότε μεταβαίνουμε σε αυτήν και δεν μπορούμε με τα βελάκια να πάμε στην επόμενη ,αρχική ,προηγούμενη ή στο τέλος. Πατώντας το κουμπι αυτό μας εμφανίζει όλες τις εγγραφές.
- 12 Κλείνουμε την εφαρμογή.





8.2 Επεξήγηση της λειτουργίας της εφαρμογής των σταθερών υπολογιστών για την γραμματεία.

Σε αυτό το κεφάλαιο θα δείξουμε στον χρήστη πώς λειτουργεί αυτή η εφαρμογή. Αρχικά αφού θα έχουμε εγκαταστήσει την εφαρμογή στον σταθερό υπολογιστή της γραμματείας, η γραμματεία θα είναι αυτή που θα κάνει την εισαγωγή των βασικών στοιχείων του ασθενή στον ηλεκτρονικό φάκελο του.

Εικόνα 39. Αρχική φόρμα εισαγωγής βασικών στοιχείων τους ασθενή στον ηλεκτρονικό φάκελο του.



Όταν έρχεται ο ασθενής πηγαίνει από την γραμματεία και δίνει τα στοιχεία του. Η γραμματεία εισάγει τον ΑΜΚΑ του ασθενή στο πεδίο "Συμπληρώστε το ΑΜΚΑ του ασθενή" και πατάει Enter. Αν στα πεδία από κάτω εμφανιστούν τα στοιχεία του σημαίνει ότι έχει εισαχτεί ξανά και τα στοιχεία του υπάρχουν ήδη στην βάση δεδομένων του νοσοκομείου. Αν ισχύει αυτό μεταβαίνει στη φόρμα 'Στοιχεία εισαγωγής ασθενή' πατώντας το κουμπί 'Άνοιγμα Φακέλου' που βρίσκεται από κάτω.



Εικόνα 40. Φόρμα στοιχεία εισαγωγής ασθενή.

Στην συνέχεια πατάει στο navigation bar το  για να δημιουργήσει μια καινούρια εγγραφή και μετά το  για να αποθηκευτεί η νέα εγγραφή στην βάση. Εάν ο ασθενής δεν έχει εισαχτεί άλλη φορά τότε εμφανίζετε ένα message box που τον ειδοποιεί ότι δεν υπάρχει εγγραφή. Η γραμματεία πρέπει να πατήσει  ώστε να δημιουργηθεί μια νέα εγγραφή και συμπληρώνει τα στοιχεία του. Μόλις τα συμπληρώσει πατάει  για να γίνει η αποθήκευση και συνεχίζει την ίδια διαδικασία με πριν. Τα απαραίτητα πεδία που πρέπει

να συμπληρωθούν είναι αυτά με το αστεράκι δίπλα. Σε διαφορετική περίπτωση εμφανίζεται ένα message box που παροτρύνει τον χρήστη να συμπληρώσει τα σωστά στοιχεία. Για την εισαγωγή του ασθενή η δουλειά της γραμματείας τελειώνει εκεί. Ακόμα η γραμματεία μπορεί να βλέπει τα στοιχεία των γιατρών και να προσθέτει νέα αν χρειαστεί.

Εικόνα 41. Φόρμα "Στοιχεία Γιατρών".



Αυτό γίνεται ως εξής , από την κεντρική φόρμα πατάμε να ανοίξει ο "Φάκελος των Γιατρών". Θα ανοίξει η φόρμα "Στοιχεία Ασθενή" όπου σε αυτή μπορούμε να δούμε όλα τα στοιχεία των γιατρών που εργάζονται στο νοσοκομείο. Σε αυτήν την φόρμα μπορούμε να συμπληρώσουμε τον κωδικό του γιατρού αν τον ξέρουμε και να μας εμφανίσει πατώντας enter τα στοιχεία του ή από το ComboBox εμφανίζοντας ανά ειδικότητα όλους τους εγγεγραμμένους . Εκτός από αυτό μπορούμε να προσθέσουμε μια νέα εγγραφή πατώντας  και αφού συμπληρώσουμε τα στοιχεία πατάμε  για να γίνει η αποθήκευση στην βάση. Ακόμα μπορούμε να διαγράψουμε την εγγραφή ενός γιατρού

που τυχόν έχει φύγει από το νοσοκομείο. Εφόσον βρούμε την εγγραφή που θέλουμε να διαγράψουμε πατάμε  και στην συνέχεια  για να αποθηκευτεί η αλλαγή που κάναμε στην βάση.

8.3 Επεξήγηση της λειτουργίας της εφαρμογής των σταθερών υπολογιστών για τους γιατρούς.

Όπως έχουμε επισημάνει και σε προηγούμενο κεφάλαιο, σε όλα τα γραφεία των γιατρών θα υπάρχει από ένας υπολογιστής όπου θα υπάρχει εγκατεστημένη η εφαρμογή που δημιουργήσαμε. Εφόσον έχουν περάσει οι ασθενείς από την γραμματεία ο γιατρός θα τους δέχεται και θα μπορεί από την εφαρμογή στον υπολογιστή του να βλέπει τα στοιχεία του ασθενή που πριν λίγο αποθηκεύτηκαν στην βάση. Η κεντρική φόρμα είναι ίδια με την αρχική φόρμα της γραμματείας (εικόνα 39). Αφού ο γιατρός βρει την εγγραφή του ασθενή από το ΑΜΚΑ, ανοίγει την φόρμα “Στοιχεία εισαγωγής ασθενή ” (εικόνα 40). Επιλέγει από το πεδίο group Box “Ημερομηνία που έχει επανεισαχτεί” την ημερομηνία που επιθυμεί, εφόσον ο ασθενής έχει επανεισαχτεί οι ημερομηνίες θα είναι περισσότερες από μια και λογικά θα επιλέξει την τελευταία που συμπίπτει με την ημερομηνία εκείνης της μέρας. Στην συνέχεια συμπληρώνει τα στοιχεία όπως την αιτία εισαγωγής του ασθενή, τον κωδικό του και τη θεραπευτική αγωγή που πρότεινε στον ασθενή. Εάν θέλει μπορεί να ανοίξει την φόρμα “Παρακολούθηση Ασθενή” και να συμπληρώσει περισσότερα στοιχεία.

Εικόνα 42. Φόρμα καθημερινή εξέταση ασθενή.

Σε αυτή την φόρμα ο γιατρός αποθηκεύει πιο λεπτομερή στοιχεία της εξέτασης όπως είναι οι καρδιακοί παλμοί του ασθενή και η θερμοκρασία του. Για να προσθέσει τα καινούρια δεδομένα ακολουθεί την συνηθισμένη διαδικασία, πατάει το κουμπί  από το navigation bar συμπληρώνει με τα νέα δεδομένα τα κατάλληλα πεδία και για να τα αποθηκεύσει πατάει . Όταν αποθηκευτούν τα δεδομένα στην κεντρική βάση θα εμφανιστεί ένα message box ότι τα δεδομένα αποθηκεύτηκαν με επιτυχία. Στα επόμενα κεφάλαια θα εξηγήσουμε πως λειτουργεί η εφαρμογή που έχουμε δημιουργήσει για τις φορητές συσκευές.

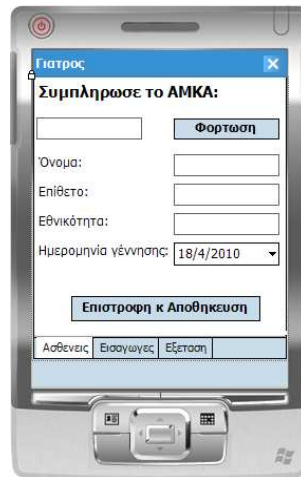
8.4 Επεξήγηση της λειτουργίας της εφαρμογής των φορητών υπολογιστών τσέπης για τους γιατρούς.

Ανοίγοντας την εφαρμογή που έχουμε δημιουργήσει για τους υπολογιστές τσέπης βλέπουμε στην αρχική φόρμα ότι μας δίνεται η επιλογή να επιλέξουμε αν θα χρησιμοποιήσουμε την εφαρμογή για τους γιατρούς, τους νοσηλευτές ή με το κουμπί έξοδος να κλείσουμε την εφαρμογή (εικόνα 43).



Εικόνα 43. Φόρμα εισόδου του PPC.

Σε αυτήν την περίπτωση επιλέγουμε και πατάμε το κουμπί "Γιατρός" και αμέσως εμφανίζεται η φόρμα που από εκεί θα δούμε τα στοιχεία του ασθενή (εικόνα 44).



Εικόνα 44. Φόρμα «Ασθενείς» της εφαρμογής για τον γιατρό.

Στο πάνω αριστερό textbox ο χρήστης συμπληρώνει το ΑΜΚΑ του ασθενή που θέλει να εξετάσει και πατάει το κουμπί "Φόρτωση" όπου πατώντας το θα εμφανιστούν τα στοιχεία του ασθενή. Μόλις εμφανιστούν επιλέγουμε να ανοίξουμε την φόρμα «Εισαγωγή» και το κάνουμε πατώντας το tab Page "Εισαγωγές".



Εικόνα 45. Φόρμα «Εισαγωγές» της εφαρμογής για τον γιατρό.

Στην συνέχεια πατάμε το κουμπί ανανέωση και εμφανίζετε ένας πίνακας με τα δεδομένα του ασθενή όπως η ημέρα που έγινε η εισαγωγή του ,τον λόγο της εισαγωγής και άλλα. Αφού επιλέξουμε μια γραμμή από τον πίνακα, με την γραφίδα του PPC, ανάλογα με την ημερομηνία που θέλουμε, πατάμε στο tab Page για να δούμε τα στοιχεία της φόρμας "Εξέταση" (εικόνα 46).

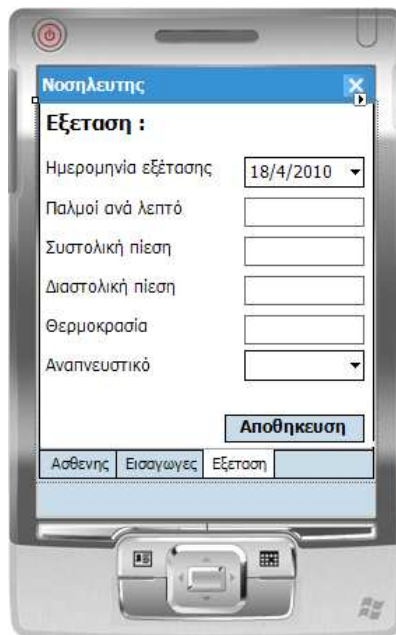


Εικόνα 46. Φόρμα «εξέταση» της εφαρμογής για τον γιατρό.

Σε αυτήν την φόρμα συμπληρώνει ο γιατρός την Αιτία Εισαγωγής και την θεραπευτική αγωγή που πρότείνει στον ασθενή και στην συνέχεια πατάει το κουμπί αποθήκευση. Εφόσον ο γιατρός έχει συμπληρώσει όλα τα στοιχεία για να αποθηκευτούν στον Server πηγαίνουμε πάλι στην φόρμα "Ασθενείς" και πατάμε το κουμπί "Επιστροφή και Αποθήκευση". Αν τα δεδομένα αποθηκευτούν με επιτυχία εμφανίζετε μήνυμα που μας ειδοποιεί ότι τα δεδομένα έχουν αποθηκευτεί με επιτυχία και επιστρέφουμε στην αρχική φόρμα.

8.5 Επεξήγηση της λειτουργίας της εφαρμογής των φορητών υπολογιστών τσέπης για τους νοσηλευτές.

Ανοίγοντας την εφαρμογή βλέπουμε την κεντρική φόρμα (εικόνα 43). Αυτήν την φορά θα επιλέξουμε να ανοίξουμε την εφαρμογή για τους νοσηλευτές, έτσι θα πατήσουμε το κουμπί που γράφει «Νοσηλευτής». Στην συνέχεια όπως και στην φόρμα για τους γιατρούς έτσι και τώρα η πρώτη φόρμα είναι ίδια όπως και στη εικόνα 44. Βάζουμε πάλι το ΑΜΚΑ του ασθενή που θέλουμε να εξετάσουμε και πατάμε το κουμπί που γράφει Φόρτωση. Εφόσον εμφανιστούν τα στοιχεία του ασθενή επιλέγουμε την tab page «Εισαγωγές» και βλέπουμε την φόρμα (εικόνα 45). Εκεί πατάμε το κουμπί ανανέωση και μας εμφανίζονται τα στοιχεία του ασθενή σε έναν πίνακα. Αφού επιλέξουμε μια γραμμή του πίνακα με την ημερομηνία που επιθυμούμε, συνήθως διαλέγουμε την τελευταία, και πατάμε την tab page Εξέταση για να μας εμφανιστεί η φόρμα Εξετάσεις (εικόνα 46).



Εικόνα 47. Φόρμα εξέταση της εφαρμογής για τον νοσηλευτή.

Σε αυτήν πάλι αφού συμπληρώσουμε τα στοιχεία της εξέτασης πατάμε το κουμπί αποθήκευση για να αποθηκευτούν προσωρινά. Για να μεταφερθούν όλα τα δεδομένα στην βάση του Server πρέπει να γυρίσουμε στην φόρμα «Ασθενείς» και να πατήσουμε το κουμπί που γράφει Επιστροφή και Αποθήκευση. Αν εμφανιστεί μήνυμα που να γράφει ότι τα δεδομένα αποθηκεύτηκαν στον Server με επιτυχία τότε η μεταφορά και η αποθήκευση των δεδομένων ήταν επιτυχής.

* Είναι πολύ σημαντικό η συσκευή που τρέχει την εφαρμογή να μην είναι εκτός του δικτύου. Δηλ πρέπει να υπάρχει ασύρματη επικοινωνία με τον Router.

9.0 Developer Guide

9.1 Εφαρμογή σταθερού υπολογιστή

Το κεφάλαιο αυτό είναι ένας οδηγός για τους προγραμματιστές, ώστε να δουν με ποιον προγραμματιστικό τρόπο έγιναν συγκεκριμένες ενέργειες.

Αρχικά οι βιβλιοθήκες που χρησιμοποιούμε για την υλοποίηση της εφαρμογής των σταθερών υπολογιστών είναι οι παρακάτω.

- System
- System.Collections.Generic
- System.ComponentModel
- System.Data
- System.Drawing
- System.Text
- System.Windows.Forms

- Η αποθήκευση των δεδομένων στον πίνακα patient, από την φόρμα "Στοιχεία Ασθενή" γίνεται με τον κώδικα :

```
private void patientBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    this.Validate ();
    this.patientBindingSource.EndEdit ();
    this.patientTableAdapter.Update (this.nosokomioDataSet.Patient);
}
```

Η αποθήκευση των δεδομένων στον πίνακα doctor, από την φόρμα "Στοιχεία Γιατρών γίνεται" με τον κώδικα :

```
private void doctorBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    this.Validate();
    this.doctorBindingSource.EndEdit();
    this.doctorTableAdapter.Update(this.nosokomioDataSet.Doctor);
}
```

Η αποθήκευση των δεδομένων στον πίνακα inputpatient, από την φόρμα "Στοιχεία Εισαγωγής Ασθενή" γίνεται με τον κώδικα :

```
private void inputPatientBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    this.Validate();
    this.inputPatientBindingSource.EndEdit();
    this.inputPatientTableAdapter.Update(this.nosokomioDataSet.InputPatient);
}
```

Η αποθήκευση των δεδομένων στον πίνακα weekdaychek, από την φόρμα "Καθημερινή Εξέταση Ασθενή" γίνεται με τον κώδικα :

```
private void weekdayChekBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    this.Validate();
    this.weekdayChekBindingSource.EndEdit();
    this.weekdayChekTableAdapter.Update(this.nosokomioDataSet.WeekdayChek);
}
```


- Η φόρτωση των δεδομένων από την βάση, στα textbox της φόρμας "Στοιχεία Ασθενή" γίνεται με τον κώδικα :

```
private void Form1_Load(object sender, EventArgs e)
{
    this.patientTableAdapter.Fill(this.nosokomioDataSet.Patient);
}
```

Ο παρακάτω κώδικας εμφανίζει στα textbox της φόρμας "Στοιχεία Ασθενή" τα δεδομένα που επιθυμούμε σύμφωνα με το ΑΜΚΑ του ασθενή. Τα δεδομένα εμφανίζονται αφού συμπληρώσουμε το ΑΜΚΑ στο κατάλληλο textbox της φόρμας και πατήσουμε το Enter του πληκτρολογίου μας :

```
private void txtEvresiAstheni_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        e.SuppressKeyPress = true;
        patientTableAdapter.FillByAMKA(nosokomioDataSet.Patient, txtEvresiAstheni.Text);
    }
}
```

Η εμφάνιση των στοιχείων του γιατρού στην φόρμα "Στοιχεία Γιατρού" ανάλογα με τον κωδικό του γίνεται με τον κώδικα :

```
private void Form2_Load(object sender, EventArgs e)
{
    int a = int.Parse("0" + this.κωδικός_γιατρούTextBox.Text);
    this.doctorTableAdapter.FillBykod(this.nosokomioDataSet.Doctor, a);
}
```

Ο παρακάτω κώδικας εμφανίζει στα textbox της φόρμας " Στοιχεία Γιατρών " τα δεδομένα που επιθυμούμε σύμφωνα με τον κωδικό του γιατρού. Τα δεδομένα εμφανίζονται αφού συμπληρώσουμε τον κωδικό στο κατάλληλο textbox της φόρμας και πατήσουμε το Enter του πληκτρολογίου μας :

```
private void textBox1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        e.SuppressKeyPress = true;
        int a = int.Parse("0" + this.textBox1.Text);
    }
}
```

```

        this.doctorTableAdapter.FillBykod(this.nosokomioDataSet.Doctor, a);
    }
}

```

Η εμφάνιση των στοιχείων του γιατρού στην φόρμα "Στοιχεία Γιατρού" ανάλογα με την ειδικότητά του γίνεται με τον κώδικα:

```

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    this.doctorTableAdapter.FillByidikotita(nosokomioDataSet.Doctor, comboBox1.Text);
}

```

Η εμφάνιση δεδομένων του ασθενή στην φόρμα "Στοιχεία Εισαγωγής Ασθενή" ανάλογα με τον κωδικό του γίνεται με τον κώδικα :

```

private void Form4_Load(object sender, EventArgs e)
{
    this.κωδικός_ασθενήTextBox.Text((Form1)this.form1inForm4).κωδικός_ασθενήTextBox.Text;
    this.inputPatientTableAdapter.FillBykodastheni(this.nosokomioDataSet.InputPatient,κωδικός_ασθε
νήTextBox.Text);
}

```

Η εμφάνιση δεδομένων του ασθενή στην φόρμα "Καθημερινή Εξέταση Ασθενή" ανάλογα με τον κωδικό εισαγωγής του στο νοσοκομείο γίνεται με τον κώδικα :

```

private void Form6_Load(object sender, EventArgs e)
{
    this.κωδικός_εισαγωγήςTextBox.Text((Form4)this.form4inForm6).κωδικός_εισαγωγήςTextBox.Text;
    int a = int.Parse("0" +this.κωδικός_εισαγωγήςTextBox.Text);
    this.weekdayChekTableAdapter.FillBykodisagogi(this.nosokomioDataSet.WeekdayChek, a);
}

```

- Για να ανοίξουμε μια άλλη φόρμα όπως την "Στοιχεία Γιατρού" εκτελούμε:

```

private void btnStixiaGiatrou_Click(object sender, EventArgs e)
{
    Form2 form2 = new Form2();
    form2.Show();
}

```

Με τον ίδιο τρόπο αν θέλουμε να ανοίξουμε μια καινούρια φόρμα προσθέτουμε ένα κουμπί και γράφουμε τον παραπάνω κώδικα αλλάζοντας το όνομα της φόρμας. Αντί για Form2 βάζουμε το όνομα της νέας φόρμας,

- Για να κλείσει η αρχική φόρμα εκτελούμε τον κώδικα :

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

Όπως και για όλες τις φόρμες προσθέτουμε ένα κουμπί στην φόρμα και γράφουμε `this.close()`; σε αυτό.

Παραπάνω είδαμε τον κώδικα που χρησιμοποιήσαμε στην εφαρμογή για τους σταθερούς υπολογιστές έτσι ώστε να μπορεί αυτή να κάνει βασικές ενέργειες, όπως αποθήκευση δεδομένων ή εμφάνιση δεδομένων από την βάση στην εφαρμογή.

9.2 Εφαρμογή φορητού υπολογιστή PPC

Είναι βασικό και σε αυτήν την εφαρμογή, για του φορητούς υπολογιστές, να προσθέσουμε τις κατάλληλες βιβλιοθήκες. Οι βιβλιοθήκες που θα χρησιμοποιήσουμε είναι οι παρακάτω :

- System;
- System.Collections.Generic;
- System.ComponentModel;
- System.Data;
- System.Drawing;
- System.Text;
- System.Windows.Forms;
- System.Data.SqlClient;
- System.Data.SqlServerCe;
- System.IO;

Στην συνέχεια θα δούμε με ποιον προγραμματιστικό τρόπο κάνουμε βασικές ενέργειες, όπως η σύνδεση της συσκευής με την βάση, η μεταφορά των δεδομένων σε αυτή και η επιστροφή των δεδομένων πίσω σε αυτήν.

- Ο παρακάτω κώδικας δείχνει πως γίνεται η σύνδεση της φορητής συσκευής με την βάση, πως παίρνουμε από αυτήν συγκεκριμένα στοιχεία από τους πίνακες και τα αποθηκεύουμε τοπικά στην φορητή συσκευή.

```
private void btnfortosi_Click(object sender, EventArgs e)
{
    string sqlServerConnection = @"Provider =SQLOLEDB;Data Source ='192.168.1.4';Initial
        Catalog = nosokomio;User Id=sa;Password=123";

    SqlCeRemoteDataAccessmyRDA=newSqlCeRemoteDataAccess("http://192.168.1.4/nosokomioPD
        A/sqlcesa35.dll", "DataSource=\\Program Files\\nosokomioPPC\\nosokomio.sdf");

    SqlCeEngine engine = new SqlCeEngine();

    engine.LocalConnectionString="DataSource=\\ProgramFiles\\nosokomioPPC\\nosokomio.sdf";
    engine.CreateDatabase();

    string strCmd = "Select * From Patient where Κωδικός_ασθενή= " + selectkodastheni;
    myRDA.Pull("Patient", strCmd, sqlServerConnection, RdaTrackOption.TrackingOnWithIndexes,
        "ErrorTable");

    string strCmd2 = @"Select * From InputPatient where Κωδικός_ασθενή=" + selectkodastheni;
    myRDA.Pull("InputPatient",strCmd2,sqlServerConnection,RdaTrackOption.TrackingOnWithIndexes
        , "ErrorInputTable");
}
```

- Εδώ βλέπουμε πως από έναν πίνακα εμφανίζουμε τα στοιχεία στα textbox με κριτήριο τον κωδικό του ασθενή.

```
void FillTextBox()
{
    myConn = new SqlCeConnection("DataSource=\\ProgramFiles\\nosokomioPPC\\nosokomio.sdf");
    myConn.Open();

    SqlCeDataAdaptermyAdapter=newSqlCeDataAdapter("SelectΌνομα,Επίθετο,Εθνικότητα,Ημερομηνία_γέννησης
        From Patient Where Κωδικός_ασθενή =" + selectkodastheni, myConn);

    DataTable Table = new DataTable("Patient");
```

```
myAdapter.Fill(Table);
```

```
txtonoma.DataBindings.Add("Text", Table, "Όνομα");
txtonoma.DataBindings.Clear();
txtepitheto.DataBindings.Add("Text", Table, "Επίθετο");
txtepitheto.DataBindings.Clear();
```

```
txtethnikotita.DataBindings.Add("Text", Table, "Εθνικότητα");
txtethnikotita.DataBindings.Clear();
dateTimePicker2.DataBindings.Add("Value", Table, "Ημερομηνία_γέννησης");
dateTimePicker2.DataBindings.Clear();
```

```
myAdapter.Dispose();
myConn.Close();
```

```
}
```

- Για να μπορέσουμε να γεμίσουμε ένα datagrid με δεδομένα από τον πίνακα που μεταφέραμε από την κεντρική βάση, χρησιμοποιούμε τον παρακάτω κώδικα :

```
private void btnananeosi_Click(object sender, EventArgs e)
{
    txtamkaisagogi.Text = txtamka.Text;
    selectkodisagogisastheni = txtamkaisagogi.Text;
```

```
myConn = new SqlConnection("DataSource=\\ProgramFiles\\nosokomioPPC\\nosokomio.sdf");
myConn.Open();
```

```
SqlCeDataAdapter myAdapter= new SqlCeDataAdapter("Select
Κωδικός_εισαγωγής,Κωδικός_ασθενή,Ημέρα_εισαγωγής,Κωδικός_γιατρού,Αιτία_εισαγωγής,Θεραε
υτική_αγωγή From InputPatient where Κωδικός_ασθενή = " + selectkodisagogisastheni,myConn);
```

```
DataSet myDataSet = new DataSet();
myAdapter.Fill(myDataSet, "Result");
dataGrid1.DataSource = myDataSet.Tables["Result"].DefaultView;
```

```
myConn.Close();
```

```
}
```

- Η διόρθωση και η αποθήκευση των νέων δεδομένων στην βάση της φορητής συσκευής γίνεται με τον κώδικα :

```
private void btnapothikevsi_Click(object sender, EventArgs e)
{
    myConn = new SqlConnection("DataSource=\\ProgramFiles\\nosokomioPPC\\nosokomio.sdf");

    myConn.Open();

    myCmd = myConn.CreateCommand();
    myCmd.CommandType = CommandType.Text;

    myCmd.CommandText = "UPDATE InputPatient SET Αιτία_εισαγωγής=" +
    textBox5.Text + ",Κωδικός_γιατρού=" + textBox4.Text + ",Θεραπευτική_αγωγή=" +
    textBox6.Text + " WHERE Κωδικός_εισαγωγής =" + cellValue0;

    myCmd.ExecuteNonQuery();
    myConn.Close();
}
```

- Εμφάνιση των δεδομένων που έχουμε επιλέξει από το datagrid της φόρμας, με την πένα του υπολογιστή τσέπης, σε textboxes και σε μια μεταβλητή :

```
private void dataGrid1_Click(object sender, EventArgs e)
{
    cellValue0 = dataGrid1[int.Parse(dataGrid1.CurrentRowIndex.ToString()), 0].ToString();
    textBox4.Text = dataGrid1[int.Parse(dataGrid1.CurrentRowIndex.ToString()), 3].ToString();
    textBox5.Text = dataGrid1[int.Parse(dataGrid1.CurrentRowIndex.ToString()), 4].ToString();
    textBox6.Text = dataGrid1[int.Parse(dataGrid1.CurrentRowIndex.ToString()), 5].ToString();
}
```

- Έλεγχος των τιμών της στήλης «Κωδικός Εξέτασης» του πίνακα WeekdayChek1 και αποθήκευση της τελευταίας τιμής σε μια μεταβλητή :

```
private void LastID()
{
    SqlConnection myConn = new SqlConnection("DataSource=\\Program
Files\\nosokomioPPC\\nosokomio.sdf");

    myConn.Open();
```

```
SqlCeDataAdapter da = new SqlCeDataAdapter("SELECT * FROM WeekdayChek1",myConn);
```

```
    DataTable dt = new DataTable();
    da.Fill(dt);

    for (int i = 0; i < dt.Rows.Count; i++)
    {
        m = Int32.Parse(dt.Rows[i][0].ToString());
    }
    m++;
    myConn.Close();
}
```

- Προσθήκη νέας γραμμής στον πίνακα WeekdayChek και τοποθέτηση μιας τιμής στην πρώτη στήλη του πίνακα :

```
private void AlterWeekdayChek()
{
    SqlConnection myConn = new SqlConnection("DataSource=\\Program
Files\\nosokomioPPC\\nosokomio.sdf");
    myConn.Open();

    myCmd = myConn.CreateCommand();
    myCmd.CommandText = @"ALTER TABLE WeekdayChek ALTER COLUMN
Κωδικός_εξέτασης IDENTITY (" + m + " ,1)";
    myCmd.ExecuteNonQuery();

    myConn.Close();
}
```

- Αποθήκευση δεδομένων σε μια νέα γραμμή στον πίνακα WeekdayChek :

```
private void button2_Click(object sender, EventArgs e)
{
    myConn = new SqlConnection("DataSource=\\Program
Files\\nosokomioPPC\\nosokomio.sdf");
    myConn.Open();

    myCmd = myConn.CreateCommand();
    myCmd.CommandText = @"INSERT INTO
WeekdayChek(Κωδικός_εισαγωγής,Κωδικός_ασθενή,Κωδικός_γιατρού,Ημερομηνία_εξέτασης,Παλμοί_ανά_λεπτό,Συστολική_πίεση,Διαστολική_πίεση,Θερμοκρασία,Αναπνευστικό) VALUES
(@Κωδικός_εισαγωγής,@Κωδικός_ασθενή,@Κωδικός_γιατρού,@Ημερομηνία_εξέτασης,@Παλμοί_ανά_λεπτό,@Συστολική_πίεση,@Διαστολική_πίεση,@Θερμοκρασία,@Αναπνευστικό)";
```

```
myCmd.Parameters.Add("@Κωδικός_εισαγωγής", SqlDbType.NChar, 10);
myCmd.Parameters.Add("@Κωδικός_ασθενή", SqlDbType.NChar, 10);
myCmd.Parameters.Add("@Κωδικός_γιατρού", SqlDbType.Int);
myCmd.Parameters.Add("@Ημερομηνία_εξέτασης", SqlDbType.DateTime);
myCmd.Parameters.Add("@Παλμοί_ανά_λεπτό", SqlDbType.NChar, 5);
myCmd.Parameters.Add("@Συστολική_πίεση", SqlDbType.NChar, 10);
myCmd.Parameters.Add("@Διαστολική_πίεση", SqlDbType.NChar, 5);
myCmd.Parameters.Add("@Θερμοκρασία", SqlDbType.NChar, 5);
myCmd.Parameters.Add("@Αναπνευστικό", SqlDbType.NChar, 10);
```

```
myCmd.Parameters["@Κωδικός_εισαγωγής"].Value = cellValue0;
myCmd.Parameters["@Κωδικός_ασθενή"].Value = txtamkaisagogi.Text;
myCmd.Parameters["@Κωδικός_γιατρού"].Value = cellValue3;
myCmd.Parameters["@Ημερομηνία_εξέτασης"].Value = dateTimePicker1.Value;
myCmd.Parameters["@Παλμοί_ανά_λεπτό"].Value = textBox1.Text;
myCmd.Parameters["@Συστολική_πίεση"].Value = textBox2.Text;
myCmd.Parameters["@Διαστολική_πίεση"].Value = textBox3.Text;
myCmd.Parameters["@Θερμοκρασία"].Value = textBox4.Text;
myCmd.Parameters["@Αναπνευστικό"].Value = comboBox1.Text;
myCmd.ExecuteNonQuery();
myConn.Close();
}
```

- Μεταφορά μόνο των νέων δεδομένων, που προστέθηκαν στους πίνακες της φορητής συσκευής και στους πίνακες της βάσης του Server.

```
private void button1_Click(object sender, EventArgs e)
{
    string sqlServerConnection = @"Provider =SQLOLEDB;Data Source =192.168.1.4;Initial Catalog
    = nosokomio;User Id=sa;Password=123";

    SqlCeRemoteDataAccess myRDA =
    new SqlCeRemoteDataAccess("http://192.168.1.4/nosokomioPDA/sqlcesa35.dll",
    "DataSource=\\Program Files\\nosokomioPPC\\nosokomio.sdf");

    myRDA.Push("InputPatient", sqlServerConnection, RdaBatchOption.BatchingOff);
    myRDA.Dispose();
}
```


10.0 Πηγές

- <http://el.wikipedia.org/wiki/%CE%94%CE%AF%CE%BA%CF%84%CF%85%CE%BF%85%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CF%8E%CE%BD>
- <http://videogameslab.wordpress.com/2009/01/08/intro-csharp-net/>
- <http://msdn.microsoft.com/en-us/>
- .Net Compact Framework Programming with C#
- <http://www.codeproject.com/>