Ατομική Διπλωματική Εργασία

Υπηρεσίες διαδικτύου για υποστήριξη Κατ' οίκον φροντίδας

Ευθυμία Παπαδοπούλου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υπηρεσίες διαδικτύου για υποστήριξη κατ' οίκον φροντίδας ευπαθών ομάδων

Ευθυμία Παπαδοπούλου

Επιβλέπων Καθηγητής

Ανδρέας Πιτσιλλίδης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2010

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Ανδρέα Πιτσιλλίδη για την ανάθεση αυτής της εργασίας, καθώς επίσης και για τις συμβουλές για την ολοκλήρωση και συγγραφή της.

Επίσης, θα ήταν παράλειψη να μην ευχαριστήσω τον υποψήφια Διδάκτορα κ. Δημοσθένη Γεωργιάδη για την πολύτιμη βοήθεια του για την διεκπεραίωση της εργασίας. Καθώς και την κ. Βαρβάρα για τον χρόνο που μας αφιέρωσε για την συνέντευξη που κάναμε.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για τη συμπαράστασή σε όλους τους τομείς όχι μόνο κατά τη διάρκεια εκπόνησης της εργασίας αυτής, αλλά καθ' όλα τα χρόνια των σπουδών μου.

Περίληψη

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μιας διαδικτυακής εφαρμογής, χρησιμοποιώντας τις δυνατότητες που μας δίνει η τεχνολογία του διαδικτύου στις μέρες μας.

Συγκεκριμένα στόχος μας ήταν η δημιουργία μιας διαδικτυακής εφαρμογής που θα εξυπηρετεί ασθενείς που πάσχουν από χρόνιες ασθένειες καθώς και τις ιατρικές ομάδες που τους παρακολουθούν και θα διευκολύνει την μεταξύ τους ανταλλαγή στοιχείων και πληροφοριών για μια σωστή διαχείριση της ασθένειας.

Μέσα από την εφαρμογή αυτή ο Ασθενής θα μπορεί να δει τα προσωπικά του στοιχεία και στοιχεία που αφορούν την ασθένεια του, χωρίς όμως να μπορεί να τροποποιήσει κάποια από τα παραπάνω στοιχεία παρά μόνο το δημογραφικά (διεύθυνση, τηλέφωνα κ.α.). Επιπλέον, πρόσβαση θα έχουν οι γιατροί, νοσοκόμες και άλλοι εμπλεκόμενοι οι οποίοι θα βλέποντας τα δεδομένα των ασθενών και θα τα τροποποιούν αν χρειάζεται αναλόγως καθώς επίσης θα μπορούν να διευθετήσουν ραντεβού κ.α..

Η διαδικτυακή εφαρμογή υλοποιήθηκε σε λειτουργικό περιβάλλον windows XP με την χρήση της γλώσσας προγραμματισμού Visual Basic και των εργαλείων που προσφέρουν οι τεχνολογίες Asp.net και Visual studio.net. Ο web server που χρησιμοποιήθηκε είναι ο iis της Microsoft, ο οποίος έχει τη δυνατότητα να εμφανίζει δυναμικές σελίδες και να επικοινωνεί με την βάση δεδομένων.

Η βάση δεδομένων που χρησιμοποιήθηκε είναι η υφιστάμενη της εφαρμογής DITIS και η πλατφόρμα διαχείρισης που χρησιμοποιήσαμε για την μελέτη και τροποποίηση της είναι η SQL Server 2005.

Η προσπάθειά μας τέλος επικεντρώθηκε στο να υλοποιήσου μια εύχρηστη και λειτουργική διεπιφάνεια βασισμένη στην τεχνολογία του Web Usability.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ 1

1.1. Σκοπός 2

1.2. Μηχανογράφησης δεδομένων ιατρικής υφής 2

1.3. Διαδικτυακής εφαρμογής έναντι Λογισμικού 3

1.4. Το διαδίκτυο και οι εφαρμογές του μέχρι σήμερα 5

ΚΕΦΑΛΑΙΟ 2

ΕΥΧΡΗΣΤΙΑ ΚΑΙ ΔΙΑΔΙΚΤΥΟ 7

2.1. Χρήσιμα σημεία σχεδίασης και το περιεχόμενο 8
2.1.1. Περιεχόμενο πληροφορίας 8
2.1.2. Συνοχή περιεχομένου 8
2.1.3. Μέγεθος οθόνης 8
2.1.4. Ταχύτητα πλοήγησης 9
2.1.5. Χρήση πολυμέσων (multimedia) 10

2.2. Σημαντικές αρχές 10
2.2.1. Αρχική σελίδα 11
2.2.2. Πλοήγηση 11

2.3. Ενσωμάτωση ευχρηστίας σε ένα σύστημα 12

2.4. Παραδείγματα σχετικών εφαρμογών και ανάλυση ευχρηστίας τους 13
2.4.1. Διαδικτυακές Εφαρμογές που μελετήθηκαν 14
2.4.2. WorldHealthRecord 14
2.4.3. NoMoreClipboard 16
2.4.4. GoogleHealth 17
2.4.5. Dr.I-Net 18

ΚΕΦΑΛΑΙΟ 3

ΑΠΑΙΤΗΣΕΙΣ 20

3.1. Γενική Περιγραφή Συστήματος 20
3.1.1. Διεπαφές συστήματος 20
3.1.2. Τυπικοί Χρήστες 21
3.1.3. Τυπικές Εργασίες 23

3.2. Περιορισμοί 24

3.3. Απαιτήσεις Απόδοσης 24

3.4. Χαρακτηριστικά Συστήματος 25
3.4.1. Αξιοπιστία 25
3.4.2. Διαθεσιμότητα 26
3.4.3. Ασφάλεια 26
3.4.4. Συντηρησιμότητα 26
3.4.5. Μεταφερσιμότητα 26

ΚΕΦΑΛΑΙΟ 4

ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ 28

4.1. IIS Web Server 28

4.2. .NET Framework και Web forms 29

4.3. ASP.NET 32

4.4. HTML 33

4.5. Visual Studio 34
4.5.1. Visual Studio 2005 35
4.5.2. ADO.NET 35
4.5.3. Visual Web Developer 36

4.6. Βάση Δεδομένων 37
4.6.1. SQL 37
4.6.2. SQL Server 2005 38

4.7. Visual Basic 2005 39

ΚΕΦΑΛΑΙΟ 5

ΑΝΑΛΥΣΗ ΤΗΣ ΥΠΑΡΧΟΝΤΟΣ ΣΥΣΤΗΜΑΤΟΣ 40

5.1. ΕΦΑΡΜΟΓΗ DITIS 40
5.1.1. Αρχιτεκτονική DITI 41

5.1.2. Διαπροσωπία που υπάρχουν στον DITI 42

5.2. Πίνακες Βάσης Δεδομένων 44

5.3. Διαχωρισμός του συστήματος σε ενότητες και ανάλυσή τους 47
5.3.1. Appointment module 48
5.3.2. Medication module 49
5.3.3. Health care teams' module 50
5.3.4. Messaging module 50
5.3.5. Exams' module 51
5.3.6. Diagnosis module 52
5.3.7. Therapies module 53
5.3.8. Symptom Diary module 53

ΚΕΦΑΛΑΙΟ 6

ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ 55

6.1. Αρχιτεκτονική 55
6.2.2. Αρχιτεκτονική Server - Side scripting 56
6.1.2. Αρχιτεκτονική Διεπιφάνειας 57

6.2. Ασφάλεια 57

6.3. Χρήση συγκεκριμένων component της ASP.Net 59
6.3.1. Χρήση StyleSheet CSS 59
6.3.2. Το αρχείο web.config 61
6.3.3. Χρήση Master page 63
6.3.4. Χρήση Sessions 65
6.3.5. Υποστήριξη Πολλών Γλωσσών (Multilanguage Support) 66
6.3.6. Χρήση Site Maps 67
6.3.7. Χρήση Stored Procedures 69

6.4. Περιγραφή υλοποίησης για κάθε σελίδα 70
6.4.1. Login σελίδα 70
6.4.2. Σελίδες specialists 72
6.4.3. Σελίδες patients 84

ΚΕΦΑΛΑΙΟ 7

ΑΞΙΟΛΟΓΗΣΗ 87

7.1. Αξιολόγηση 87

7.2. Πίνακες Αξιολόγησης 88

ΚΕΦΑΛΑΙΟ 8

ΣΥΜΠΕΡΑΣΜΑΤΑ 91

8.1. Συμπεράσματα 91

8.2. Μελλοντική εργασία 92

ΒΙΒΛΙΟΓΡΑΦΙΑ 94

ΠΑΡΑΡΤΗΜΑ Α 1

Α1. Ερωτηματολόγιο διαπροσωπείας χρήστη 1

ΠΑΡΑΡΤΗΜΑ Β 1

Β.1. Σενάριο χρήσης 1

Κεφάλαιο 1

Εισαγωγή

1.1.	Σκοπός	1
1.2.	Μηχανογράφησης δεδομένων ιατρικής υφής	2
1.3.	Διαδικτυακής εφαρμογής έναντι Λογισμικού	3
1.4.	Το διαδίκτυο και οι εφαρμογές του μέχρι σήμερα	5

Στους έντονους ρυθμούς του σύγχρονου τρόπου ζωής, αυτό που όλοι σχεδόν επιθυμούμε είναι περισσότερες ανέσεις, ώστε να μπορέσουμε να απολαύσουμε όσο το δυνατόν καλύτερα την κάθε στιγμή. Ο σημερινός πολίτης έχει περισσότερες απαιτήσεις από ότι στο παρελθόν όσον αφορά τις υπηρεσίες που του προσφέρονται και ιδικά στον τομέα της υγείας που είναι ένας χώρος ευαίσθητος.

Υπάρχει η απαίτηση από τους ασθενείς, ο γιατρός να είναι ενήμερος για την κατάσταση της υγείας τους και να ενημερώνει και τους ίδιους, αποτελεσματικά. Επομένως η ανάγκη για εύκολη πρόσβαση σε μεγάλο αριθμό πληροφοριών όσο αφορά την εικόνα ενός ασθενή, όπως και η ανάγκη για πληρέστερη εικόνα παρελθόντων ιατρικών εξετάσεων και ενεργειών που αφορούν τον ασθενή, μας οδηγούν σε προβληματισμό.

Μια βασική ανάγκη στις μέρες μας είναι η συνεχής παρακολούθηση και περίθαλψη ανθρώπων με χρόνιες ασθένειες. Δυστυχώς, χρόνο με τον χρόνο τόσο στην Κύπρο όσο και στο εξωτερικό αυξάνονται οι ασθενείς με χρόνιες ασθένειες(όπως με καρκίνο, καρδιακές παθήσεις κ.α.). Σημαντική πρόκληση, είναι η εύρεση λύσης με στόχο ο ασθενείς να έχει την δυνατότητα επιλογής να παραμείνει στο σπίτι και στην θαλπωρή της οικογένειας του, ανεξάρτητα αν η αρρώστια του είναι ιάσιμη, ελεγχόμενη ή σε τελικό στάδιο.

Επομένως, αυξάνεται η ανάγκη για ένα ιατρικό περιβάλλον όπου δίνεται έμφαση στην κατ' οίκον περίθαλψη, προσφέροντας στον ασθενή νοσηλευτική φροντίδα και ιατρική παρακολούθηση όταν χρειάζεται καθώς επίσης και ψυχοκοινωνική στήριξη τόσο σε αυτόν όσο και στην οικογένεια του.

Πως μπορούμε λοιπόν να δώσουμε λύση στα πιο πάνω;
Βλέπουμε πως για την εύκολη και άμεση ενημέρωση τόσο του ασθενή αλλά και όλων των εμπλεκομένων ανά πάσα στιγμή μπορούμε να χρησιμοποιήσουμε την τεχνολογία.
Η τεχνολογία είναι αενάως αναπτυσσόμενο κομμάτι του ανθρώπινου πολιτισμού, που σκοπό δεν έχει άλλο, από το να βελτιώνει συνεχώς το βιοτικό μας επίπεδο σε όλους τους τομείς. Συγκεκριμένα μπορούμε να εκμεταλλευτούμε την ευρεία χρήση του διαδικτύου που υπάρχει στις μέρες μας και να δώσουμε λύση αναπτύσσοντας συστήματα στον παγκόσμιο ιστό.

1.1. Σκοπός

Σκοπός της παρούσας διπλωματικής εργασίας είναι η χρησιμοποίηση των δυνατοτήτων που μας δίνει η τεχνολογία του διαδικτύου (internet) προς όφελος του τομέα υγείας. Συγκεκριμένα στόχος μας είναι η δημιουργία μιας διαδικτυακής εφαρμογής που θα εξυπηρετεί ασθενείς που πάσχουν από χρόνιες ασθένειες καθώς και τις ιατρικές ομάδες που τους παρακολουθούν και θα διευκολύνει την μεταξύ τους ανταλλαγή στοιχείων και πληροφοριών για μια σωστή διαχείριση της ασθένειας . Απώτερος σκοπός είναι η δημιουργία υπηρεσιών διαδικτύου με δυνατότητα την απομακρυσμένη ιατρική παρακολούθηση και περίθαλψη ευπαθών ομάδων στο οικιακό τους περιβάλλον μέσω της αποτελεσματικής διαχείρισης και της συνεργασίας των εικονικών ομάδων ιατρικής περίθαλψης.

1.2. Μηχανογράφησης δεδομένων ιατρικής υφής

Το πρόβλημα της συνεχούς επικοινωνίας ανάμεσα σε ιατρικό και παραϊατρικό προσωπικό στον ιατρικό τομέα και συγκεκριμένα στην κατ' οίκον περίθαλψη δεν αποτελεί ένα καινούργιο πρόβλημα.

Η ιδέα της μηχανογράφησης δεδομένων ιατρικής υφής πρωτοξεκίνησε από τα λογιστήρια σχετικών οργανισμών και όχι από το πρόβλημα που περιγράψαμε πιο πάνω. Στην συνέχεια έχει γίνει μια προσπάθεια μηχανογράφησης των ιατρικών πληροφοριών ενός ασθενή. Βλέπουμε πως για την διευκόλυνση κυρίως εσωτερικά ενός οργανισμού (π.χ. νοσοκομείου) έχουν αναπτυχθεί διάφορα συστήματα. Η ανάπτυξη των εφαρμογών αυτών γίνεται ειδικά για τις ανάγκες της εκάστοτε εταιρίας.

Ωστόσο με αυτή την λύση δεν έχουμε ταυτόχρονα άμεση, εύκολη, εύρωστη αλλά και ασφαλή πρόσβαση στο ιατρικό αρχείο του ασθενή από τις ομάδας περίθαλψης ενός ασθενή καθώς και τον ίδιο το ασθενή , με αποτέλεσμα να μην έχουμε μια πλήρη εικόνα για την κατάσταση του ασθενή από οποιοδήποτε χώρο.
Με την χρήση των διαφορών εργαλείων που μας παρέχει στις μέρες μας το διαδίκτυο θα μπορούμε να έχουμε οποιαδήποτε ώρα σε οποιοδήποτε χώρο τα δεδομένα που χρειαζόμαστε. [1]

1.3. Διαδικτυακής εφαρμογής έναντι Λογισμικού

Η ομάδα ιατρικής περίθαλψης απαιτείται να είναι σε συνεχή επαφή χωρίς απαραίτητα να βρίσκεται στον ίδιο φυσικό χώρο. Η κινητή συσκευή ή ο κινητός προσωπικός υπολογιστής παίζουν σημαντικό παράγοντα αφού παρέχουν στους διάφορους εμπλεκομένους ελευθερία κινήσεων.

Ένα τοπικό λογισμικό είναι εγκατεστημένο στο προσωπικό υπολογιστή είτε στην κινητή συσκευή. Δυστυχώς όμως όταν μιλάμε για τοπικό λογισμικό, εκτός από την αρχική του εγκατάσταση στη συσκευή θα πρέπει στην περίπτωση οποιαδήποτε αλλαγή στο λογισμικό να γίνει ανανέωση του σε όλες της συσκευές που εγκαταστάθηκε. Αυτό είναι ένα πολύ σημαντικό μειονέκτημα όσο αφορά το τοπικό λογισμικό αφού δεν είναι όλοι οι χρήστες καλοί γνώστες των υπολογιστών. Επομένως τόσο για την εγκατάσταση όσο και για την ενημέρωση σε πολλές περιπτώσεις θα χρειαστεί κάποιο τρίτο πρόσωπο να αναλάβει την επίτευξη τους.

Ο συντονισμός, καθώς και η συνεχής ενημέρωση της χρησιμοποιούμενης συσκευής είναι ένα από τα στοιχεία όπου μειονεκτεί σε σύγκριση με την ιδέα δημιουργίας

ιστοσελίδας. Εκτός από αυτά όμως υπάρχει και το πρόβλημα της συμβατότητας του λογισμικού με το λειτουργικό σύστημα που χρησιμοποιεί η συσκευή. Το λογισμικό είναι δεσμευμένο να δουλεύει μόνο για το λειτουργικό σύστημα που γράφτηκε. Για παράδειγμα αν αγοράσει λογισμικό για Windows είναι υποχρεωμένος να έχει το συγκεκριμένο λειτουργικό σύστημα στη συσκευή που χρησιμοποιεί και στη περίπτωση που επιθυμεί να το αλλάξει για παράδειγμα σε Linux το λογισμικό δεν θα μπορούσε να λειτουργήσει.

Στις μέρες μας η αναγκαιότητα χρήσης του διαδικτύου είναι προφανής ακόμα και για ανθρώπους που δεν έχουν άμεση ή έμμεση σχέση με το αντικείμενο των Ηλεκτρονικών Υπολογιστών. Ο πλούτος της παρεχόμενης πληροφορίας, η αμεσότητα ενημέρωσης οι προσφερόμενες υπηρεσίες και τα εργαλεία που συγκροτούν την έννοια που αποκαλούμε συνολικά διαδίκτυο, προκαλεί δέος.

Πολλοί μάλιστα μιλούν για 'τεχνολογικό αποκλεισμό' σε περίπτωση που κάποιος δεν προσπαθήσει να προσαρμοστεί και να ενσωματώσει τη νέα αυτή πραγματικότητα στο τρόπο ζωής και εργασίας του. Εμείς θα εκμεταλλευτούμε αυτή την διάδοση για να εξαλείψουμε κάποια από τα μειονεκτήματα του τοπικού λογισμικού.

Με την δημιουργία μιας ιστοσελίδας δεν θα χρειάζεται ο απλός χρήστης να κάνει εγκατάσταση ή ενημέρωση κάπου λογισμικού. Το μόνο που θα πρέπει να κάνει είναι μέσο ενός φυλλομετρητή να καλέσει την ιστοσελίδα. Ο διαχειριστής της ιστοσελίδας θα αλλάζει κάτι μια φορά και αμέσως θα γίνονται οι αλλαγές για όλους τους χρήστες.

Η ιδέα υλοποίησης μιας ιστοσελίδας ωστόσο έχει και αυτή κάποια μειονεκτήματα που αφορούν όμως τους σχεδιαστές της και όχι τον απλό χρήστη. Πρέπει να υπάρχει συμβατότητα με όλους τους φυλλομετρητές έτσι ώστε να μπορεί κάποιος χρήστης να δει την ιστοσελίδα μας. Πρέπει να υπάρχει διαχωρισμός στην τεχνοτροπία και την λογική σχεδίαση της ιστοσελίδας για να μπορεί να δουλεύει σε διάφορες οθόνες (κινητό, 15 ιντσών οθόνη κ.α.).

Με την ιδέα της ιστοσελίδας δεν χάνουμε την ελευθερία κινήσεων στους διάφορους εμπλεκομένους. Αυτό συμβαίνει γιατί τόσο στις μέρες μας όσο και στο μέλλον όλες οι

κινητές συσκευές μπορούν να ενωθούν με το διαδίκτυο. Επομένως δεν χάνουμε την κινητικότητα που είναι απαραίτητη για την εφαρμογή μας.

1.4. Το διαδίκτυο και οι εφαρμογές του μέχρι σήμερα

Ολόκληρος ο χώρος της πληροφορικής, επομένως και το διαδίκτυο, αλλάζει συνεχώς μορφή και προσαρμόζεται στις τεχνολογικές εξελίξεις. Ο νόμος του Μουρ, μας λέει πως η υπολογιστική ισχύ των PC διπλασιάζεται κάθε 18 μήνες και οι τιμές τους πέφτουν στο μισό. Αυτός ο νόμος είχε ανάλογες επιπτώσεις στο διαδίκτυο και στις τεχνολογίες που το περιβάλλουν. Μπορούμε να πούμε πως υπάρχουν τρείς γενιές που αφορούν την ανάπτυξη του διαδικτύου.

Την πρώτη αποτελούσαν στατικές σελίδες με έξυπνα γραφικά που έλκυαν την προσοχή του επισκέπτη. Αυτή η γενιά των εφαρμογών περιοριζόταν σε στατικές φόρμες προς συμπλήρωση, οι οποίες σπάνια συνδέονταν με βάσεις δεδομένων. Η λειτουργικότητα όσο και η εμφάνιση των ιστοσελίδων της πρώτης γενιάς ήταν φτωχικές.

Η δεύτερη γενιά των εφαρμογών διαδικτύου έφερε ένα πιο δυναμικό περιβάλλον χάρη στις προεκτάσεις της γλώσσας HTML(DHTML -Dynamic HTML) και των ενεργών ιστοσελίδων (ASP-Active Server Pages, JSP-JavaServer Pages, PHP). Σαν αποτέλεσμα έχουμε ένα πιο λειτουργικό και φιλικό περιβάλλον λόγο της σύνδεσης του δυναμικού περιεχόμενου με τις στατικές σελίδες. Σε αυτό βοήθησε και η JavaScript, μια γλώσσα για τον προγραμματισμό δυναμικών ιστοσελίδων που υποστηρίζεται από όλα τα προγράμματα αναζήτησης. Ο κώδικας της JavaScript βρίσκεται ενσωματωμένος στην ίδια την ιστοσελίδα και εκτελείται στον υπολογιστή του χρήστη (για αυτό και ονομάζεται Client side scripting), σε αντίθεση με τις σελίδες ASP ή JSP, οι οποίες εκτελούνται στο διακομιστή (Server side).

Σήμερα, βρισκόμαστε στην τρίτη γενιά του διαδικτύου και βλέπουμε νέες, έξυπνες και ιδιαίτερα λειτουργικές εφαρμογές, να κατακλύζουν τους δικτυακούς τόπους. Στο χρήστη παρέχονται περισσότερες λειτουργίες, σύνθετες αναζητήσεις και πλούσιο περιεχόμενο. Το εμπόριο και οι τραπεζικές συναλλαγές είναι μερικές από τις περιοχές

στις οποίες έχει εισβάλει το διαδίκτυο, παρέχοντας μία εύκολη λύση για τις καθημερινές μας εργασίες.

Σημαντικό ρόλο στην εξέλιξη των εφαρμογών αλλά και του διαδικτύου γενικότερα, εκτός από την υποστήριξη των προγραμμάτων αναζήτησης και τις βελτιώσεις στην γλώσσα HTML, έπαιξαν τα προγράμματα διακομιστών, όπως και η τηλεπικοινωνιακή υποδομή. Μια γραμμή ISDN είναι σε θέση να προσφέρει εύρος ζώνης 128Mb και σταθερότητα, με αποτέλεσμα την ταχεία εμφάνιση των σελίδων. Ενώ με την χρήση της νέας τεχνολογία DSL, μπορούμε να "κατεβάσουμε" στον τοπικό υπολογιστή μας το περιεχόμενο μια ιστοσελίδα σε δευτερόλεπτα. Επιπλέον, η ανάγκη για άμεση ανταλλαγή των πληροφοριών μεταξύ διαφορετικών εφαρμογών οδήγησε στη δημιουργία της γλώσσας XML, ενός προτύπου που αναδεικνύεται στο ASCII του διαδικτύου.

Η συνεχείς χρησιμοποίηση του διαδικτύου στην καθημερινή μας ζωή ανέβασε τις προσδοκίες των χρηστών ακόμα πιο ψηλά με αποτέλεσμα οι απαιτήσεις των εφαρμογών να είναι πλέον μεγαλύτερες.

Κεφάλαιο 2

Ευχρηστία και Διαδίκτυο

2.1. Χρήσιμα σημεία σχεδίασης και το περιεχόμενο 8

2.2. Σημαντικές αρχές 10

2.3. Ενσωμάτωση ευχρηστίας σε ένα σύστημα 12

2.4. Παραδείγματα σχετικών εφαρμογών και ανάλυση ευχρηστίας τους 13

Μια διεπιφάνεια χρήστη-εφαρμογής ή στα αγγλικά interface, είναι το κομμάτι της εφαρμογής που είναι υπεύθυνο για την αλληλεπίδραση με το χρήστη. Από τη στιγμή λοιπόν που αυτός είναι και ο μόνος τρόπος άμεσης επικοινωνίας των δύο, επιβάλλεται ο προσεκτικός και μελετημένος σχεδιασμός και υλοποίηση της. Και αυτό γιατί η αλληλεπιδραστικότητα μιας εφαρμογής καθορίζει σε μεγάλο βαθμό την αποδοχή της από την πλευρά των χρηστών.

Μια σωστά σχεδιασμένη ιστοσελίδα πρέπει να είναι εύχρηστη, λειτουργική και όμορφη, ώστε να πετυχαίνει το στόχο για τον οποίο κατασκευάστηκε. Το καλό design δεν είναι καθόλου υποκειμενικό, γιατί δεν έχει να κάνει με το γούστο του καθενός. Βέβαια για να είναι μια εφαρμογή επιτυχημένη θα πρέπει να ακολουθεί κάποιες προδιαγραφές. Ένας από τους σημαντικότερους παράγοντες είναι η ευχρηστία (usability).

Μελετώντας σχετικές εργασίες και άρθρα από εξειδικευμένους επιστήμονες σε θέματα ευχρηστίας στο Διαδίκτυο, θα προσπαθήσουμε να εστιάσουμε στην έννοια της ευχρηστίας (usability), να δώσουμε παραδείγματα σωστού και λανθασμένου σχεδιασμού ιστοσελίδων και να προτείνουμε τη χρυσή τομή ανάμεσα στην όμορφη παρουσίαση και την εύχρηστη και λειτουργική πλοήγηση.

Η ευχρηστία είναι μια παράμετρος που συχνά δεν λαμβάνει την δέουσα προσοχή, αν και είναι σημαντική όσο αφορά την επιτυχία ενός site. Ευχρηστία σημαίνει σχεδίαση για τους επισκέπτες (χρήστες) ενός site, και όχι βάσει των προσωπικών προτιμήσεων του σχεδιαστή και δημιουργού του. [9,10]

2.1. Χρήσιμα σημεία σχεδίασης και το περιεχόμενο

Υπάρχει ένα σύνολο κρίσιμων σημείων, τα οποία πρέπει να λαμβάνει υπόψη ο σχεδιαστής κατά τη διαδικασία ανάπτυξης μιας ιστοσελίδας. Παρακάτω αναφέρονται ενδεικτικά τα κυριότερα από αυτά:

2.1.1. Περιεχόμενο πληροφορίας

Το περιεχόμενο μίας σελίδας είναι πρωταρχικής σημασίας. Αν το περιεχόμενο υστερεί, τότε ούτε ο τρόπος που θα το παρουσιάσουμε θα μπορέσει να βελτιώσει την ευχρηστία της σελίδας μας.

Το κείμενο θα πρέπει να είναι σύντομο, περιεκτικό και να μπορεί να διαβαστεί επιλεκτικά. Δυστυχώς η προσοχή των χρηστών στο διαδίκτυο είναι περιορισμένης διάρκειας με αποτέλεσμα η ανάγνωση μακροσκελών κειμένων στον υπολογιστή να είναι κουραστική.

2.1.2. Συνοχή περιεχομένου

Όλα τα μέρη της ιστοσελίδας πρέπει να αποτελούν μέρος ενός αρμονικού συνόλου. Μέρη της ιστοσελίδας που εξυπηρετούν τον ίδιο ή παρόμοιο σκοπό πρέπει να έχουν την ίδια ή παρόμοια εμφάνιση (π.χ. εικονίδια).

2.1.3. Μέγεθος οθόνης

Λαμβάνοντας υπόψη ότι οι χρήστες διαθέτουν οθόνες με διαφορετική ανάλυση, είναι καλό να υπάρχει μια εικόνα για τη μέση ανάλυση που χρησιμοποιούν ανά πάσα στιγμή οι περισσότεροι χρήστες στο Διαδίκτυο, και να αξιοποιηθεί ανάλογα κατά το σχεδιασμό των σελίδων. Τέτοια στοιχεία διατίθενται από εταιρίες στατιστικών ερευνών (π.χ. www.statmarket.com).

Τα τελευταία χρόνια η χρήση της ανάλυσης 640 X 480 έχει μειωθεί ιδιαίτερα, και πάνω από το 50% των χρηστών του Internet χρησιμοποιεί ανάλυση οθόνης 800 X 600. Είναι λοιπόν χρήσιμο οι σχεδιαστές να χρησιμοποιούν την παραπάνω ανάλυση ως βασική για τη δημιουργία σελίδων, ώστε να αποφεύγεται το κατέβασμα (scroll down).

2.1.4. Ταχύτητα πλοήγησης

Ο σημαντικότερος παράγοντας που πρέπει να υπολογίζεται στη διαδικασία σχεδίασης ιστοσελίδων είναι η ταχύτητα με την οποία οι θα μπορούν οι χρήστες να "κατεβάζουν" το συνολικό περιεχόμενο στον υπολογιστή τους.

Σύμφωνα με τον διακεκριμένο ερευνητή Robert B. Miller, όταν το σύστημα αποκρίνεται στο χρήστη μέσα σε 0,1 δευτερόλεπτα, ο πρώτος θεωρεί ότι το σύστημα έχει άμεση επικοινωνία και ακαριαία απόκριση. Όταν η απόκριση ολοκληρώνεται μέσα σε ένα δευτερόλεπτο, ο χρήστης αντιλαμβάνεται την καθυστέρηση αλλά είναι ακόμη πλήρως ικανοποιημένος από το χρόνο ανοίγματος της σελίδας. Το μέγιστο χρονικό όριο αναμονής, όπως αποδεικνύεται από στατιστικές έρευνες, φτάνει τα 10 δευτερόλεπτα. Κατά τη διάρκεια των 10 αυτών δευτερολέπτων, καλό είναι να φαίνεται κάποιο σήμα προόδου. Πέρα από το συγκεκριμένο χρονικό διάστημα ο χρήστης δυσανασχετεί ως προς την απόδοσή του συστήματος και τείνει προς εύρεση άλλων.

Ο παρακάτω πίνακας(Πίνακας 1.1) παρουσιάζει το μέγιστο μέγεθος που πρέπει να έχει μια ιστοσελίδα (σε kilobytes), σε σχέση με την ταχύτητα μεταφοράς, έτσι ώστε ο χρόνος αναμονής να μην υπερβαίνει τα 10 δευτερόλεπτα.

Πίνακας 2.1

Ταχύτητα Σύνδεσης	Ταχύτητα Μεταφοράς	Προτεινόμενο Μέγιστο Μέγεθος για 10 sec
Modem 56 K	7 K/s (Kbytes/Sec)	70 K
ISDN 64 K	8 K/s	80 K
ADSL 256 K	27 K/s	270 K

Όπως φαίνεται και στον πίνακα, αν μία σελίδα είναι επιβαρυμένη με πολλά γραφικά, αυτό αποβαίνει σε βάρος της ταχύτητας. Αυτό βέβαια δεν αποτελεί κανόνα για όλους, αφού σε κάποιες περιπτώσεις σκοπός της επιχείρησης είναι να προβληθεί με τον

ομορφότερο και εντυπωσιακότερο δυνατό τρόπο, αψηφώντας τη χρονική επιβάρυνση για τους χρήστες της.

2.1.5. Χρήση πολυμέσων (multimedia)

Μολονότι η χρήση πολυμέσων σε μια σελίδα καθιστά την παρουσίαση πιο εντυπωσιακή και φιλική προς το χρήστη, η αναπόφευκτη χρονική καθυστέρηση συνήθως κάνει τους σχεδιαστές πολύ προσεκτικούς κατά την ενσωμάτωσή του.

Οι εικόνες καλό είναι να μην καταλαμβάνουν μεγάλο χώρο στην οθόνη και να έχουν το μέγεθος που ακριβώς απαιτείται για την παρουσίασή τους. Η συρρίκνωση της εικόνας, που πολλές φορές λανθασμένα χρησιμοποιείται από τους σχεδιαστές, δεν φέρνει τα αναμενόμενα αποτελέσματα. Αντί αυτού, όταν θέλουμε να φέρουμε μια εικόνα σε συγκεκριμένες διαστάσεις χρησιμοποιούμε τις μεθόδους της εστίασης και της κοπής, οι οποίες μειώνουν και το μέγεθος της εικόνας. Αφού γίνει αυτό, αποθηκεύουμε την τελική εικόνα σε συμπιεσμένη μορφή (συνήθως JPG ή GIF).

Η κινούμενη εικόνα και το video, παρόλο που επιβαρύνουν το website, είναι αρκετές φορές σκόπιμα. Παρακάτω αναφέρονται ενδεικτικά κάποιες περιπτώσεις που η εφαρμογή τέτοιων μέσων είναι χρήσιμη:

- Απεικόνιση τρισδιάστατων αντικειμένων
- Απεικόνιση αλλαγής καταστάσεων σε αντικείμενα
- Εντυπωσιασμός και εστίαση του ενδιαφέροντος του χρήστη
- Προβολή διαφημιστικών spots
- Προώθηση τηλεοπτικών γεγονότων μικρής χρονικής διάρκειας
- Slide show ή video για την γρήγορη εκμάθηση του site

Σε κάθε περίπτωση χρήσης πολυμέσων, πρέπει να δίνεται μεγάλη προσοχή στο τελικό μέγεθος και να μη θυσιάζεται η ουσιαστική πληροφορία.

2.2. Σημαντικές αρχές

Πέρα από τις προτεινόμενες τεχνικές κατά την κατασκευή μιας ιστοσελίδας, υπάρχουν και κάποιες σημαντικές αρχές που αναφέρονται σε ολόκληρο το website, τις οποίες

προτείνουμε να εφαρμόζει ο σχεδιαστής του. Παρακάτω αναφέρονται οι πιο ενδεικτικές:

2.2.1. Αρχική σελίδα

Η αρχική σελίδα είναι η πρώτη με την οποία έρχεται σε επαφή ο χρήστης και σίγουρα παίζει τον κύριο ρόλο στην αξιολόγηση όλου του δικτυακού τόπου από τον επισκέπτη. Πρωταρχικός στόχος της αρχικής σελίδας είναι να απαντά στα ερωτήματα "Πού βρίσκομαι;" και "Τι δείχνει αυτό το site;". Επίσης, στην αρχική σελίδα θα πρέπει να παρουσιάζονται οι πιο πρόσφατες ανακοινώσεις αλλά και κάποια σημαντικά κομμάτια πληροφορίας που θέλουμε να διαβάσει ο επισκέπτης.

Όλες οι σημαντικές πληροφορίες θα πρέπει να βρίσκονται στο πάνω μέρος της οθόνης χωρίς να χρειάζεται ολίσθηση (scrolling) για να τις δει κανείς. Ακόμη καλό θα ήταν η οικοσελίδα να έχει πολλά links.

2.2.2. Πλοήγηση

Ο επισκέπτης ενός δικτυακού τόπου πρέπει να έχει τη δυνατότητα να κινείται με αρκετά εύκολο τρόπο και χωρίς να χάνεται. Συγκεκριμένα, το περιβάλλον πλοήγησης που υπάρχει στη σελίδα θα πρέπει να πληροφορεί το χρήστη σχετικά με το πού βρίσκεται, ποιες σελίδες ή συνδέσμους έχει ήδη επισκεφθεί και πού βρίσκονται οι επιλογές πλοήγησης.

Αν και πολλές από τις παραπάνω πληροφορίες παρέχονται απευθείας από το φυλλομετρητή, οι σελίδες του website πρέπει να πληροφορούν τόσο για τη θέση του χρήστη ως προς το Web αλλά και ως προς το συγκεκριμένο site, όσο και για τη συγκεντρωτική δομή του δικτυακού τόπου, έτσι ώστε να είναι δυνατή η πλήρης κατανόηση του χρήστη για την όλη δομή του ιστότοπου.

Γενικά πρέπει να έχουμε σαν στόχο την κατατόπιση του χρήστη γύρω από το θέμα και τα περιεχόμενα της ιστοσελίδας.[9]

2.3. Ενσωμάτωση ευχρηστίας σε ένα σύστημα

Σύμφωνα με τον Jakob Nielsen, (http://www.useit.com/papers/heuristic), οι δέκα όροι που πρέπει να πληρούνται για να ικανοποιούνται οι απαιτήσεις ευχρηστίας ενός συστήματος είναι:

- Ορατότητα της κατάστασης του συστήματος

  Το σύστημα πρέπει να κρατά πάντα ενήμερους τους χρήστες για την κατάστασή του, μέσω κατάλληλων πληροφοριών απόκρισης, μέσα σε λογικό χρονικό διάστημα.
- Εναρμονισμός συστήματος και πραγματικού κόσμου

  Το σύστημα πρέπει να αποκρίνεται σε "γλώσσα" κατανοητή από το χρήστη, και η πληροφορία να παρουσιάζεται με φυσικό τρόπο και λογική σειρά.
- Έλεγχος από το χρήστη και ελευθερία κινήσεων

  Οι χρήστες συχνά ενεργούν κατά λάθος, και δεν μπορούν να "επανορθώσουν" βάσει των εργαλείων που τους παρέχονται. Κάθε σύστημα πρέπει να προσφέρει δυνατότητες αναίρεσης και επανάληψης.
- Συνέπεια και Σταθερότητα

  Οι χρήστες δεν πρέπει να αμφιταλαντεύονται για το νόημα συγκεκριμένων λέξεων, καταστάσεων ή ενεργειών. Πρέπει να χρησιμοποιούνται ενιαίοι και σταθεροί κανόνες σε όλες τις περιπτώσεις.
- Παρεμπόδιση σφαλμάτων

  Ένας καλός σχεδιασμός που αποτρέπει προβλήματα εκ των προτέρων είναι πολύ καλύτερη τεχνική από τη χρήση μηνυμάτων λάθους.
- Αναγνώριση

  Τα αντικείμενα, οι ενέργειες και οι επιλογές πρέπει να είναι ορατά. Ο χρήστης δεν μπορεί να θυμάται πληροφορίες κατά τη διάρκεια της πλοήγησής του από μια οθόνη σε μια άλλη. Οι οδηγίες για τη χρήση του συστήματος πρέπει να είναι επίσης ορατές και διαθέσιμες όποτε χρειαστούν.
- Προσαρμοστικότητα και αποδοτικότητα

  Το σύστημα πρέπει να σχεδιαστεί με τέτοιο τρόπο, ώστε να επιτρέπει τόσο στους έμπειρους όσο και στους άπειρους χρήστες να διαμορφώνουν τον τρόπο

και την ταχύτητα με την οποία αλληλεπιδρούν με το σύστημα, σύμφωνα με τα δικά τους κριτήρια.

- Αισθητική και απλός σχεδιασμός

  Κάθε αντικείμενο ή οθόνη πρέπει να περιέχει μόνο τις απαραίτητες για το χρήστη πληροφορίες. Κάθε επιπλέον πληροφορία μπορεί να τον αποπροσανατολίσει και να τον αποτρέψει από να τη χρησιμοποιήσει.
- Βοήθεια για αναγνώριση, διάγνωση και διόρθωση λαθών

  Τα μηνύματα λάθους πρέπει να εκφράζονται σε απλή και κατανοητή γλώσσα. Επίσης πρέπει να εντοπίζουν επακριβώς το πρόβλημα και να προτείνουν σαφώς τη λύση.
- Παροχή Βοήθειας και Τεκμηρίωσης

  Παρόλο που είναι καλύτερο να μπορεί ο χρήστης να χρησιμοποιήσει το σύστημα χωρίς να χρειάζεται να ανατρέξει σε οποιοδήποτε ηλεκτρονικό εγχειρίδιο, πολλές φορές η παροχή εγγράφου λεπτομερούς τεκμηρίωσης είναι απαραίτητη. Κάθε τέτοιου είδους πληροφορία πρέπει να είναι εύκολα προσβάσιμη, ακριβής και όχι πολύ μεγάλη.[9]

2.4. Παραδείγματα σχετικών εφαρμογών και ανάλυση ευχρηστίας τους

Ανά καιρούς έχουν δημιουργηθεί κάποιες εφαρμογές ηλεκτρονικού ιατρικού φακέλου στο web. Η έννοια του ηλεκτρονικού ιατρικού φακέλου αντικατοπτρίζει, κυρίως, την διαχείριση της ιατρικής πληροφορίας με μια κοινή ονομασία και κοινό περιεχόμενο. Ο οργανισμός τυποποίησης ISO έχει ιδρύσει την Τεχνική Επιτροπή 215(TC215) με στόχο την προτυποποίηση στον τομέα της ιατρικής πληροφορίας και την ανάπτυξη προτύπων για τη διαχείριση της ιατρικής πληροφορίας και των ιατρικών διαδικασιών. Η επιδίωξη είναι η κατάλληλη πληροφορία να είναι διαθέσιμη όταν και όπου απαιτείται η υποστήριξη αποφάσεων.

Μπορούμε να πούμε πως η λύση του ηλεκτρονικού ιατρικού φακέλου είναι ελλιπής ως προς το σύστημα που εμείς θέλουμε να εφαρμόσουμε. Ωστόσο θεωρήσαμε καλό να μελετήσουμε κάποιες από τις υπάρχουσες αυτές εφαρμογές για να πάρουμε μια ιδέα πως υλοποιούνται στις μέρες μας διαδικτυακές εφαρμογές που αφορούν την καταγραφή ιατρικών πληροφοριών.

Πιο κάτω βλέπουμε παραδείγματα που αφορούν την υλοποίηση του ηλεκτρονικού ιατρικού φακέλου στο web.

2.4.1. Διαδικτυακές Εφαρμογές που μελετήθηκαν

Συγκεκριμένα μελετήθηκαν οι διαδικτυακές εφαρμογές WorldHealthRecord, NoMoreClipboard, GoogleHealth και Dr.I-Net. Οι κατασκευαστές των πιο πάνω διαδικτυακών εφαρμογών προσπάθησαν να κάνουν ένα καλό και εύχρηστο σχεδιασμό. Ωστόσο παρατηρήθηκαν κάποια κοινά πλεονεκτήματα και μειονεκτήματα.

Κάποια από τα κοινά πλεονεκτήματα είναι ότι:

- υπάρχει slide show που σε βοηθά να πλοηγηθείς ευκολότερα στην ιστοσελίδα
- μπορεί κανείς να τυπώσει μια κάρτα που θα την έχει στο πορτοφόλι του σε περίπτωση εκτάκτου ανάγκης να μπορούν να βρεθούν βασικά στοιχεία (όπως είναι η κάρτα που γράφει την ομάδα αίματος)
- μπορεί να τυπώσεις όλες τις πληροφορίες σε αναγνώσιμη μορφή
- υπάρχει ασφάλεια

Κάποια κοινά μειονεκτήματα που εντοπίσαμε είναι τα ακόλουθα:

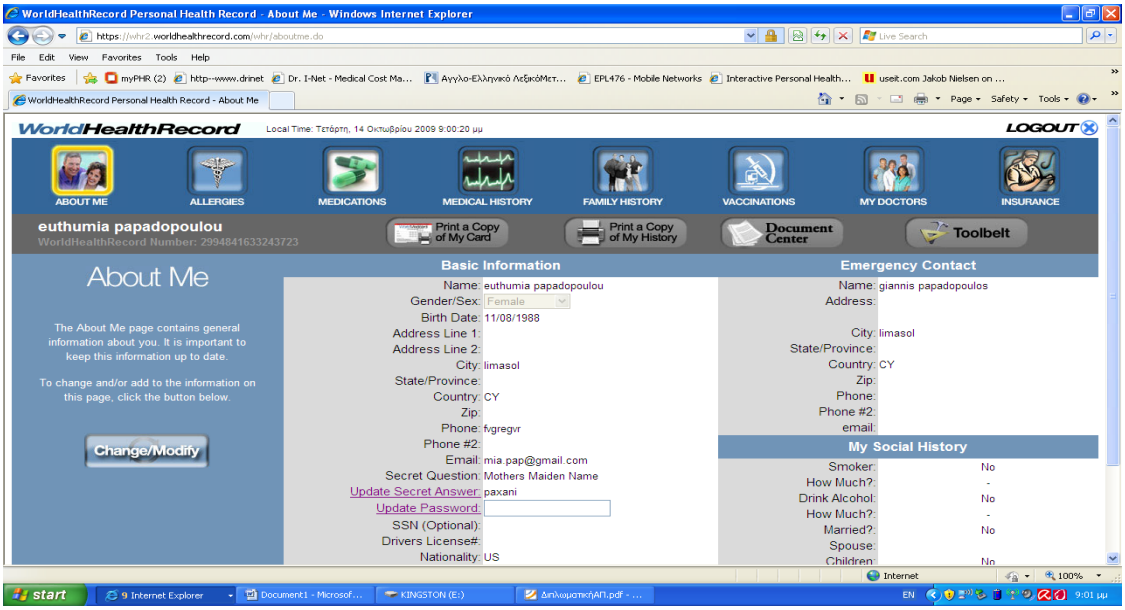
- κάθε φορά που ενωνόμαστε με άλλο link, επειδή μπαίνει σε ασφαλισμένη περιοχή, βγάζει σχετικό μήνυμα και αυτό είναι ενοχλητικό για τον χρήστη
- ένα μεγάλο μειονέκτημα είναι πως καθώς γεμίζονται οι φόρμες δεν ελέγχονται διάφορα πεδία αν εισήχθησαν σωστά (π.χ. μπορεί να βάλλει κανείς ημερομηνία του μέλλοντος)
- απαλά χρώματα που βοηθούν την ανάγνωση της πληροφορίας
- ξεχωρίζουν τα link

Εκτός από τα κοινά χαρακτηριστικά παρατηρήσαμε και κάποια άλλα χαρακτηριστικά που παρουσιάζονται σε κάθε μια από αυτές.

2.4.2. WorldHealthRecord

Το WorldHealthRecord είναι μια ιστοσελίδα αρκετά εύχρηστη. Η απλότητα που διέπει αυτή την ιστοσελίδα και ο καλός σχεδιασμός της, όπως χρήση εικονιδίων, βοηθούν πολύ τον χρήστη να πλοηγηθεί άνετα και γρήγορα στην ιστοσελίδα αυτή.

Σχήμα 2.1



Κάποια από τα πλεονεκτήματα είναι ότι:

- μπορεί να δουλέψει και πάνω σε iPhone ή iTouch προσθέτοντας απλά στην διεύθυνση το /iPhone
- αν σε μια φόρμα δεν έβαλες κάποιο απαραίτητο στοιχείο σε ενημερώνει ακριβώς ποιο στοιχείο δεν συμπλήρωσες
- μπορεί ένας χρήστης να μεταφέρει και να αποθηκεύσει στο λογαριασμό του πληροφορίες που βρίσκονται σε έντυπη μορφή

Ωστόσο υπάρχουν και κάποια μειονεκτήματα όπως για παράδειγμα:

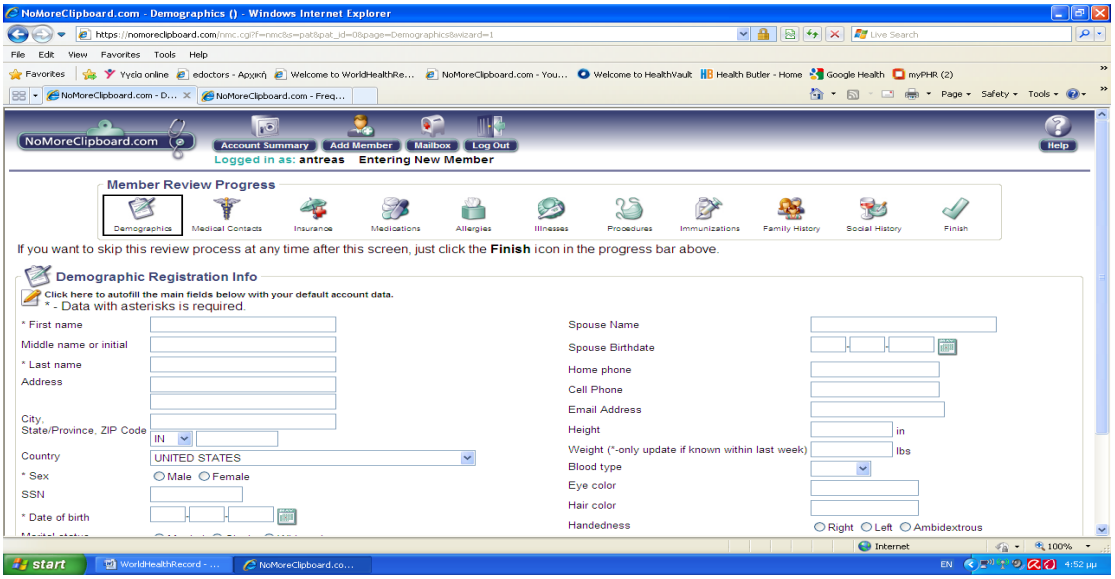
- τα μηνύματα που βγάζει για να ενημερώσει τους χρήστες φεύγουν πολύ γρήγορα με αποτέλεσμα να μην μπορούν οι πλείστοι χρήστες να τα διαβάσουν
- δεν ενημερώνει ότι τέλειωσε η διαδικασία δημιουργίας του λογαριασμού
- η διαδικασία εισαγωγής Medication και Allergies είναι αργή (Ωστόσο είναι καλό το γεγονός ότι εκτός από την λίστα που υπάρχει στην βάση σου αφήνει χώρο να γράψεις εσύ κάτι άλλο)

Ένα ωραίο εργαλείο που υπάρχει είναι το Toolbelt μέσα στο οποίο υπάρχουν εργαλεία που σε βοηθούν να βρεις την μάζα σώματος σου, διαβήτη (για διαβητικούς), πόσες θερμίδες πρέπει να καις την ημέρα με βάση το φύλο σου, την ηλικία σου και το βάρος σου και τέλος μπορείς να βάλεις κάποιες υπενθυμίσεις.

2.4.3. NoMoreClipboard

Το nomoreclipboard.com είναι και αυτό μια ιστοσελίδα εύχρηστη και καλά σχεδιασμένη. Βλέπουμε πως και εδώ οι σχεδιαστές έκαναν χρήση σχετικών εικονιδίων για να κάνουν το περιβάλλον πιο όμορφο αλλά και πιο εύκολο ως προς την χρήση.

Σχήμα 2.2



Κάποια πλεονεκτήματα είναι ότι:

- μπορείς να διαλέξεις μέγεθος γραμμάτων και αυτό είναι καλό αφού έχουμε να κάνουμε με άτομα όλων τον ηλικιών
- το site αυτό σου επιτρέπει να έχεις μέχρι και 10 εγγραφές στον λογαριασμό που θα ανοίξεις
- υπάρχει μια λίστα από αλλεργίες, αρρώστιες, φάρμακα κ.τ.λ., έτσι ο χρήστη δεν κάνει λάθος την ονομασία
- μπορεί ένας χρήστης να μεταφέρει και να αποθηκεύσει στο λογαριασμό του πληροφορίες που βρίσκονται σε έντυπη μορφή
- μπορείς να βάλεις φωτογραφία για κάθε άτομο που έχεις οργανώσει στο λογαριασμό σου, έτσι ώστε αν γίνει κάποιο ατύχημα να μπορούν να δουν ποια ιατρικά στοιχεία ανταποκρίνονται σε ποιον

Από την άλλη υπάρχουν και κάποια μειονεκτήματα:

- το μήνυμα που βγάζει αν συμπληρώσαμε τα πεδία που είναι απαραίτητα σε μια φόρμα είναι γενικό και δεν μας λέει που ακριβώς είναι το λάθος

- περιορίζει τον χρήστη με την επιλογή από την λίστα που δίνεται της ονομασίας π.χ. μιας αλλεργίας έτσι αν δεν ξέρει την ακριβή ονομασία δεν μπορεί να συμπληρώσει την φόρμα

Σχήμα 2.3

2.4.4. GoogleHealth

Η ιστοσελίδα αυτή δεν είναι τόσο ελκυστική αφού δεν χρησιμοποιεί πολλές μεταφορές και εικονίδια. Ωστόσο είναι και αυτή μια από τις πιο καλές ιστοσελίδες όσον αφορά δημιουργία και επεξεργασία ιατρικού φακέλου στο web.

Η google.com/health είναι μια ιστοσελίδα που μας προσφέρει και ιατρικές πληροφορίες π.χ. για τα συμπτώματα που μπορεί να έχει μια αρρώστια. Παράλληλα, συνεργάζεται με νοσοκομεία, εργαστήρια και φαρμακεία.

Έχει πολλά πλεονεκτήματα όπως:

- μπορείς να δημιουργήσεις στο λογαριασμό σου πάνω από ένα ιατρικό φάκελο (συγγενών, φίλων)
- το γεγονός ότι ελέγχει για κάποια σύγκρουση μεταξύ των αλλεργιών που έχει κάποιος χρήστης και τα φάρμακα που ίσως παίρνει και τον ειδοποιά σχετικά
- μπορείς να μοιραστείς τα αρχεία σου με άλλους και ανά πάσα στιγμή να κάνει add ή delete κάποιο που έχει πρόσβαση στα δεδομένα σου

- δεν σου ζητά προσωπικά στοιχεία όπως όνομα, ταυτότητα απλά μόνο τα βασικά που χρειάζεται να γνωρίζει και ο γιατρός, όπως φύλο ημερομηνία γέννησης βάρος και ύψος
- μπορείς να βρεις γιατρό από κάθε περιοχή κάνοντας search με βάση ιδιότητα, περιοχή ή με το όνομα. Και μας δίνει την περιοχή και το website αν υπάρχει

Εκτός από τα πολλά πλεονεκτήματα που μας δίνει έχουμε και κάποια μειονεκτήματα όπως:

- στην αναζήτηση του γιατρού δεν μας δίνει πάντα τα σωστά αποτελέσματα
- ένας απλός ασθενής δεν γνωρίζει πάντα το όνομα μιας ασθένειας ή μιας αλλεργίας και δυστυχώς δεν μας δίνει εναλλακτική λύση για να βάλουμε κάποιο στοιχείο

2.4.5. Dr.I-Net

Αυτή η ιστοσελίδα δεν είναι και τόσο καλά σχεδιασμένη σε σύγκριση με τις προηγούμενες που αναφέρθηκαν. Υπάρχουν πολλά πράγματα στην οθόνη και αυτό

Σχήμα 2.4

κουράζει και συγχύζει ένα απλό χρήστη. Δεν πρέπει να ξεχνάμε ότι οι χρήστες της ιστοσελίδας μπορεί να είναι και ηλικιωμένοι.

Υπάρχουν τόσο πλεονεκτήματα τόσο και κάποια μειονεκτήματα.

Κάποια από τα πλεονεκτήματα είναι τα ακόλουθα:

- έχει χώρο για να γράψει κάποιος το feed back
- στα lab result μπορείς να βάλεις εικόνα και φαίνεται αυτόματα η ώρα και η ημερομηνία

Κάποια μειονεκτήματα που εντοπίσαμε είναι τα ακόλουθα:

- αφήνει πολύ κενό χώρο στο τέλος πολλών hyperlink
- δεν ανοίγει άλλο tag όταν πατήσουμε σε ένα link που οδηγά εκτός της ιστοσελίδας
- δεν μπορείς να βάλεις τα δικά σου αποτελέσματα όσον αφορά τα labs results
- υπάρχουν πολλές πληροφορίες στην οθόνη με αποτέλεσμα να επέρχεται σύγχυση στο χρήστη μας

Κεφάλαιο 3

Απαιτήσεις

3.1. Γενική Περιγραφή Συστήματος 20

3.2. Περιορισμοί 24

3.3. Απαιτήσεις Απόδοσης 24

3.4. Χαρακτηριστικά Συστήματος 25

Σε αυτό το κεφάλαιο θα δούμε μια γενική περιγραφή του συστήματος . Θα καθοριστεί η σχέση του συστήματος με διάφορες διαπροσωπείες όπως οι χρήστες, το υλικό και το λογισμικό που θα χρησιμοποιηθεί. Στη συνέχεια θα αναφερθούμε στους τυπικούς χρήστες του συστήματος και θα δούμε κάποιες από τις τυπικές εργασίες. Τέλος, θα δούμε θέματα αξιοπιστίας , διαθεσιμότητας και ασφάλειας του συστήματος.

3.1. Γενική Περιγραφή Συστήματος

Η υλοποίηση της διαδικτυακής εφαρμογής θα είναι βασισμένη σε προηγούμενα έγγραφα μελέτης και ανάλυση των απαιτήσεων. Τα έγγραφα αυτά έχουν μια γενική μορφή έτσι ώστε να μπορεί να εφαρμοστεί σε οποιοδήποτε ιατρικό τομέα της κατ' οίκον φροντίδας. Συγκεκριμένα σε αυτά καταγράφονται επακριβώς οι βασικές απαιτήσεις ενός ιατρικού περιβάλλοντος στο οποίο δίνεται έμφαση στην κατ' οίκον φροντίδα. Ποίο κάτω θα δούμε μερικά σημαντικά κομμάτια που αφορούν τις απαιτήσεις και προδιαγραφές για να κατανοήσουμε τόσο τις λειτουργίες της εφαρμογής μας όσο και κάποια χαρακτηριστικά που πρέπει να έχει.

3.1.1. Διεπαφές συστήματος

Το σύστημα που θα υλοποιήσουμε δε συνεργάζεται με κάποιο άλλο σύστημα, ωστόσο επικοινωνεί με μια βάση στην οποία θα φυλάγονται και θα βρίσκονται όλες οι σχετικές

πληροφορίες για κάθε εγγεγραμμένο χρήστη. Στόχος είναι οποιοσδήποτε χρήστης του διαδικτύου να έχει τη δυνατότητα να γίνει μέλλος στο συγκεκριμένο σύστημα και να επωφεληθεί από τις λειτουργίες του.

3.1.1.1. Διεπαφές χρηστών:

Δίνεται η δυνατότητα αλληλεπίδρασης χρήστη – συστήματος μέσω διαφόρων σελίδων. Η κάθε σελίδα μπορεί να αποτελεί κάποια οθόνη ή κάποια φόρμα συγκεκριμένης λειτουργίας. Ο χρήστης μέσω των σελίδων αυτών θα μπορεί να πλοηγηθεί στις διάφορες λειτουργίες που του παρέχει το σύστημα.

3.1.1.2. Επιφάνεια υλικού:

Για τη σωστή λειτουργία του συστήματος χρειαζόμαστε ένα web server στον οποίο θα βρίσκεται η βάση δεδομένων του συστήματος. Ακόμη για να μπορεί να λειτουργήσει το σύστημα ο χρήστης θα πρέπει να έχει κάποια συσκευή (π.χ. ηλεκτρονικός υπολογιστής) που να ενώνεται με το διαδίκτυο και μέσω της οποίας θα μπορεί να συνδέεται στο λογαριασμό του.

3.1.1.3. Επαφές λογισμικού:

Για το λόγο ότι το σύστημά μας θα λειτουργεί μέσω του διαδικτύου, ο οποιοσδήποτε χρήστης θα μπορεί να έχει πρόσβαση σ' αυτό οποιαδήποτε χρονική στιγμή σε οποιοδήποτε χώρο φτάνει να υπάρχει σύνδεση με το διαδίκτυο. Για να μπορούμε να το κάνουμε αυτό χρειάζεται να υπάρχει στον υπολογιστή του χρήστη ο φυλλομετρητής (browser). Για την υλοποίηση του συστήματος χρειάζεται να έχουμε τα λογισμικά SQL server, πάνω στο οποίο θα τρέξει και θα μελετηθεί η υπαρκτή βάση δεδομένων. Ακόμη θα χρειαστεί το λογισμικό visual studio 2005 στο οποίο θα υλοποιήσουμε τη διαπροσωπία του συστήματος.

3.1.2. Τυπικοί Χρήστες

Οι χρήστες που σχετίζονται άμεσα και έμμεσα με το σύστημα συχνά είναι οι τυπικοί χρήστες του συστήματος. Οι χρήστες αυτοί πρόκειται αν επηρεαστούν σημαντικά από την εισαγωγή του νέου συστήματος. Ίσως υποχρεωθούν να αλλάξουν σημαντικά τον τρόπο που εκτελούσαν παλαιότερα την συγκεκριμένη εργασία. Αυτοί είναι:

3.1.2.1. Πρωτεύοντες χρήστες:

Οι πρωτεύοντες χρήστες είναι αυτοί που θα χρησιμοποιούν το σύστημα μας επί καθημερινής βάσης.

Patient: Οι ασθενείς είναι μία από τις ομάδες των βασικών, τυπικών χρηστών του συστήματος. Οι ασθενείς μπορούν να ανήκουν σε διάφορες ηλικίες με βασικές τα παιδιά άνω των 10 χρονών, τους έφηβους, ενήλικες και τους μεσήλικες. Αυτοί οι χρήστες θα αλληλεπιδρούν με το σύστημα σε σχεδόν καθημερινή βάση και ο κάθε ένας από αυτούς έχει διαφορετικό μορφωτικό επίπεδο.

Nurse: Οι νοσοκόμες είναι και αυτή μια σημαντική ομάδα του συστήματός μας και αποτελούν μια από τις ομάδες των βασικών, τυπικών χρηστών του συστήματος. Έχουν μια συγκεκριμένη μόρφωση στο τομέα της υγείας. Παρατηρούμε ότι είναι διαφόρων ηλικιών.

Specialist: Σε αυτή την ομάδα χρηστών ανήκουν όλες οι ομάδες των γιατρών (π.χ. παθολόγος, φυσιοθεραπευτής κ.α.). Οι χρήστες αυτού του συστήματος είναι πολύ βασική ομάδα. Έχουν μια συγκεκριμένη μόρφωση στο τομέα της υγείας και το μορφωτικό τους επίπεδο είναι ψηλό.

3.1.2.2. Δευτερεύοντες χρήστες:

Οι δευτερεύοντες χρήστες είναι αυτοί που θα χρησιμοποιούν το σύστημα αλλά όχι τόσο συχνά.

Social Worker: Οι κοινωνικοί λειτουργοί ανήκουν και αυτοί στις ομάδες των χρηστών που αλληλεπιδρούν με το σύστημα μας και μπορούν να κάνουν συγκεκριμένες εργασίες. Έχουν μια συγκεκριμένη μόρφωση και μπορεί να έχουν διαφορετικό μορφωτικό επίπεδο. Παρατηρούμε ότι και αυτή η ομάδα χρηστών μπορεί να αποτελείται από άτομα διαφορετικών ηλικιών.

Συγγενείς ασθενών: Οποιοσδήποτε συγγενής κάποιου ασθενή μπορεί αν θελήσει να δει την κατάσταση της υγείας του ασθενή. Μπορεί αυτό να το κάνει με το να συνδεθεί με το σύστημα μέσω του λογαριασμού του χρήστη. Στην ομάδα αυτή μπορεί να ανήκουν άτομα με διαφορετικό μορφωτικό επίπεδο και με διαφορετική ηλικία.

Είναι σημαντικό να αναφέρουμε πως κάθε χρήστης έχει διαφορετική εμπειρία και γνώση από υπολογιστές. Από ανύπαρκτη (ίσως χρειαστεί να γίνουν και κάποια μαθήματα εκμάθησης του συστήματος) έως πάρα πολύ καλή. Άρα για να καλύψουμε αυτό το κενό θα πρέπει το σύστημα μας να είναι εύχρηστο και απλό για να βοηθά τα

άτομα με ελάχιστες έως καθόλου γνώσεις υπολογιστών και προήγησης στο διαδίκτυο να μάθουν να το χρησιμοποιούν.

3.1.3. Τυπικές Εργασίες

Όλα τα δοθέντα από το χρήστη στοιχεία βρίσκονται στη βάση μας. Μπορούμε να έχουμε πρόσβαση σ' αυτά δυναμικά μέσω SQL query. Ο χρήστης μέσω αυτών θα έχει πρόσβαση σε όλα τα στοιχεία που τον αφορούν. Πιο κάτω βλέπουμε τις πιο σημαντικές λειτουργίες που μπορεί να εκτελέσει κάποιος εγγεγραμμένος χρήστης. Οι τυπικές εργασίες του συστήματος διαφέρουν ανάλογα με την ομάδα που ανήκει ο χρήστης.

Όλοι οι χρήστες πρέπει να συνδεθούν με το σύστημα και στη συνέχεια θα εμφανιστεί στον καθένα από αυτούς το μενού επιλογών με τις λειτουργίες που έχουν δικαίωμα να εκτελέσουν. Στην εργασία σύνδεσης με το σύστημα το σύστημα θα ελέγχει αν το username και το password που δόθηκαν είναι σωστά. Σε περίπτωση που κάποιο από τα δύο είναι λάθος θα εμφανιστεί στο χρήστη μία οθόνη με το κατάλληλο μήνυμα λάθους.

Οι χρήστες Nurse, social worker, specialists στο μενού επιλογών τους θα έχουν τις επιλογές appointments, search, messages, my virtual teams, tools, help.

- Στην επιλογή appointments ο χρήστης θα μπορεί να προσθέτει, διορθώνει, βλέπει ή να διαγράφει κάποιο από τα υφιστάμενα ραντεβού.
- Μέσω της επιλογής search μπορεί κάποιος από τους πλοηγητές να ψάξει και να βρει πληροφορίες για συγκεκριμένο ασθενή ή ακόμη για οποιοδήποτε άλλο πρόσωπο που είναι εγγεγραμμένο στο πρόγραμμα.
- Επιλέγοντας το message μπορεί κανείς να διαχειριστεί τα μηνύματα που έχει λάβει από κάποιο άλλο μέλος που βρίσκεται στην εικονική του ομάδα. Συγκεκριμένα μέσω αυτής της επιλογής μπορεί κανείς να διαβάσει και να στείλει γραπτό μήνυμα ή και να παντήσει σε ένα εισερχόμενο.
- Το σύστημα μέσω του my virtual teams μας δίνει την επιλογή να προσθέτουμε, να επανεπεξεργαζόμαστε ή ακόμη και να διαγράφουμε άτομα από το προσωπικό αρχείο οποιουδήποτε χρήστη φτάνει να ανήκει στην ίδια εικονική ομάδα.

Η ομάδα Patients έχει άλλες πρωτεύουσες επιλογές στο μενού, συγκεκριμένα ένας Patient μπορεί να δει μόνο βασικές λειτουργίες και μπορεί να αλλάξει μόνο το δημογραφικά του στοιχεία. το σύνηθες log in και log out.

- Μέσω της επιλογής my medication μπορεί να δει ανά πάσα στιγμή τα φάρμακα που λάμβανε και αυτά που θα πρέπει να λαμβάνει και τη συχνότητα λήψης τους.
- Μπορεί να δει τα δημογραφικά στοιχεία του και να τα τροποποιήσει.
- Και πάλι επιλέγοντας το log out πηγαίνει στην επιλογή log in για να συνδεθεί αν επιθυμεί με έναν άλλο λογαριασμό.

3.2. Περιορισμοί

Η εφαρμογή αποτελεί ένα web application που τρέχει σε οποιοδήποτε browser. Γίνεται χρήση του SQL Server 2005 και είναι γραμμένη σε ASP.Net. Δεν υπάρχουν απαιτήσεις συγκεκριμένης μνήμης που πρέπει να έχουν οι υπολογιστές που θα τρέχουν το λογισμικό αφού μιλάμε για ιστοσελίδα. Το μόνο που χρειάζεται είναι να υπάρχει κάποιος browser και σύνδεση στο διαδίκτυο.

Όσον αφορά τους περιορισμούς σχεδίασης πρέπει βασικά να λάβουμε υπόψη μας τις διάφορες τεχνολογίες που θα χρησιμοποιήσουμε. Η βάση δεδομένων είναι υλοποιημένη σε Microsoft SQL SERVER και η υλοποίηση των σελίδων θα γίνει σε ASP.Net με χρήση της γλώσσας προγραμματισμού Visual Basic. Με τις πιο πάνω τεχνολογίες θα αποφευχθούν συγκεκριμένοι και ιδιαίτεροι περιορισμοί στον σχεδιασμό γιατί η Visual Studio διαθέτει όλα τα εργαλεία που υποστηρίζουν τη δημιουργία διαπροσωπείας βάσεων δεδομένων με ιστοσελίδες.

Ένας ακόμη περιορισμός είναι ότι χωρίς internet browser είναι αδύνατη η χρήση του συστήματος.

3.3. Απαιτήσεις Απόδοσης

Όπως κάθε σύστημα έτσι και το δικό μας πρέπει να έχει κάποιες απαιτήσεις. Το σύστημα θα πρέπει να μπορεί να χρησιμοποιείται ταυτόχρονα από πολλούς χρήστες αφού πρόκειται για ένα σύστημα web application.

Ακόμη όπως προαναφέραμε πρέπει να υπολογίζεται στη διαδικασία σχεδίασης η ταχύτητα με την οποία θα μπορούν οι χρήστες να "κατεβάζουν" το συνολικό περιεχόμενο στον υπολογιστή τους. Βασικά ο χρόνος αναμονής δεν πρέπει να υπερβαίνει τα 10 δευτερόλεπτα.

Τέλος πρέπει να λάβουμε ακόμη υπόψη των αριθμών των δεδομένων που θα αποθηκεύονται στη βάση.

3.4. Χαρακτηριστικά Συστήματος

3.4.1. Αξιοπιστία

Η αξιοπιστία είναι ένας πολύ σημαντικός παράγοντας για την ομαλή λειτουργία του συστήματος. Το σύστημα όταν υλοποιηθεί θα πρέπει να ελεχθεί για τυχόν σφάλματα και λάθη που μπορεί να υπάρχουν στο σύστημα. Έτσι ο χρήστης δεν θα έλθει αντιμέτωπος ποτέ με τυχόν σφάλματα που δεν μπορεί να αντιμετωπίσει και να χειριστεί. Ακόμη πριν παραδοθεί το σύστημα στους χρήστες θα πρέπει να είμαστε σίγουροι ότι παράγει σωστά αποτελέσματα.

Για μεγαλύτερη αξιοπιστία το σύστημα για κάθε είδος χρήστη θα δίνει πρόσβαση μόνο στους πίνακες που έχει δικαίωμα να δει, ή να επεξεργαστεί. Με αυτό τον τρόπο δεν θα μπορεί να αλλάξει δεδομένα τα οποία βρίσκονται καταχωρημένα στη βάση στα οποία δεν έχει δικαιώματα.

Τέλος πρέπει τα εργαλεία που θα χρησιμοποιήσουμε για την κατασκευή του συστήματος να είναι δοκιμασμένα προκειμένου να είναι αξιόπιστα. Η Visual Studio είναι κατασκευασμένη και δοκιμασμένη προκειμένου να είναι συνεχώς αξιόπιστη, ασφαλής, εφαρμόσιμη και συμβατή με τις εφαρμογές. Επίσης προσφέρει έναν ασύγκριτο συνδυασμό χαρακτηριστικών ασφαλείας, πολλαπλών κλιμάκων (scalability) και ενδολειτουγικότητα. Παρόλο που η Visual Studio ενσωματώνει προοδευτικά στοιχεία, είναι κατασκευασμένη έτσι ώστε να υπάρχει εναρμόνιση με προηγούμενες εκδόσεις, όποτε αυτό είναι δυνατόν.

3.4.2. Διαθεσιμότητα

Χρησιμοποιώντας το διαδίκτυο το σύστημα μας θα είναι διαθέσιμο ανά πάσα στιγμή για όλο το εικοσιτετράωρο και σε οποιοδήποτε χώρο (φτάνει να υπάρχει σύνδεση το Internet). Οι εγγεγραμμένοι χρήστες θα μπορούν απλά με το να κάνουν login να παρακολουθούν τυχόν εξελίξεις.

3.4.3. Ασφάλεια

Ένα από τα σημαντικότερα προβλήματα των δικτυακών εφαρμογών, είναι η ασφάλεια. Λόγο της φύσης της εφαρμογής, όσοι διαθέτουν σύνδεση με το internet μπορούν να επισκεφτούν την ηλεκτρονική διεύθυνση της εφαρμογής. Αυτό προσφέρει την δυνατότητα στα μέλη του συστήματος να μπαίνουν στην εφαρμογή από οποιοδήποτε σημείο του κόσμου, αλλά παράλληλα το ίδιο δικαίωμα έχουν και όλοι οι άλλοι χρήστες του internet.

3.4.4. Συντηρησιμότητα

Η συντήρηση είναι ένα σημαντικό στάδιο στην ανάπτυξη και εφαρμογή ενός συστήματος. Το σύστημα θα πρέπει να επιβιώσει μέσα από τις συνεχείς τεχνολογικές εξελίξεις, δηλαδή θα πρέπει να υπάρχει δυνατότητα αναβάθμισης και προσαρμογής του σε νέα τεχνολογικά δεδομένα.
Ακόμη ο κώδικας με τον οποίο θα υλοποιηθεί το σύστημα θα πρέπει να είναι γραμμένος προσεκτικά, για να μπορεί εύκολα να το κατανοήσει οποιοσδήποτε προγραμματιστής έστω και αν δεν είναι ο αρχικός κατασκευαστής.

3.4.5. Μεταφερσιμότητα

Επειδή το σύστημα θα είναι ουσιαστικά μια ιστοσελίδα το μόνο που θα χρειαστεί ο οποιοσδήποτε χρήστης είναι να έχει πάνω στον υπολογιστή του ένα browser. Επομένως δεν χρειάζεται κάθε φορά που θα θέλουμε να χρησιμοποιήσουμε διαφορετικό υπολογιστή να εγκαταστήσουμε κάποιο πρόγραμμα.

Η αρχιτεκτονική του συστήματος επηρεάζει τη λειτουργικότητα, απόδοση, αποτελεσματικότητα, συντήρηση και επέκταση του για όλη τη διάρκεια ζωής του. Για αυτό χρειάζεται ιδιαίτερη προσοχή στην σχεδίαση ώστε το σύστημα να είναι σωστά δομημένο.

Κεφάλαιο 4

Εργαλεία Ανάπτυξης

4.1.	IIS Web Server	28
4.2.	.NET Framework και Web forms	29
4.3.	ASP.NET	32
4.4.	HTML	33
4.5.	Visual Studio	34
4.6.	Βάση Δεδομένων	37
4.7.	Visual Basic 2005	39

Το επόμενο στάδιο για την ανάπτυξη της εφαρμογής μας είναι η εύρεση κατάλληλης τεχνολογίας για την υλοποίηση αυτή. Για την υλοποίηση της εφαρμογής μας χρησιμοποιήθηκαν διάφορα σύγχρονα εργαλεία/προγράμματα.

Η διαδικτυακή εφαρμογή μας υλοποιήθηκε σε λειτουργικό περιβάλλον windows XP με την χρήση των εργαλείων που προσφέρουν οι τεχνολογίες Asp.net και Visual studio.net. Η βάση δεδομένων χρησιμοποιεί την πλατφόρμα διαχείρισης SQL 2005. Θα έχουμε τη δυνατότητα της δυναμικής εμφάνισης της σελίδας και της επικοινωνίας με την βάση δεδομένων.

Στο συγκεκριμένο κεφάλαιο γίνεται μια παρουσίαση όλων των τεχνολογιών που χρησιμοποιήθηκαν για την υλοποίηση μας.

4.1. IIS Web Server

Ο IIS (Internet Information Server) είναι ένα σύνολο από εξυπηρετητές (servers) διαδικτύου ανεπτυγμένος από την Microsoft και αποτέλεσε την πλέον

ανταγωνιστική απάντηση στον πολύ δημοφιλή apache server. Στο σύνολο των εξυπηρετητών που εμπεριέχονται στον IIS, προς το παρόν, συμπεριλαμβάνονται οι FTP, SMTP, NNTP και HTTP/HTTPS εξυπηρετητές.

Ο IIS προσφέρει επιπρόσθετες δυνατότητες για τα λειτουργικά συστήματα των windows, στα οποία η Microsoft τον παρέχει ενσωματωμένο. Οι εκδόσεις του IIS server και τα αντίστοιχα λειτουργικά συστήματα στα οποία έχει ενσωματωθεί φαίνονται παρακάτω.

- IIS 1.0, Windows NT 3.51
- IIS 2.0, Windows NT 4.0
- IIS 3.0, Windows NT 4.0 Service Pack 3
- IIS 4.0, Windows NT 4.0 OOption Pack
- IIS 5.0, Windows 2000
- IIS 5.1, Windows XP Professional, Windows MCE
- IIS 6.0, Windows Server 2003 and Windows XP Professional x64 Edition
- IIS 7.0, Windows Server 2008 and Windows Vista and Windows 7 (Pre-beta)

Μαζί με τον IIS η Microsoft παρέχει και ένα σύνολο από προγράμματα και εργαλεία για την κατασκευή και την διαχείριση ιστοσελίδων καθώς και για την επικοινωνία διαδικτυακών εφαρμογών με βάσεις δεδομένων.

Στα μειονεκτήματα του IIS αναφέρεται η ευπάθειά του, κυρίως των εκδόσεων 4.0 και 5.0, στις επιθέσεις ιών όπως ο Code Red και ο Nimda. Παρ'όλ' αυτά οι νεότερες εκδόσεις 6.0 και 7.0 έχουν αποδειχθεί ιδιαίτερα ασφαλείς. Επίσης στην έκδοση 6.0 προστέθηκε ένα καινούριο χαρακτηριστικό σε σχέση με τις παλαιότερες το επονομαζόμενο 'web server extensions' το οποίο αποτρέπει τον IIS από το να εκκινήσει οποιοδήποτε πρόγραμμα χωρίς την άδεια από τον διαχειριστή.

4.2. .NET Framework και Web forms

Η Microsoft δημιούργησε μια νέα πλατφόρμα, την οποία ονομάζει .NET, προκειμένου να δώσει ώθηση στις κατανεμημένες εφαρμογές της επόμενης γενιάς. Η πλατφόρμα

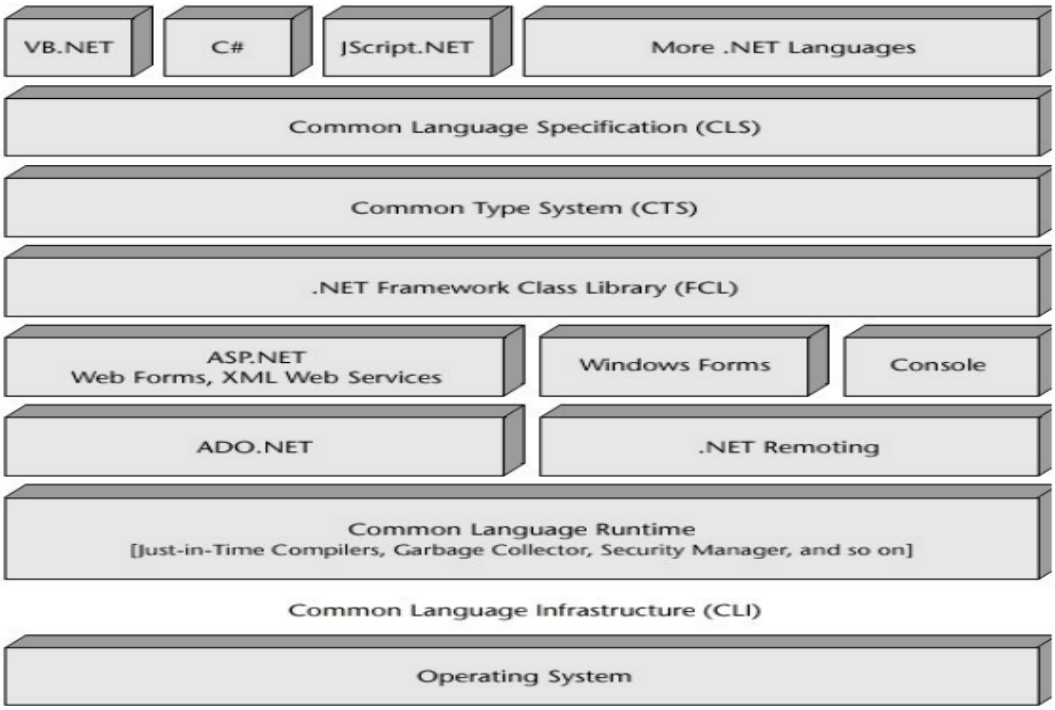
αυτή αποτελείται από δύο βασικές συνιστώσες, την υποδομή που πρέπει να ενσωματώνει ένας διακομιστής, και τις εφαρμογές που εκτελούνται και προσδιορίζουν τις λειτουργίες που έχει ο χρήστης στη διάθεσή του.

Η υποδομή (.NET Framework) διευκολύνει τις εφαρμογές να επικοινωνήσουν με το χρήστη, να αποθηκεύσουν και να εμφανίσουν στοιχεία από την βάση δεδομένων. Συγκεκριμένα η Microsoft .NET Framework είναι μια τεχνολογία λογισμικού η οποία είναι διαθέσιμη σε διάφορα λειτουργικά συστήματα Microsoft Windows. Είναι η καρδιά των .NET τεχνολογιών και συμπεριλαμβάνει μια ευρεία βιβλιοθήκη έτοιμων κλάσεων (Framework Class Library, FCL) και μια εικονική μηχανή η οποία διαχειρίζεται την εκτέλεση προγραμμάτων γραμμένων ειδικά για το Framework (Common Language Runtime, CLR). Οι λεπτομέρειες του .NET Framework δίνονται στο Common Language Specification (CLS), που παρέχει πληροφορίες μεταξύ άλλων για την αποθήκευση τύπων δεδομένων.

Τα προγράμματα μεταγλωττίζονται μέσω του .NET Framework σε δύο βήματα. Αρχικά το πρόγραμμα μεταγλωττίζεται σε ενδιάμεσο κώδικα (bytecode) γραμμένο σε Microsoft Intermediate Language (MSIL), ο οποίος ουσιαστικά ορίζει οδηγίες για το CLR που λειτουργεί ως μια εικονική μηχανή (virtual machine). Στην συνέχεια, ένας άλλος μεταγλωττιστής στο CLR μεταφράζει τη MSIL σε κώδικα μηχανής για μια συγκεκριμένη πλατφόρμα, δημιουργώντας έτσι μια μόνο εφαρμογή.

Η αρχιτεκτονική αυτή των δύο βημάτων παρέχει διαλειτουργικότητα ανάμεσα στις γλώσσες και έλεγχο πάνω στα χαρακτηριστικά διαχείρισης της εκτέλεσης, όπως στη διαχείριση μνήμης και την ασφάλεια. Η αρχιτεκτονική του .NETFramework παρουσιάζεται λεπτομερώς στο επόμενο σχήμα (Σχήμα2.1):

Σχήμα4.1



Η δημιουργία μιας εφαρμογής για το διαδίκτυο γίνεται με τον ίδιο τρόπο, μέσα από το Visual Studio .NET, επιλέγοντας το ASP .NET Web Application αντί του Windows Application, το οποίο χρησιμοποιείται για τις συνηθισμένες παραθυρικές εφαρμογές
Η δεύτερη συνιστώσα της πλατφόρμας απαιτεί την εγκατάσταση και τη χρήση του Visual Studio .NET, το οποίο επιτρέπει τη δημιουργία εφαρμογών και υπηρεσιών που εκτελούνται σε ένα διακομιστή. Ένα παράδειγμα είναι η νέα μορφή των σελίδων ASP (Active Server Pages), που ονομάζονται ASP .NET. Η νέα αυτή έκδοση απαιτεί πολύ λιγότερο κώδικα. Αντί για κώδικα VBScript μέσα σε εντολές HTML, μπορεί κανείς να χρησιμοποιήσει έναν πιο αποδοτικό κώδικα Basic, που περιλαμβάνει όλα τα χαρακτηριστικά της Visual Basic, καθώς και τις νέες επεκτάσεις της γλώσσας όπως είναι ο αντικειμενοστραφής προγραμματισμός.

Ουσιαστικά το πλαίσιο εφαρμογών .NET (.NET Framework) αποτελεί μία βασική τεχνολογία για την ανάπτυξη εφαρμογών ASP.NET. Περιλαμβάνει τις βασικές υπηρεσίες συστήματος που υποστηρίζουν την ASP.NET, όπως επίσης και την ανάπτυξη φορμών.

Τα Web Form είναι μια νέα μέθοδο για να δημιουργήσουμε εφαρμογές στο διαδίκτυο. Ουσιαστικά εξαλείφει το κενό μεταξύ της ανάπτυξης αυτόνομων εφαρμογών για τα λειτουργικά συστήματα και για το διαδίκτυο, καθώς χρησιμοποιεί τα ίδια εργαλεία,

κενές φόρμες, μια εργαλειοθήκη και κώδικα σε Visual Basic (ή οποιαδήποτε άλλη γλώσσα). Απαραίτητη προϋπόθεση είναι η ύπαρξη του αντίστοιχου διακομιστή της πλατφόρμας .NET .

Μια φόρμα Web Form αποτελείται από δύο μέρη. Το αρχείο HTML που αντιπροσωπεύει την οπτική απεικόνιση της σελίδας (όπως θα τη δει ο τελικός χρήστης στο PC) και ένα αρχείο με κώδικα σε Visual Basic που χειρίζεται τα γεγονότα, τα αντικείμενα και τα αποτελέσματα στη συγκεκριμένη σελίδα. Ο διακομιστής ανταλλάσσει στοιχεία και δεδομένα σε μορφή XML μέσα από τις φόρμες Web Form και εμφανίζει τα αποτελέσματα σε μορφή HTML.

4.3. ASP.NET

Το ASP.NET είναι έλα Framework το οποίο βασίζεται στο Microsoft.NET Framework και χρησιμοποιείται για την δημιουργία διαδικτυακών τόπων, δικτυακών εφαρμογών και υπηρεσιών. Κυκλοφόρησε από την Microsoft τον Ιανουάριο του 2002 και αποτελεί διάδοχο της παλαιότερης τεχνολογίας ASP. Βασίζεται και αυτό στο CLR .

Η τεχνολογία ASP (Active Server Pages) είναι ένα προγραμματιστικό περιβάλλον το οποίο παρέχει την δυνατότητα να συνδυαστούν μαζί html, wml, xml, scripting και components για την δημιουργία ισχυρών διαδικτυακών εφαρμογών. Χρησιμοποιόντας ASP.NET, μπορούμε να δημιουργήσουμε ιστοσελίδες που περιλαμβάνουν δυναμικές web pages. Η χρησιμοποίηση των δυνατοτήτων του ASP.NET προϋποθέτει την εγκατάσταση του εξυπηρετητή Internet Information Services (IIS), έκδοσης νεώτερης της 3.0.

Υπάρχει ένας διαχωρισμός μεταξύ της παρουσίασης και της λογικής του προγράμματος, αφού για κάθε μια HTML σελίδα που περιέχει τα ASP.NET controls, μπορεί να υπάρχει μια αντίστοιχη σελίδα με τον κώδικα της σελίδας, δηλαδή τι πρέπει να γίνει όταν ο χρήστης κάνει κλικ σε κάποιο button ή όταν φορτώσει η σελίδα. Αυτή η σελίδα με τον κώδικα μπορεί να είναι σε C# ή Virsual Basic .NET ή κάποια άλλη γλώσσα προγραμματισμού που υποστηρίζεται από το ASP.NET και ονομάζεται σελίδα “code-behind”.

Με την τεχνολογία αυτή επιτυγχάνουμε την εκτέλεση ενός κώδικα στην πλευρά του server πριν ακόμα φορτωθεί στον υπολογιστή του χρήστη η παραγόμενη ιστοσελίδα (server-side scripting). Η γλώσσα προγραμματισμού ASP γράφεται μαζί με τον HTML κώδικα και ξεχωρίζει με τα tags <% και %>, ενώ η ιστοσελίδα αποθηκεύεται με επέκταση .asp ή και .aspx.

Το ASP.NET βοηθά πολύ τον έλεγχο της εισόδου με τα διάφορα στοιχεία ελέγχου (Validation Controls) που έχει. Ακόμη υποστηρίζει master page . Μια master page είναι μια απλή ASP.NET σελίδα η οποία περιέχει μια σειρά από κανονικά στοιχεία τα οποία θα εμφανίζονται σε όλες τις άλλες σελίδες που την χρησιμοποιούν ως template. Ένα άλλο χαρακτηριστικό είναι το Active Calendar, το οποίο είναι ένα ASP.NET server control το οποίο επιτρέπει στους χρήστες να επιλέγουν γρήγορα, μια ημερομηνία και/ή ώρα χρησιμοποιώντας ένα επαγγελματικής εμφάνισης date picker με σύστημα διαχείρισης γεγονότων, client-side. Τέλος υποστηρίζει το GridView το οποίο είναι ένας πίνακας που τις περισσότερες φορές εμφανίζει δεδομένα από μια σχεσιακή βάση δεδομένων. Συνήθως γίνεται η συσχέτιση του GridView με μια συγκεκριμένη ερώτηση προς τη βάση, που ονομάζεται DataSource. [13]

4.4. HTML

Το 1990 ο Tim Berners-Lee από το Cern, το εργαστήριο φυσικής της Γενεύης, δημιούργησε ένα νέο πρωτόκολλο με το οποίο θα μπορούσαν να μεταφέρονται κάθε είδος αρχείων και αντικειμένων μέσα από το διαδίκτυο. Το πρωτόκολλο αυτό ονομάστηκε HTTP (HyperText Transfer Protocol) και σηματοδότησε την αρχή του WWW όπως το ξέρουμε σήμερα. Οι σελίδες που ήταν η βάση του WWW ήταν γραμμένες στην πρώτη έκδοση της γλώσσα HTML.

Το 1994 αναπτύχθηκε το πρότυπο HTML 2.0 από ένα διεθνή οργανισμό (Internet Engineering Task Forse). Η επόμενη έκδοση η 3.0 δεν έγινε αποδεκτή από τις Microsoft και Netscape οπότε γρήγορα αντικαταστάθηκε από την έκδοση 3.2 (1996). Η τελευταία περιελάμβανε πολλές από τις σημάνσεις (tags) που είχαν εισάγει οι δύο εταιρίες. Η έκδοση 4.0 παρουσιάστηκε τον Ιούνιο του 1997.

Η html είναι βασικά μία γλώσσα προγραμματισμού. Χρησιμοποιείται για την σήμανση ενός τμήματος κειμένου και για να το κάνει να εμφανίζεται καλύτερα. Επιτρέπει την ενσωμάτωση ήχου και εικόνων στις web σελίδες.

Η γλώσσα χρησιμοποιεί ένα αριθμό από tags για την μορφοποίηση κειμένου, για την δημιουργία συνδέσμων (links) μετάβασης ανάμεσα στις σελίδες, για την εισαγωγή εικόνων, ήχου κ.α. Ένα οποιοδήποτε κείμενο αποτελείται από επιμέρους στοιχεία όπως πίνακες παραγράφους, λίστες κ.α. Σε ένα HTML κείμενο πρέπει να μαρκάρεις την αρχή και το τέλος κάθε τέτοιου στοιχείου γράφοντας την κατάλληλη εντολή. Κάθε εντολή αρχίζει με το σύμβολο < και τελειώνει με το > δηλαδή μια εντολή είναι κάπως έτσι <εντολή> . Για να δηλώσουμε το τέλος της ισχύς μιας εντολής προσθέτουμε στην εντολή μια κάθετο, δηλαδή </εντολή>.

Όταν ένας Web Browser ανοίγει ένα αρχείο HTML τα στοιχεία (tags) μεταφράζονται σε κατάλληλα χαρακτηριστικά με αποτελέσματα στην εμφάνιση και στην λειτουργικότητα της συγκεκριμένης σελίδας.

4.5. Visual Studio

Το σύστημα ανάπτυξης Microsoft Visual Studio είναι ένα ισχυρό και πλήρως προσαρμοζόμενο περιβάλλον εργασίας, το οποίο περιλαμβάνει όλα τα εργαλεία που θα χρειαστούμε για να κατασκευάσουμε τόσο σταθερά προγράμματα για τα Microsoft Windows όσο και για τον ιστό, γρήγορα και αποτελεσματικά. Έχει σχεδιαστεί για να ευκολύνει τους προγραμματιστές (αρχάριους ή έμπειρους) να αντιμετωπίζουν πολύπλοκες προκλήσεις και να αναπτύσσουν καινοτόμες λύσεις.

Καθημερινά οι προγραμματιστές λογισμικού βρίσκονται αντιμέτωποι με δυσκολίες προκειμένου να αναπτύξουν λογισμικό το οποίο βοηθά στην καλυτέρευση των ζωών των άλλων. Ο ρόλος του Visual Studio είναι να βελτιώσει την διεργασία του προγραμματισμού και να κάνει πιο εύκολη και ικανοποιητική την επίτευξη των καινοτομιών. Τα εργαλεία του παρέχουν συνεχώς καλύτερες λύσεις στους προγραμματιστές έτσι ώστε να κάνουν περισσότερα πράγματα χωρίς να χάνουν χρόνο σε επαναλήψεις και ασήμαντες λεπτομέρειες.

Τα περισσότερα χαρακτηριστικά του περιβάλλοντος ανάπτυξης ισχύουν για όλες τις γλώσσες που προσφέρει το Visual Studio, δηλαδή τη Visual Basic, τη Visual C++, τη Visual C#, και τη Visual J#.

Τέλος, μας προσφέρει αξιοπιστία αφού είναι κατασκευασμένο και δοκιμασμένο προκειμένου να είναι συνεχώς αξιόπιστο, ασφαλές, εφαρμόσιμο και συμβατό με τις εφαρμογές. Παρόλο που το Visual Studio ενσωματώνει προοδευτικά στοιχεία, είναι κατασκευασμένο έτσι ώστε να υπάρχει εναρμόνιση με προηγούμενες εκδόσεις, όποτε αυτό είναι δυνατόν.[15]

4.5.1. Visual Studio 2005

Το Visual Studio 2005 κυκλοφόρησε τον Οκτώβρη του 2005. Η Microsoft αφαίρεσε την ονομασία ".NET" από το Visual Studio 2005 (όπως επίσης από κάθε άλλο προϊόν με την κατάληξη .NET στο όνομά του), αλλά ακόμα βασίζεται στην τεχνολογία .NET Framework, η οποία αναβαθμίστηκε στην έκδοση 2.0.

Το Visual Studio 2005 αναβαθμίστηκε προκειμένου να υποστηρίζει όλα τα νέα χαρακτηριστικά που εισήχθησαν στην .NET Framework 2.0, συμπεριλαμβανομένης και της ASP.NET 2.0. Το χαρακτηριστικό IntelliSense του Visual Studio αναβαθμίστηκε και νέου τύπου project προστέθηκαν για να υποστηρίξουν τις υπηρεσίες web της ASP.NET. Το Visual Studio 2005 περιλαμβάνει επίσης έναν τοπικό web server ο οποίος μπορεί να χρησιμοποιηθεί για να φιλοξενηθούν ASP.NET εφαρμογές κατά την ανάπτυξη και δοκιμή τους.

Επίσης υποστηρίζει όλες τις βάσεις δεδομένων του SQL Server 2005. Άλλα νέα γνωρίσματα του Visual Studio 2005 περιλαμβάνουν το "Deployment Designer", το οποίο επιτρέπει σχέδια εφαρμογών να επικυρωθούν πριν τεθούν σε εφαρμογή, ένα βελτιωμένο περιβάλλον για δημοσίευση ιστοσελίδων σε συνδυασμό με την ASP.NET 2.0 και ένα περιβάλλον για την πραγματοποίηση δοκιμών από τον χρήστη προκειμένου να παρατηρήσει την απόδοση των εφαρμογών.

4.5.2. ADO.NET

Το Visual Studio 2005 δεν έχει σχεδιαστεί για την δημιουργία νέων βάσεων δεδομένων, αλλά για την εμφάνιση, την ανάλυση, και το χειρισμό των πληροφοριών από υπάρχουσες βάσεις δεδομένων. Για αυτό το σκοπό το Visual Studio 2005 προσφέρει ένα βελτιωμένο μοντέλο δεδομένων, που ονομάζεται ADO.NET 2.0, το οποίο μπορεί να χρησιμοποιηθεί με ακόμα μεγαλύτερο αριθμό μορφών βάσεων δεδομένων και σεναρίων προσπέλασης. Συγκεκριμένα, το ADO.NET 2.0 έχει σχεδιαστεί για χρήση στο Internet.

Η εσωτερική μορφή δεδομένων του ADO.NET είναι XML. Η χρήση της XML κάνει ευκολότερη τη χρήση του με υπάρχουσες προελεύσεις δεδομένων του Internet, και απλοποιεί τη συγγραφή προσαρμογέων, ή παροχέων, από κατασκευαστές λογισμικού για τη μετατροπή άλλων μορφών βάσεων δεδομένων ώστε να είναι συμβατές με το ADO.NET.

Στο ADO.NET χρησιμοποιούνται διάφορα αντικείμενα για την ανάκτηση και την τροποποίηση πληροφοριών από μια βάση δεδομένων. Πρώτα γίνεται σύνδεση , η οποία καθορίζει τις πληροφορίες σύνδεσης για την βάση δεδομένων και δημιουργεί κάτι πάνω στο οποίο δεσμεύονται άλλα χειριστήρια και συστατικά στοιχεία. Στη συνέχεια, ο Οδηγός Διευθέτησης Προελεύσεων Δεδομένων δημιουργεί ένα σύνολο δεδομένων που είναι μια αναπαράσταση ενός ή περισσοτέρων πινάκων της βάσης τους οποίους θα χρησιμοποιήσουμε. [3]

4.5.3. Visual Web Developer

Με το εργαλείο αυτό της Visual Studio 2005 κατασκευάζουμε τοποθεσίες Ιστού και ιστοσελίδες. Το Visual Web Developer έχει την εμφάνιση και αίσθηση του περιβάλλοντος ανάπτυξης του Visual Studio, αλλά έχει προσαρμοστεί για προγραμματισμό στον Ιστό με την Microsoft ASP.NET 2.0. [3]

4.6. Βάση Δεδομένων

Η Microsoft Access μπορεί να είναι μια καλή επιλογή βάσεων δεδομένων όταν χρειάζεστε ένα σχετικά small-footprint, και μια φτηνή επιλογή βάσης δεδομένων. Ωστόσο, η απόδοση όταν έχουμε να κάνουμε με μεγάλες βάσεις δεδομένων και για πολλούς ταυτόχρονους χρήστες δεν είναι τόσο καλή. Χρησιμοποιώντας την Access σαν server-side database ακόμα και για μικρές βάσεις δεδομένων θα έχουμε χαμηλή απόδοση. Αυτό συμβαίνει αφού ο client πάντα πρέπει να διαβάσει ολόκληρο τον πίνακα από το διακομιστή αρχείων (ο υπολογιστής στον οποίο βρίσκεται η βάση μας).

Από την άλλη η Microsoft SQL Server έχει καλή απόδοση για μεγάλα footprint. Ακόμη έχει καλή απόδοση για medium-sized databases, και έχει server component το οποίο σχεδιάστηκε για να λειτουργεί πολύ αποτελεσματικά με δίκτυα. Υποστηρίζει επίσης stored procedures που μπορούν να εκτελέσουν σημαντικά καλύτερα από τα Access's queries. Η SQL έχει ένα πολύ πλουσιότερο και πιο γερό πρότυπο ασφάλειας από την Access. Τέλος, ο SQL Server επιτρέπει maintenance operations, αφού ενημερώνει ένα αρχείο συναλλαγής που μπορεί να χρησιμοποιηθεί για να ανακτήσει μια αλλοιωμένη βάση δεδομένων χωρίς απώλεια στοιχείων. [15]

4.6.1. SQL

Η πιο γνωστή γλώσσα που χρησιμοποιείται σε Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) είναι η SQL, της οποίας το όνομα προέρχεται από το Structured Query Language (Δομημένη Γλώσσα Ερωτήσεων).

Αρχικά η SQL ονομαζόταν SEQUEL (από το Structured English Query Language, Γλώσσα Ερωτήσεων Δομημένων Αγγλικών) και σχεδιάστηκε και υλοποιήθηκε στο ερευνητικό τμήμα της IBM ως διεπαφή για ένα πειραματικό σχεσιακό σύστημα βάσεων δεδομένων που ονομαζόταν SYSTEM R. Σήμερα η γλώσσα SQL αποτελεί μαζί με την QUEL, μια από τις πρώτες υψηλού επιπέδου γλώσσες βάσεων δεδομένων.

Η SQL είναι μια πλήρης γλώσσα βάσεων δεδομένων. Διαθέτει εντολές για ορισμό δεδομένων, ερωτήσεις και ενημερώσεις. Επομένως είναι ταυτόχρονα Γλώσσα Ορισμού

Δεδομένων αλλά και Γλώσσα Χειρισμού Δεδομένων. Επιπλέον, περιλαμβάνει ευκολίες για τον ορισμό όψεων στη βάση δεδομένων, για τη δημιουργία και διαγραφή ευρετηρίων πάνω σε αρχεία που παριστάνουν σχέσεις, και για την εμφύτευση εντολών της SQL σε γενικής χρήσης γλώσσες προγραμματισμού όπως η C ή η Pascal ή η Php ή η Visual Basic.

Η SQL χρησιμοποιήθηκε κυρίως στις εφαρμογές της Visual Basic για την λήψη από και εγγραφή δεδομένων προς τη βάση δεδομένων της εφαρμογής.

4.6.2. SQL Server 2005

Ο SQL Server 2005, κυκλοφόρησε τον Οκτώβριο του 2005, ως διάδοχος του SQL Server 2000. Περιέχει δική του ενσωματωμένη υποστήριξη διαχείρισης δεδομένων XML, επιπρόσθετα των σχεσιακών δεδομένων.

Γι' αυτό τον σκοπό, όρισε ένα τύπο δεδομένων XML ο οποίος μπορεί να χρησιμοποιηθεί είτε ως τύπος δεδομένου σε στήλες βάσεων δεδομένων ή ως literals στα ερωτήματα. Τα αποθηκευμένα XML δεδομένα επαληθεύονται μέσω της παράστασης. Η XML μετατρέπεται σε ένα εσωτερικό δυαδικό τύπο δεδομένων προτού αποθηκευτεί στην βάση δεδομένων. Εξειδικευμένες μέθοδοι για indexing έγιναν διαθέσιμες για τα XML δεδομένα.

Τα XML δεδομένα καλούνται χρησιμοποιώντας την XQuery – ο SQL server 2005 πρόσθεσε κάποιες επεκτάσεις στην γλώσσα T-SQL προκειμένου να επιτρέψει την ενσωμάτωση των XQuery ερωτημάτων στην T-SQL. Επιπλέον, καθορίζει μια νέα επέκταση στην XQuery, την αποκαλούμενη XML DML, η οποία επιτρέπει μετατροπές σε query-based XML δεδομένα.

Ο SQL server 2005 επίσης επιτρέπει σε κάποιο server βάσης δεδομένων να δημοσιευτεί (exposed) μέσω web υπηρεσιών χρησιμοποιώντας πακέτα TDS «κρυμμένα» μέσα σε SOAP ερωτήματα. Όταν τα δεδομένα προσπελαύνονται από υπηρεσίες web, τα αποτελέσματα μετατρέπονται σε XML. Για σχεσιακά δεδομένα,

η T-SQL έχει εμπλουτιστεί με διάφορα χαρακτηριστικά (features) για αντιμετώπιση λαθών και υποστήριξη για recursive ερωτήματα.

Η ρύθμιση των αδειών και της πρόσβασης έχουν γίνει πιο λεπτομερειακά και ο επεξεργαστής των ερωτημάτων χειρίζεται την ταυτόχρονη εκτέλεση ερωτημάτων με ένα πιο αποδοτικό τρόπο. Οι διαμερίσεις (partitions) στους πίνακες και τους δείκτες υποστηρίζονται εσωτερικά (natively), έτσι η κλιμάκωση (scaling) μιας βάσης δεδομένων σε ένα cluster είναι ευκολότερη. Ο SQL CLR εμφανίστηκε με τον SQL server 2005 και επέτρεψε την ενοποίηση με την .NET Framework.

4.7. Visual Basic 2005

Η εφαρμογή μας θα αναπτυχθεί με χρήση της γλώσσας Visual Basic 2005. Η Visual Basic ακολούθησε εντυπωσιακή εξέλιξη την τελευταία δεκαετία, από την έκδοση 1.0 στην έκδοση .NET, ή 7.0, όπως είναι η εσωτερική αρίθμηση του προϊόντος μέχρι και την έκδοση Visual Basic 2005. Η Visual Basic 2005 είναι σημαντική αναβάθμηση και βελτίωση του δημοφιλούς συστήματος προγραμματισμού Visual Basic, και ένα βήμα πιο πέρα στο λογισμικό Visual Basic .Net 2003.

Η έκδοση Visual Basic 2005 έχει αρκετές σημαντικές διαφορές και βελτιώσεις, στη γλώσσα και στο περιβάλλον προγραμματισμού. Οι βελτιώσεις στη γλώσσα περιλαμβάνουν νέες εντολές και σχήματα με επεκτάσεις της γλώσσας σε αντικειμενοστρεφή προγραμματισμό. Από τις πιο σημαντικές βελτιώσεις σε αυτό τον τομέα είναι η δυνατότητα της κληρονομικότητας (inheritance), της δημιουργίας κλάσεων (class) αλλά και της ενσωμάτωσης πιο αυστηρών κανόνων για την αντίδραση σε λάθη.

Οι νέες εντολές, αλλά και όσα εργαλεία προστέθηκαν μέχρι τη σημερινή έκδοση, δεν άλλαξαν τον αντικειμενικό σκοπό, που είναι η εύκολη και γρήγορη ανάπτυξη εφαρμογών. Έτσι μέσα από το Visual Studio, η Visual Basic, παραμένει ένα εύκολο και ιδιαίτερα γρήγορο εργαλείο για την ανάπτυξη αυτόνομων εφαρμογών για το λειτουργικό σύστημα Windows, φορητών συσκευών αλλά και υπηρεσιών στο Web. Όλα αυτά γίνονται εύκολα, με το σύστημα να κάνει για εμάς την περισσότερη δουλειά.

Κεφάλαιο 5

Ανάλυση της Υπάρχοντος Συστήματος

5.1. Εφαρμογή DITIS 40

5.2. Πίνακες Βάσης Δεδομένων 44

5.3. Διαχωρισμός του συστήματος σε ενότητες και ανάλυσή τους 47

Κύριο συστατικό της εφαρμογής είναι η βάση δεδομένων από την οποία παίρνει και αποθηκεύει τις πληροφορίες του το πρόγραμμα. Η βάση δεδομένων που χρησιμοποιούμε για την υλοποίηση του συστήματος μας είναι η βάση δεδομένων του συστήματος DITIS. Το σύστημα αυτό είναι σε εφαρμογή εδώ και κάτι χρόνια επομένως έχει δοκιμαστεί και μπορούμε να είμαστε σίγουροι για την αξιοπιστία της βάσης δεδομένων που βρίσκεται πίσω από το συγκεκριμένο σύστημα. Είναι γραμμένη σε Microsoft SQL Server 2005 και για να μπορέσουμε να υλοποιήσουμε την ιστοσελίδα έπρεπε να μελετηθεί με προσοχή η βάση που μας δόθηκε.

Στο κεφάλαιο αυτό υπάρχει μια μικρή αναφορά για το σύστημα DITIS ενώ ακόμη, θα δούμε κάποια βασικά στοιχεία της υφιστάμενης βάσης.

5.1. Εφαρμογή DITIS

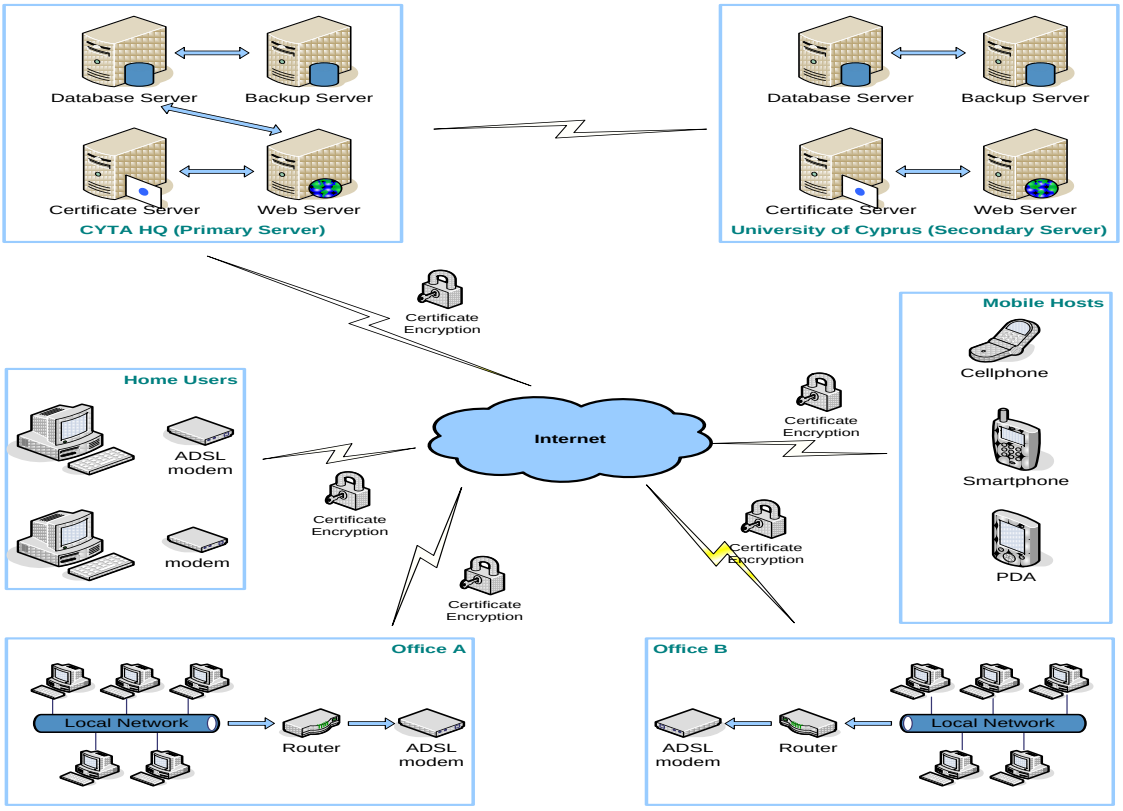
Η ανάπτυξη του συστήματος DITIS για την στήριξης και διευκόλυνσης του έργου τις ομάδας περίθαλψης ενός ασθενή καθώς και του ίδιου του ασθενή παρέχοντας ταυτόχρονα άμεση, εύκολη, αλλά και ασφαλή πρόσβαση στο ιατρικό αρχείο του ασθενή πρωτάρχισε το 1999. Το σύστημα έχει υποστηρίξει τις δραστηριότητες της ένωσης της Κύπρου των ασθενών με καρκίνο και των φίλων (PA.SY.KA.F.) που προσφέρει κατ' οίκον φροντίδα σε περισσότερους από 400 ασθενείς με καρκίνο το χρόνο στη Κύπρο.

Συγκεκριμένα μέχρι σήμερα έχουν υλοποιηθεί από ομάδα του πανεπιστημίου Κύπρου δυο συστήματα. Το πρώτο σύστημα(DITIS1) παρέχει μια ασφαλή πρόσβαση στις ιατρικές πληροφορίες από οποιαδήποτε κινητή συσκευή οποιαδήποτε ώρα και από οποιοδήποτε χώρο. Ο DITIS2 είναι μια εξέλιξη του DITIS1 και παρέχει πρόσβαση και μέσω των προσωπικών υπολογιστών. Βασικά είναι ένα λογισμικό το οποίο πρέπει να εγκατασταθεί αρχικά στην συσκευή. Περιλαμβάνει ένα σύνολο εργαλείων για την αποτελεσματική επικοινωνία και συντονισμό των μελών ομάδας, συμπεριλαμβανομένης της αυτόματης υπενθύμισης. Χρησιμοποιεί τα εργαλεία σχετικά με την οικιακή φροντίδα που βελτιώνουν την αποδοτικότητα και ελαχιστοποιούν τα λάθη του ελέγχου οικιακής φροντίδας.

5.1.1. Αρχιτεκτονική DITI

Στο πιο κάτω σχήμα(Σχήμα4.1) μπορούμε να δούμε συνοπτικά την αρχιτεκτονική που χρησιμοποιήθηκε στα προηγούμενα συστήματα DITIS.

Σχήμα5.1



Αρχικά να αναφέρουμε πως η CYTA είναι ο χειριστής τηλεπικοινωνιών που υποστηρίζει το έργο αυτό. Όπως βλέπουμε και από το σχήμα ο πρωτεύον database

server βρίσκεται στη CYTA HQ. Αυτός ο server κρατά (φυλάει) όλα τα δεδομένα που χρησιμοποιεί η εφαρμογή DITIS. Ακόμη εκεί υπάρχει ένας εφεδρικός server, ένας certificate server για την ασφάλεια και ένας web-server που επιτρέπει την πρόσβαση μέσω των κινητών ή browser-enabled συσκευών.

Για ασφάλεια όμως (αν τυχόν πέσει ο server στην CYTA) για να μην πέσει και το σύστημα μας, η ίδια αρχιτεκτονική υπάρχει και στο πανεπιστήμιο της Κύπρου.
Τώρα για να μπορεί ένας χρήστης να δει τα διάφορα δεδομένα της βάσης πρέπει να ενωθεί με τους πιο πάνω server. Αυτό γίνεται εύκολα και με ασφάλεια αφού όλες οι συσκευές DITIS συνδέονται μέσω του Διαδικτύου χρησιμοποιώντας certificate encryption. Κατά αυτόν τον τρόπο οι διαφορετικοί τύποι χρηστών (γραφείο, σπίτι και κινητοί χρήστες) μπορούν να συνδεθούν με την βάση δεδομένων DITIS. [5]

5.1.2. Διαπροσωπία που υπάρχουν στον DITI

Στα προηγούμενα συστήματα που σχεδιάστηκαν είχαμε τρεις διαφορετικούς τρόπους που μπορούσαμε να έχουμε πρόσβαση στην εφαρμογή μας.

Η πρώτη που κτίστηκε ήταν μια windows application η οποία μπορούσε να τρέξει σε έναν υπολογιστή γραφείου. Εξαιτίας όμως του γεγονότος ότι οι χρήστες χρειάζονταν την κινητικότητα στην εφαρμογή κτίστηκε το web module το οποίο μπορούσε να τρέξει από διάφορους browser. Κατά την χρησιμοποίηση των κινητών συσκευών εξαρτιόμασταν από αν είχαμε καλή σύνδεση από το δίκτυο. Για να απαλειφθεί η επίδραση των συχνών αποσυνδέσεων χτίστηκε το Mobile module που τρέχει μια mobile database πάνω στην κινητή συσκευή και συγχρονίζει τα δεδομένα μόνο όταν χρειάζεται. Ακολούθως θα δούμε πιο πολλές λεπτομέρειες για κάθε μια από τις τρεις διεπαφές. [5]

5.1.2.1. Desktop Interface

Η διεπαφή που σχεδιάστηκε για τους υπολογιστές γραφείων χαρακτηρίζεται από πλήρη λειτουργικότητα, είναι πολύμορφη και φιλική προς το χρήστη και τέλος από User customizability. Επομένως, είναι μια πλήρης υλοποίηση του συστήματος όπου μέσο από αυτήν εύκολα ένας χρήστης μπορεί να εκτελέσει όποια λειτουργία χρειάζεται. Το

μόνο μειονέκτημα είναι ότι ο χρήστης δεν μπορεί να το χρησιμοποιήσει ανά πάσα στιγμή οπουδήποτε και αν βρίσκεται.

5.1.2.2. Web-Mobile Interface

Για να απαλλαγούμε από το πιο πάνω πρόβλημα σχεδιάστηκε το Web-Mobile Interface. Ουσιαστικά μπορούσε να τρέξει από οποιαδήποτε μηχανή αναζήτησης υποστήριζε η συσκευή. Το interface σε αυτή την περίπτωση ήταν πολύ λιτό για να μπορεί να φορτωθεί πάνω στις κινητές συσκευές. Ακόμη δεν υποστηρίζονταν όλες οι λειτουργίες που έχουμε στο desktop αλλά όσες είναι απαραίτητες για την σωστή λειτουργία του οργανισμού. Σε αυτή την περίπτωση ο Server είναι υπεύθυνος για την ρύθμιση της διεπαφής πάνω σε κινητές συσκευές.

Δυστυχώς, όμως και σε αυτή την περίπτωση υπάρχουν προβλήματα. Αυτή την φορά όσον αφορά την συνδεσιμότητα, αφού εκεί που βρίσκεται κάποιος χρήστης μπορεί να να μην υπάρχει καλό σήμα με αποτέλεσμα να αποσυνδέεται συχνά.

5.1.2.3. Standalone Mobile/Smartphone Interface

Η τρίτη και τελευταία διεπαφή που δημιουργήθηκε είναι μια συνέχεια μπορούμε να πούμε της προηγούμενης. Με μόνη διαφορά ότι έχουμε ένα κάπως πιο φιλικό προς τον χρήστη interface και πάνω στην συσκευή μας τώρα υπάρχει και μια τοπική βάση. Η απευθείας σύνδεση με την βάση μας στον server γίνεται μόνο όταν χρειάζεται, για να έχουμε συντονισμό στα δεδομένα μας.

5.1.2.4. Web Interface (Υλοποίηση μας)

Το Web Interface είναι αυτό που καλούμαστε να σχεδιάσουμε και να υλοποιήσουμε σήμερα. Θα είναι μια εύχρηστη ιστοσελίδα όπου εύκολα ο κάθε χρήστης μας (έμπειρος σε προηγούμενα version, ή άπειρος) θα μπορεί να εκτελέσει οποιαδήποτε λειτουργία θελήσει.

Σίγουρα με την χρήση του διαδικτύου γνωρίζουμε ότι μπορούμε να έχουμε πρόσβαση στην εφαρμογή οποιαδήποτε ώρα σε οποιοδήποτε χώρο φτάνει να υπάρχει σύνδεση με το διαδίκτυο. Αυτό γίνεται πολύ εύκολα στις μέρες μας αφού όπως όλοι γνωρίζουμε πολύ καλά σχεδόν παντού μπορούμε να έχουμε πρόσβαση στο διαδίκτυο. Από το σπίτι

μας, την δουλειά, την καφετέρια, ακόμη και από το προσωπικό μας κινητό εικοσιτέσσερις ώρες το εικοσιτετράωρο όπου και αν βρισκόμαστε.

Χρησιμοποιώντας την ASP.NET μπορούμε κάνοντας απλά μια υλοποίηση να έχουμε μια εφαρμογή τόσο για desktop όσο και για κινητές συσκευές, χωρίς να χρειάζεται να κάνουμε άλλη σελίδα για κάθε συσκευή. Αυτό γίνεται πολύ εύκολα αφού η ASP.NET δίνει διαφορετική σήμανση για κάθε browser. Εάν η πρόσβαση στην σελίδα μας γίνεται μέσω κινητής συσκευής, η ASP.NET θα επιστρέψει τον κατάλληλο κώδικα HTML.

Ωστόσο για να γίνει αυτό σωστά πρέπει ο σχεδιαστής να είναι προσεκτικός με το μέγεθος της οθόνης κάθε χρήστη. Για να μην έχουμε προβλήματα όσον αφορά την σχεδίαση, καλό είναι να χωρίσουμε από την αρχή το λογικό με το σχεδιαστικό μέρος της εφαρμογής μας χρησιμοποιώντας css styles. Με αυτό τον τρόπο μπορούμε να είμαστε σίγουροι ότι η σελίδα μας φαίνεται καλά σε κάθε μέγεθος οθόνης.

Κάτι άλλο που πρέπει να προσέξουμε είναι τις ικανότητες που μπορεί να έχει κάθε web browser. Κάποιοι web browsers που χρησιμοποιούνται στις κινητές συσκευές δεν υποστηρίζουν javascript ή εικόνες ή cookies. Έτσι πρέπει να είμαστε προσεκτικοί στην υλοποίηση μας, αν θα τα χρησιμοποιήσουμε να ενημερώνουμε κατάλληλα τους χρήστες με συσκευές που δεν υποστηρίζουν τις τεχνικές αυτές.

5.2. Πίνακες Βάσης Δεδομένων

Η βάση δεδομένων αποτελείται από 73 πίνακες εκ των οποίων οι 30 από αυτούς αποτελούν πίνακες αναζήτησης (lookup tables) που δεν μπορούν να τροποποιηθούν από τους απλούς χρήστες αλλά βοηθούν στο να είναι η βάση σε συνεπή κατάσταση (database consistency). Κάποιοι από αυτούς παρατίθενται στην συνέχεια.

dbo.PROFESSION: περιέχει δεδομένα που αφορούν την ιδιότητα/επάγγελμα κάποιου χρήστη,
dbo.RELATIONSHIP: περιέχει δεδομένα που αφορούν την σχέση που μπορεί να υπάρχει μεταξύ δυο εμπλεκόμενων,

dbo.COUNTRIES: περιέχει όλες τις χώρες,

dbo.DRUGS: περιέχει τις ονομασίες όλων των φαρμάκων που χρησιμοποιούνται από τον οργανισμό

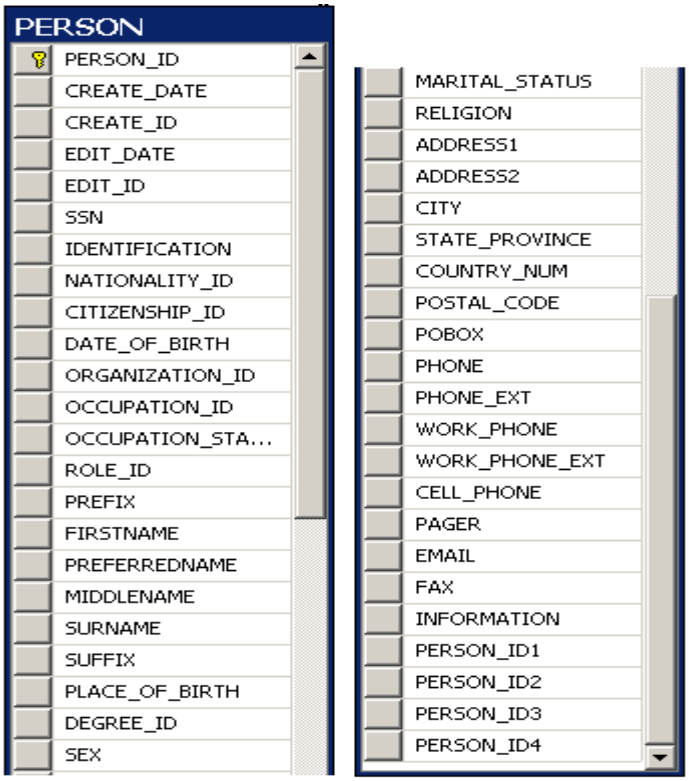
dbo.SERVICES: περιέχει την περιγραφή των υπηρεσίων που μπορούν να προσφερθούν σε ένα ασθενή,

dbo.SYMPTOMS: περιέχει συμπτώματα που μπορεί να έχει ένας ασθενής κ.α.

Δεν θα δούμε σε αυτά τα έγραφα αναλυτικά την βάση δεδομένων μας. Απλά σε αυτό το κεφάλαιο θα δούμε κάποιες από τις σημαντικότερες οντότητες τη βάσης που θεωρούμε ότι πρέπει κανείς να γνωρίζει για να μπορέσει να κατανοήσει τις διάφορες λειτουργίες του συστήματος.

Μια από της κυριότερες οντότητες είναι η dbo.PERSON (Πινάκας4.1). Σε αυτήν βρίσκονται διάφορα προσωπικά στοιχεία όλων των χρηστών του συστήματος όπως, όνομα, τηλέφωνο, διεύθυνση κ.α. Πιο κάτω μπορούμε να δούμε την οντότητα αυτή.

Πινάκας5.1.



Το πεδίο PERSON_ID είναι αυτό που διαχωρίζει τα άτομα μεταξύ τους. Είναι ένας μοναδικός αριθμός που ανατίθεται σε κάθε άτομο. Τα υπόλοιπα πεδία αποτελούν τα κύρια στοιχεία που χαρακτηρίζουν ένα άτομο. Με την οντότητα αυτή σχετίζονται δυο άλλες σημαντικές οντότητες, η οντότητα dbo.PATIENT και dbo.PERSONNEL. Ουσιαστικά κάθε πεδίο PERSON_ID αντιστοιχεί είτε σε ένα πεδίο PATIENT.ID της οντότητας dbo.PATIENT είτε σε ένα πεδίο PERSONNEL.ID της οντότητας dbo.PERSONNEL.

Επομένως, στην οντότητα PERSON_ID έχουμε όλα τα προσωπικά στοιχεία των χρηστών, είτε αυτός είναι ασθενής είτε αυτός είναι personnel. Και απλά στις άλλες δύο οντότητες φυλάγονται περισσότερα στοιχεία όσον αφορά τους ασθενείς και τους personnel αντίστοιχα. Αν το άτομό που έχει PERSON_ID ίσο με 55 είναι ασθενής αυτό σημαίνει ότι στον πίνακα dbo.PERSONNEL δεν υπάρχει PERSONNEL.ID ίσο με 55.

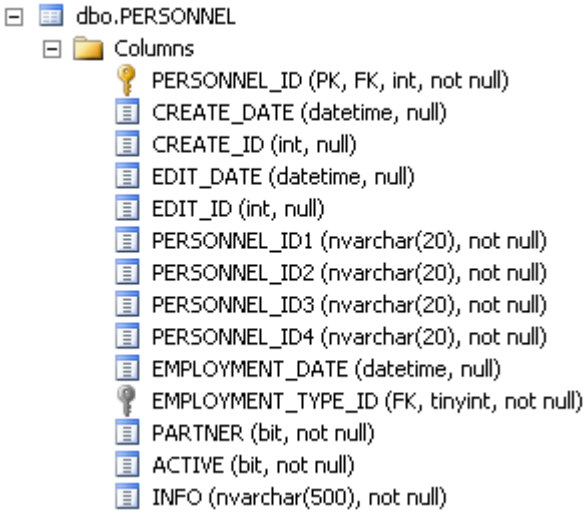
Πινάκας5.2

- dbo.PATIENT
  - Columns
    - PATIENT_ID (PK, FK, int, not null)
    - CREATE_DATE (datetime, null)
    - CREATE_ID (int, null)
    - EDIT_DATE (datetime, null)
    - EDIT_ID (int, null)
    - PATIENT_ID1 (nvarchar(20), not null)
    - PATIENT_ID2 (nvarchar(20), not null)
    - PATIENT_ID3 (nvarchar(20), not null)
    - PATIENT_ID4 (nvarchar(20), not null)
    - PATIENT_STATUS_ID (FK, tinyint, not null)
    - HOSPITAL_ID (FK, smallint, null)
    - REFUGEE (bit, not null)
    - REFUGEE_ORIGIN (nvarchar(50), not null)
    - ACTIVE (bit, not null)
    - RESTRICTED (bit, not null)
    - NUM_OF_CHILDREN (tinyint, not null)
    - VIP (bit, not null)
    - NEXT_OF_KIN_NAME (nvarchar(50), null)
    - NEXT_OF_KIN_PHONE (nvarchar(25), null)
    - NEXT_OF_KIN_WORK_PHONE (nvarchar(25), null)
    - NEXT_OF_KIN_CELL_PHONE (nvarchar(25), null)
    - NEXT_OF_KIN_ADDRESS (nvarchar(50), null)
    - NEXT_OF_KIN_RELATIONSHIP_ID (nchar(3), null)
    - MAIN_CARER_NAME (nvarchar(50), null)
    - MAIN_CARER_PHONE (nvarchar(25), null)
    - MAIN_CARER_WORK_PHONE (nvarchar(25), null)
    - MAIN_CARER_CELL_PHONE (nvarchar(25), null)
    - MAIN_CARER_ADDRESS (nvarchar(50), null)
    - MAIN_CARER_RELATIONSHIP_ID (nchar(3), null)
    - EMERGENCY_NAME (nvarchar(50), null)
    - EMERGENCY_PHONE (nvarchar(25), null)
    - EMERGENCY_WORK_PHONE (nvarchar(25), null)
    - EMERGENCY_CELL_PHONE (nvarchar(25), null)
    - EMERGENCY_ADDRESS (nvarchar(50), null)
    - EMERGENCY_RELATIONSHIP_ID (nchar(3), null)
    - EMPLOYER_NAME (nvarchar(50), null)
    - EMPLOYER_WORK_PHONE (nvarchar(25), null)
    - INSURANCE_NAME (nvarchar(50), null)
    - INSURANCE_NUMBER (nvarchar(30), null)
    - INSURANCE_WORK_PHONE (nvarchar(25), null)

Πιο πάνω βλέπουμε τα πεδία που περιέχει η οντότητα dbo.PATIENT.

Ακολουθούν τα πεδία της οντότητας dbo.PERSONELL

Πινάκας5.3

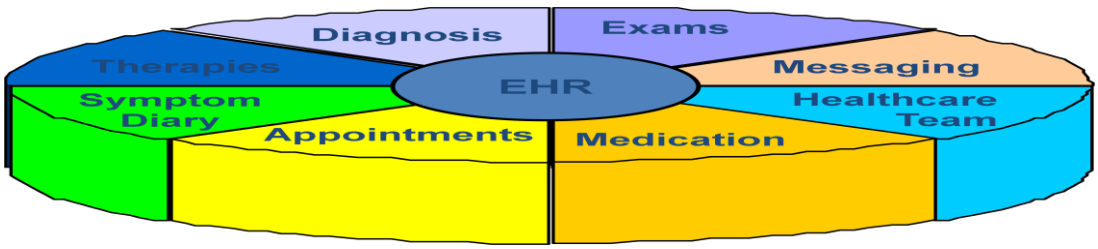


Πιο κάτω, καθώς θα δούμε τις διάφορες ενότητες (λειτουργίες) στις όποιες χωρίσαμε το σύστημα μας θα αναφερθούμε και σε κάποιες ακόμη σημαντικές οντότητες της βάσης μας.

5.3. Διαχωρισμός του συστήματος σε ενότητες και ανάλυσή τους

Το υπάρχον σύστημα DITIS μπορούμε να το χωρίσουμε σε 8 βασικές ενότητες (Σχήμα 4.2). Αυτές φαίνονται στο πιο κάτω σχήμα. Στην συνέχεια θα δούμε μια μικρή περιγραφή για τη κάθε μια και με ποιους πίνακες της βάσης μας πρέπει να αλληλεπιδράσουμε για να εκτελέσουμε πιθανές λειτουργίες. Περεταίρω πληροφορίες και λεπτομέρειες για τις οντότητες και τους πίνακες που σχετίζονται μπορούμε να βρούμε σε άλλα έγραφα.

Σχήμα5.2



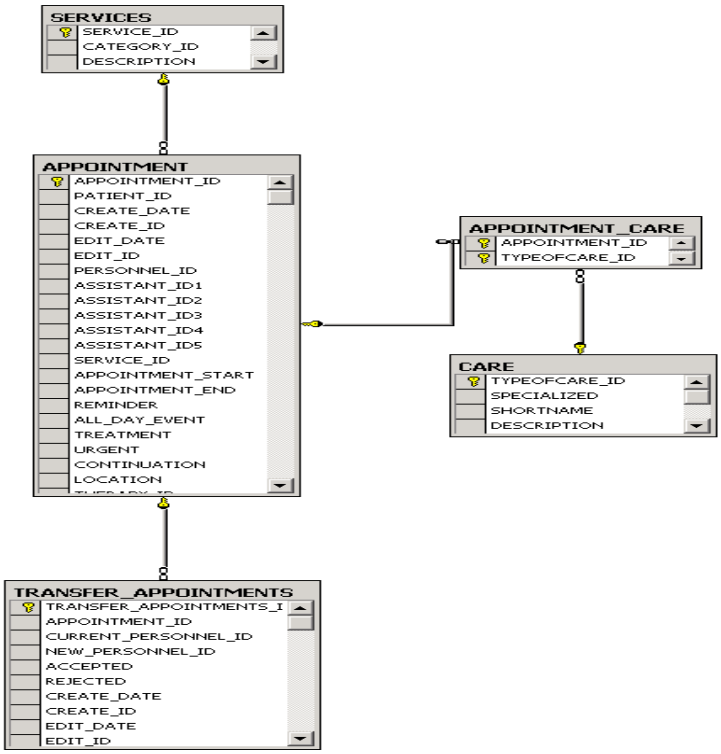
5.3.1. Appointment module

Το Appointment module είναι η αρμόδια λειτουργία για τη διαχείριση των διάφορων ραντεβού στον οργανισμό.

5.3.1.1. Entity Relationship Diagram

Όπως φαίνεται και στο πιο κάτω ERD τα appointment σχετίζονται με τους πίνακες της βάσΗς μας, dbo.APPOINTMENT, dbo.SERVICES (lookup table), dbo.CARE (lookup table), dbo.APPOINTMENT_CARE και dbo.TRANSFER_APPOINTMENTS.

Στον πίνακα dbo.APPOINTMENT φυλάγονται τα αρχεία των appointments. Στον πίνακα αναζήτησης dbo.SERVICES βρίσκεται μια λίστα από τις υπηρεσίες που μπορούν να προσφερθούν σε κάποιο ραντεβού. Ακόμη ο πίνακας αναζήτησης dbo.CARE περιέχει μια λίστα από την περιγραφή φροντίδας ενός ασθενή που μπορεί να προσφερθεί σε κάποιο ραντεβού. Σε κάθε ραντεβού μπορούν να προσφερθούν μηδέν ή και παραπάνω care services. Ενώ ο πίνακας dbo.APPOINTMENT_CARE δείχνει την σχέση many-to-many μεταξύ των πινάκων dbo.APPOINTMENT και dbo.CARE. Τέλος ο πίνακας TRANSFER_APPOINTMENTS χρησιμοποιείται για την μεταφορά των ραντεβού από ένα άτομο σε άλλο.



Σχήμα5.3

5.3.2. Medication module

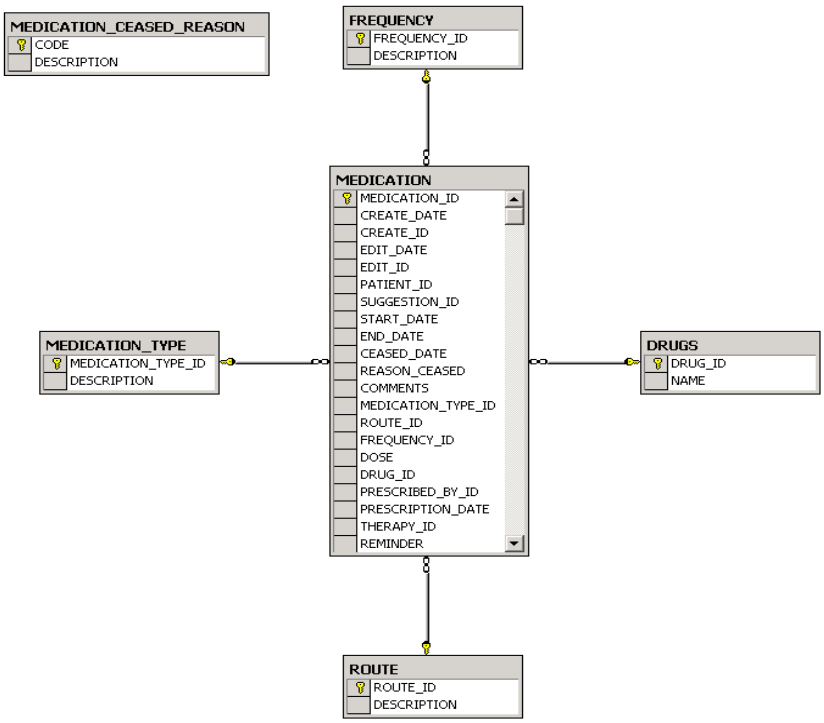
Το Medication module είναι η αρμόδια λειτουργία για τη διαχείριση φαρμάκων για κάθε ασθενή στον οργανισμό.

5.3.2.1. Entity Relationship Diagram

Σε αυτή την ενότητα εμπλέκονται οι πίνακες dbo.MEDICATION, dbo.DRUGS (lookup table), dbo.MEDICATION_TYPE (lookup table), dbo.ROUTE (lookup table), dbo.FREQUENCY (lookup table) και dbo.MEDICATION_CEASED_REASON (lookup table).

Στον πίνακα dbo.MEDICATION βρίσκονται όλες οι εγγραφές των medication. Ο πίνακας dbo.DRUGS είναι μια λίστα από φάρμακα που χρησιμοποιούνται από τον οργανισμό. Και ο πίνακας dbo.MEDICATION_TYPE είναι μια λίστα από περιγραφές medication type που περιγράφονται οι βασικοί τύποι medication. Ο πίνακας dbo.ROUTE περιέχει μια λίστα από περιγραφές που δείχνουν πως τα φάρμακα δόθηκαν στον ασθενή. Μια λίστα από τις πιθανές συχνότητές που πρέπει να παίρνουν οι ασθενείς κάθε φάρμακο βρίσκεται στον πίνακα dbo.FREQUENCY. ο τελευταίος πίνακας που συσχετίζεται είναι ο dbo.MEDICATION_CEASED_REASON ο οποίος περιέχει μια λίστα από περιγραφή τον βασικών τύπων.

Σχήμα5.4



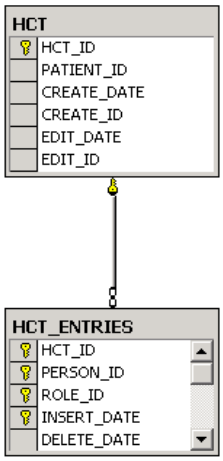
5.3.3. Health care teams' module

Η Health care teams' module είναι αρμόδια λειτουργία για τη διαχείριση των ομάδων περίθαλψης για κάθε ασθενή στον οργανισμό.

5.3.3.1. Entity Relationship Diagram

Όπως φαίνεται και από ERD αποτελείται από δυο πίνακες. Τον πίνακα dbo.HCT και τον πίνακα dbo. HCT_ENTRIES. Στο πρώτο πίνακα φυλάσσονται τα αρχεία της ομάδας περιθαλψης (health care team records) του κάθε ασθενή. Ενώ στον δεύτερο μπορεί να βρει κανείς τα μέλη για κάθε ομάδα.

Σχήμα5.5



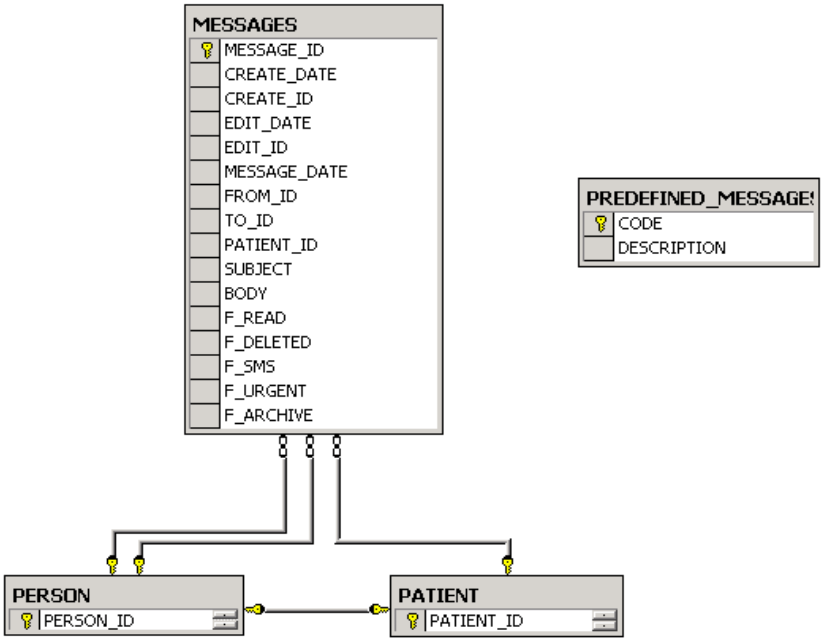
5.3.4. Messaging module

Το Messaging module είναι η αρμόδια λειτουργία για την επικοινωνία μεταξύ της ομάδας περίθαλψης και του ασθενή. Η διαχείριση των μηνυμάτων αντιμετωπίζεται με συγκεκριμένα flags που βρίσκονται στον πίνακα dbo.MESSAGES.

5.3.4.1. Entity Relationship Diagram

Αυτή η ενότητα περιλαμβάνει δυο πίνακες, τον dbo.MESSAGES που είναι υπεύθυνος για την διαχείριση των μηνυμάτων και ένα δεύτερο τον dbo.PREDEFINED_MESSEGES (lookup table) που περιέχει μια λίστα από προκαθορισμένα μηνύματα που μπορεί να επιλέξει ο χρήστης για να στείλει.

Σχήμα5.6



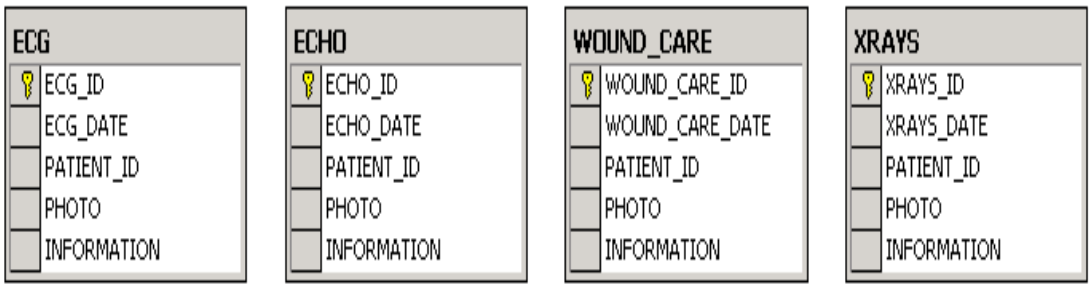
5.3.5. Exams' module

Το Exams' module είναι η αρμόδια λειτουργία για την αποθήκευση της ιατρικής εικόνας των ασθενών. Σε αυτήν περιλαμβάνονται η ECG, οι ακτίνες X, Echo κ.λπ.

5.3.5.1. Entity Relationship Diagram

Όλοι οι πίνακες που αφορούν εξετάσεις έχουν τον ίδιο σχεδιασμό. Συγκεκριμένα, ένα πεδίο για την αποθήκευση της εικόνας, ένα πεδίο για την ημερομηνία εγγραφής και τέλος ένα πεδίο όπου φυλάσσεται ο healthcare professional. Στην βάση του DITI υπάρχουν αυτή την στοιγμή τέσσερις τέτοιοι πίνακες, ο dbo.ECG, ο dbo.ECHO, ο dbo.WOUND_CARE και ο dbo.XRAYS ο καθένας για τις ανάλογες εξετάσεις.

Σχήμα5.7



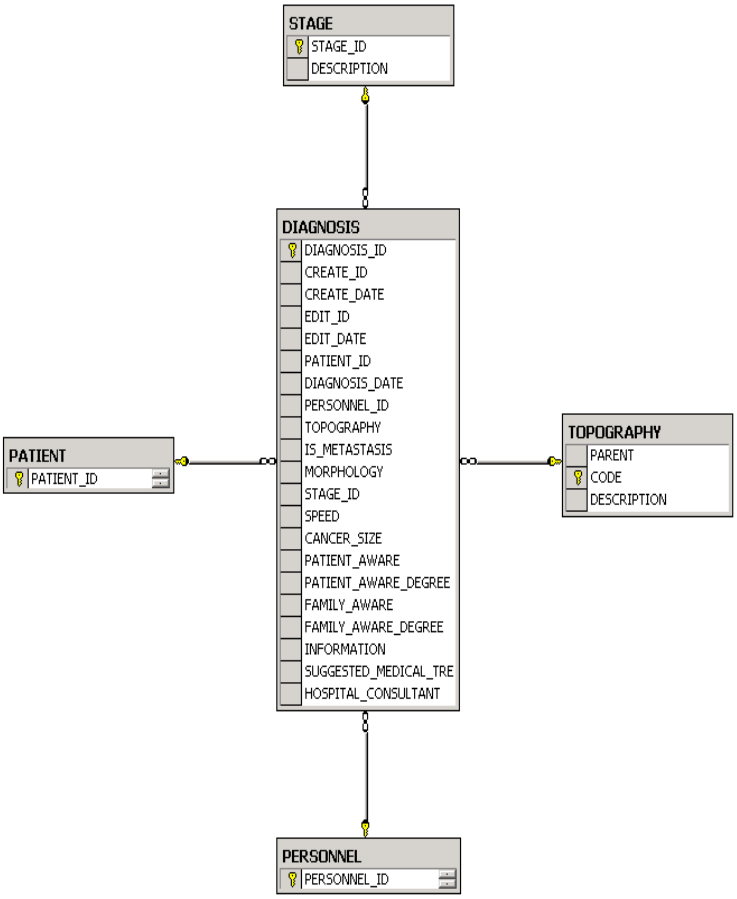
5.3.6. Diagnosis module

Η Diagnosis module είναι η αρμόδια λειτουργία για την καταγραφή των αρχείων διαγνώσεων για έναν ασθενή.

5.3.6.1. Entity Relationship Diagram

Με τις διαγνώσεις συσχετίζονται τέσσερις πίνακες. Ο dbo.DIAGNOSIS πίνακας που περιέχει τις εγγραφές για τις διάφορες διαγνώσεις, ο dbo.TOPOGRAPHY ο οποίος περιέχει πληροφορίες για την International Classification of Diseases (ICD version 10 AM), ο πίνακας dbo.STAGE (lookup table) ο οποίος είναι μια λίστα από περιγραφή τον σταδίων του καρκίνου και τέλος ο πίνακας dbo.MORPHOLOGY (lookup table) όπου και αυτός είναι μια λίστα από τις morphologies για καρκίνο.

Σχήμα5.8



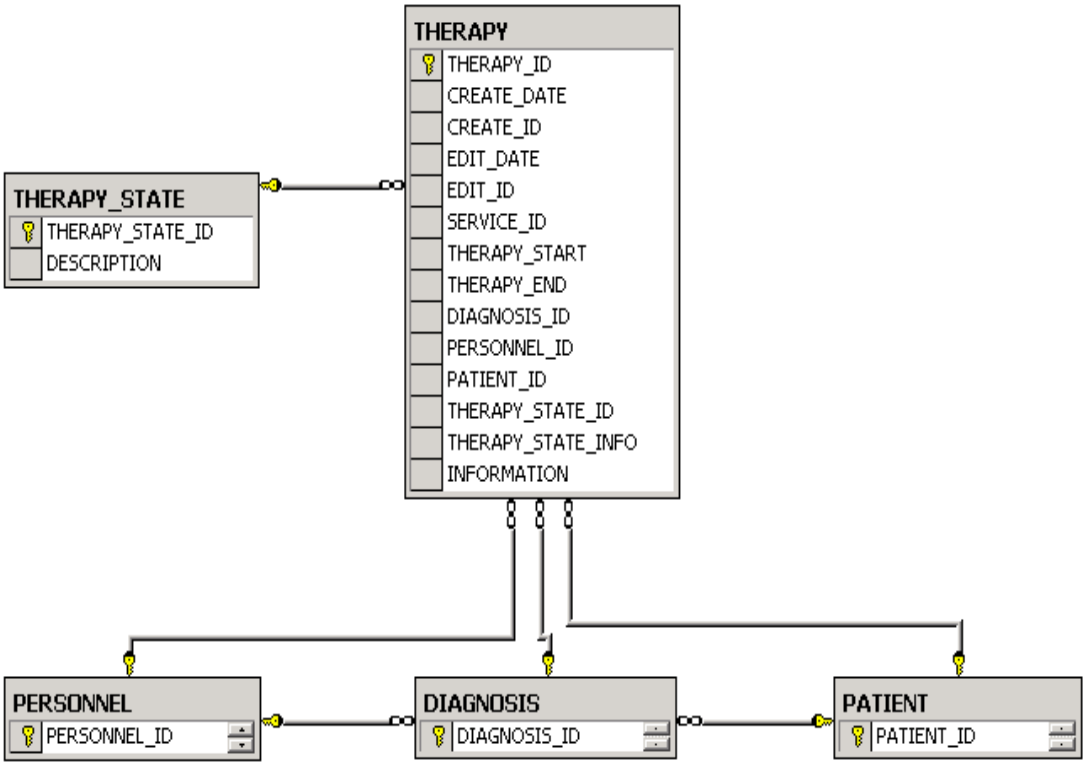
5.3.7. Therapies module

Το Therapies module είναι η αρμόδια λειτουργία για την καταγραφή των πληροφοριών θεραπείας για τον ασθενή. Μπορεί να περιλαμβάνει τυχών φυσιοθεραπεία, χημειοθεραπεία ή οποιαδήποτε θεραπεία.

5.3.7.1. Entity Relationship Diagram

Μια θεραπεία σχετίζεται με την διάγνωση αφού μόνο μετά από μια διάγνωση μπορούμε να καθορίσουμε μια θεραπεία. Έχουμε ακόμη δυο πίνακες, τον dbo. THERAPY για αποθήκευση των εγγραφών κάθε θεραπείας και τον dbo. THERAPY_STATE ο οποίος περιέχει μια λίστα από την πρόοδο που μπορεί νε έχει μια θεραπεία (In Progress, Completed, Not Started κ.α.).

Σχήμα5.9



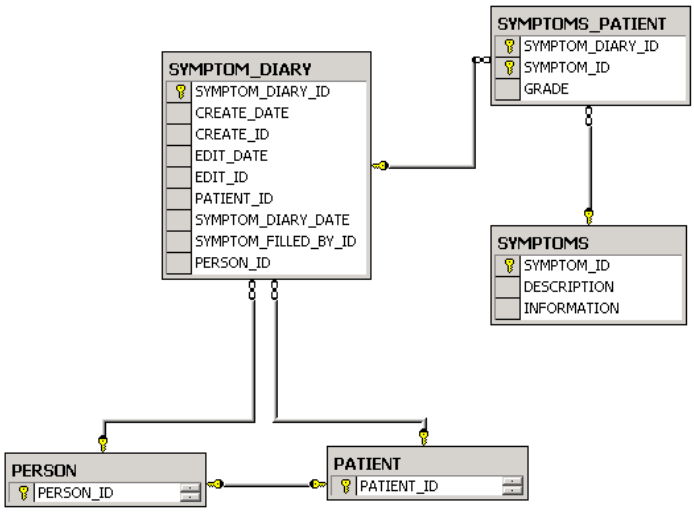
5.3.8. Symptom Diary module

Το Symptom Diary module φροντίζει για την καταγραφή των συμπτωμάτων ανά ημερομηνία ενός ασθενή.

5.3.8.1. Entity Relationship Diagram

Οι πίνακες που το συνθέτουν είναι οι ακόλουθοι. Ο πρώτος είναι ο πίνακας dbo.SYMPTOM_DIARY όπου φυλάσσονται τα symptom diaries. Ακολουθά ο πίνακας αναζήτησης dbo.SYMPTOMS (lookup table) που περιέχει μια λίστα από συμπτώματα που μπορεί να έχει ένας ασθενής. Και τελευταίος είναι ο πίνακας dbo.SYMPTOMS_PATIENT, ένας πίνακας many-to-many relationship μεταξύ των πινάκων dbo.SYMPTOM_DIARY και dbo.SYMPTOMS tables. [4]

Σχήμα5.10



Κεφάλαιο 6

Υλοποίηση Συστήματος

6.1. Αρχιτεκτονική 55

6.2. Ασφάλεια 57

6.3. Χρήση συγκεκριμένων component της ASP.Net 59

6.4. Περιγραφή υλοποίησης για κάθε οθόνη 70

Στο κεφάλαιο αυτό θα αναφερθούμε στην αρχιτεκτονική που χρησιμοποιήσαμε καθώς και στην ασφάλεια που εξασφαλίσαμε με τον τρόπο που υλοποιήθηκε η συγκεκριμένη εφαρμογή. Τέλος θα αναφερθούμε στα σημαντικότερα components της ASP.Net που χρησιμοποιήσαμε κατά την υλοποίηση της διαδικτυακής εφαρμογής μας.

6.1. Αρχιτεκτονική

Η πιο γνωστή αρχιτεκτονική τα τελευταία είκοσι χρόνια (κυρίως σε εφαρμογές βάσεων δεδομένων και web εφαρμογές) είναι αυτή του πελάτη – εξυπηρετητή (client - server). Σε αυτή την αρχιτεκτονική, ο πελάτης στέλνει ένα αίτημα (request) για δεδομένα στον εξυπηρετητή και αυτός επιστρέφει την απάντηση (response), την οποία επεξεργάζεται ο πελάτης και εμφανίζει στο χρήστη τα αποτελέσματα.

Το πρόβλημα σε αυτή την προσέγγιση είναι ότι η εμφάνιση των δεδομένων και η επεξεργασία τους γίνεται από το ίδιο πρόγραμμα, τον πελάτη. Αν υπάρχουν πολλαπλά κανάλια διάχυσης της πληροφορίας ή συχνή αλλαγή στη μορφή παρουσίασης, τότε θα πρέπει να αλλάζει κάθε φορά η client εφαρμογή.

6.2.2. *Αρχιτεκτονική Server - Side scripting*

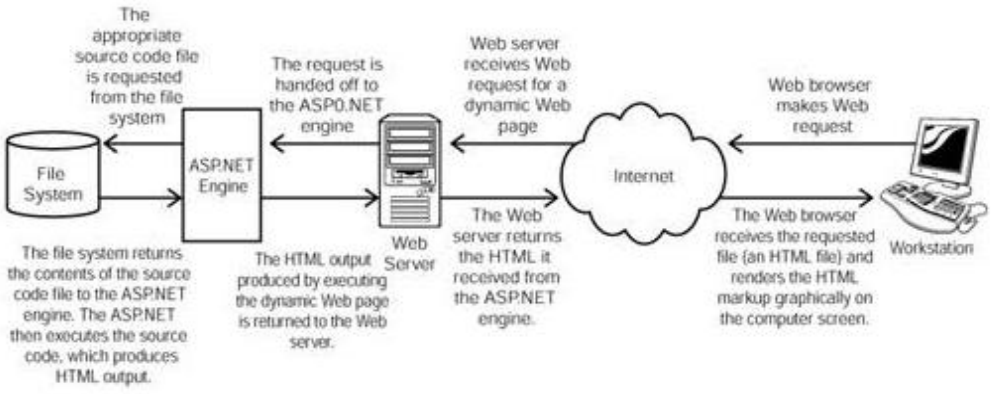
Όπως προαναφέραμε η τεχνολογία ASP (Active Server Pages).Net με την οποία εργαστήκαμε μπορούμε να δημιουργήσουμε ιστοσελίδες που περιλαμβάνουν δυναμικές web pages χρησιμοποιώντας την αρχιτεκτονική Server - Side scripting.

Επομένως στην εφαρμογή μας χρησιμοποιούμε την τεχνολογία Server – Side Scripting με την οποία επιτυγχάνουμε την εκτέλεση ενός κώδικα στην πλευρά του server πριν ακόμα φορτωθεί στον υπολογιστή του χρήστη η παραγόμενη ιστοσελίδα.

Για την εμφάνιση μιας ιστοσελίδας το πρώτο που συμβαίνει είναι ότι αρχικά ο χρήστης ζητάει μέσα από τον φυλλομετρητή του την εμφάνιση της ιστοσελίδας αυτής (μέσω του URL). Ακολούθως ο browser στέλνει ένα request στο web server (είναι ένα software application το οποίο συνέχεια περιμένει να έρθουν web requests, τα οποία είναι για συγκεκριμένα URL) και ζητά το συγκεκριμένο αρχείο. Ο web server αφού πρώτα βρει το αρχείο που του ζητήθηκε στην συνέχεια στέλνει το αρχείο πίσω στο πελάτη που έκανε το request.

Στην περίπτωση που η web page είναι στατική, τότε γίνεται αυτό που είπαμε πιο πάνω. Αν όμως είναι δυναμική και ζητείτε μια ASP.NET web page τότε ο web server περνά την ευθύνη για εκτέλεση της σελίδας στο ASP.NET engine (κομμάτι software που γνωρίζει πώς να εκτελέσει μια ASP.NET web pages), όπως φαίνεται και στο πιο κάτω σχήμα.

Σχήμα6.1.

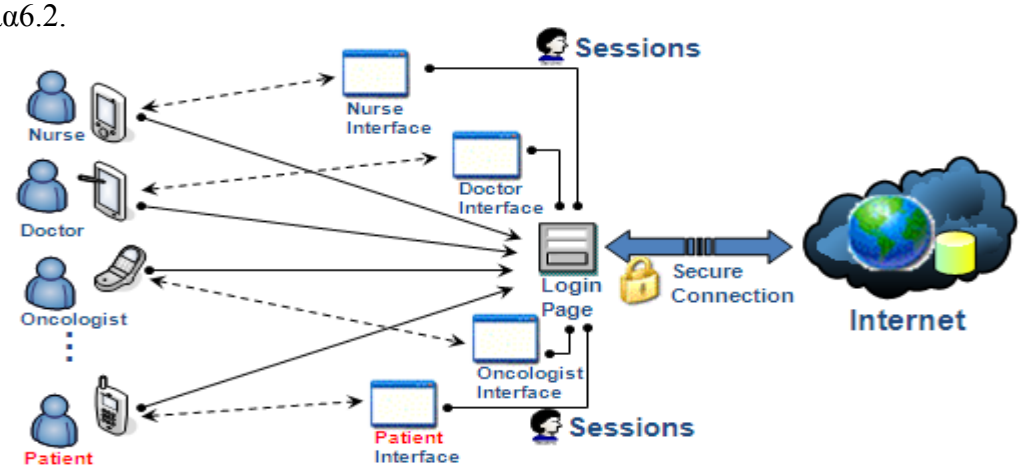


Ο web server μπορεί να καταλάβει αν η σελίδα είναι στατική ή δυναμική από το extension που έχει το αρχείο που ζητήθηκε. Αν είναι .aspx τότε ο web server ξέρει ότι η σελίδα που ζητήθηκε είναι ASP.NET, και τότε περνά το request στο ASP.NET engine. Στην συνέχεια η ASP.NET engine εκτελεί την σελίδα ASP.NET, και παράγει μια σελίδα που περιέχει μόνο HTML κώδικα. Το παραγόμενο αρχείο επιστρέφει στον web server, ο οποίος επιστρέφει το HTML στον φυλλομετρητή (browser) που έκανε το request.

6.1.2. Αρχιτεκτονική Διεπιφάνειας

Στο σχήμα που ακολουθεί μπορούμε να δούμε την βασική αρχιτεκτονική που χρησιμοποιήθηκε όσον αφορά την διεπιφάνεια της διαδικτυακής εφαρμογής μας κατά την φάση της υλοποίησης.

Σχήμα6.2.



Βλέπουμε πως κάθε εγγεγραμμένος χρήστης του συστήματος για να μπορέσει να μπει στο σύστημα καλείται να κάνει login στην ίδια σελίδα. Αν τα στοιχεία που δόθηκαν στην login φόρμα είναι σωστά τότε ο χρήστης μεταφέρεται ανάλογα με τον ρόλο του στο interface που αντιστοιχεί στο ρόλο του.

6.2. Ασφάλεια

Κατά την διάρκεια της ανάπτυξης της εφαρμογής αναπτύχθηκαν κάποια επίπεδα ασφαλείας. Αρχικά εμποδίζεται η είσοδος στο σύστημα σε όσους χρήστες δεν έχουν σχέση με το σύστημα. Ακόμη εμποδίζουμε τους χρήστες του συστήματος από το να

δούνε σελίδες που δεν έχουν δικαίωμα να δούνε, δίνοντας πρόσβαση μόνο σε αυτές που έχουν δικαιοδοσία. Η χρήση ακόμη των stored procedures μας διασφαλίζει ένα είδος ασφάλειας.
Ειδικότερα, για να αποτρέψουμε τους 'περαστικούς' χρήστες από το να μπούνε στο σύστημα, ο χρήστης πρέπει στην αρχική σελίδα να δηλώσει το όνομα χρήστη με το οποίο είναι γραμμένος στο σύστημα και τον προσωπικό του κωδικό πρόσβασης.

Παράλληλα με την παραπάνω διαδικασία, και μόνο εάν ο χρήστης έχει επιλέξει την επιλογή να θυμάται ο υπολογιστής το όνομα χρήστη και το συνθηματικό του, ο διακομιστής τοποθετεί ένα cookie στον υπολογιστή του χρήστη το οποίο περιέχει το όνομα χρήστη και το password του. Η λειτουργία αυτού του cookie είναι απλή. Όταν ο χρήστης ξαναεπισκεφτεί την εφαρμογή, το σύστημα θα ψάξει να βρει αυτό το αρχείο και θα διαβάσει από εκεί τα στοιχεία του. Έτσι δεν θα χρειαστεί να τα συμπληρώσει ο χρήστης ξανά και ξανά σε κάθε του επίσκεψη.

Ο έλεγχος τόσο του password όσο και του username γίνεται ουσιαστικά μέσο του connection με την βάση μας. Βασικά η εφαρμογή μας, αντί να έχει ένα πίνακα στην βάση που να φυλάγει τα στοιχεία εισόδου του χρήστη, κάθε χρήστης με το password και το username του ενώνεται με την βάση μας χρησιμοποιώντας τα. Αν δεν γίνει το connection σημαίνει ότι κάποιο από τα δυο στοιχεία είναι λάθος. Η σύνδεση με την βάση μας χρησιμοποιώντας το password και το username αποτελεί το 'εισιτήριο' του χρήστη στο σύστημα μας και χωρίς αυτό, δεν μπορεί να δει καμία σελίδα της εφαρμογής.

Ακόμη, το σύστημα ελέγχει τον ρόλο του κάθε χρήστη και ανάλογα με αυτό θα μπορέσει να δει την ζητούμενη σελίδα.

Τέλος με την χρήση των Stored procedures είναι σαν να έχουμε ένα πρόσθετο στρώμα ασφάλειας. Μπορούμε να αποτρέψουμε για ασφάλεια την άμεση πρόσβαση στους πίνακες μας, και απλά να προσθέσουμε ένα στρώμα από Stored procedures μέσω του οποίου οι χρήστες θα μπορούν να έχουν πρόσβαση σε δεδομένα της βάσης.

Η παροχή ενός ασφαλούς περιβάλλοντος βάσεων δεδομένων σε μια εφαρμογή Ιστού είναι ιδιαίτερα σημαντική, δεδομένου ότι πολλοί είναι αυτοί που πιθανόν να προσπαθήσουν να έχουν πρόσβαση στα δεδομένα χωρίς να δικαιούνται (π.χ. hackers). Ο Ιστός εκθέτει τα δεδομένα μας στον εξωτερικό κόσμο, έτσι εφαρμόζοντας ένα στρώμα από Stored procedures που ελέγχει τα updates, insertions, και deletions μπορούμε να έχουμε μια σημαντική βοήθεια στην προσπάθεια μας να κάνουμε πιο ασφαλή την ιστοσελίδα μας επιτρέποντας να δώσουμε πρόσβαση σε λειτουργίες της βάσης χωρίς να δίνουμε πρόσβαση στους πίνακες.

Ως γνωστόν, όλα τα λειτουργικά συστήματα έχουν κάποιες τρύπες ασφαλείας (security holes) τα οποία μπορούν να εκμεταλλευτούν ορισμένα άτομα και να μπορέσουν να δούνε τα περιεχόμενα του συστήματος. Η εφαρμογή που αναπτύξαμε εκτελείται σε ένα τέτοιο λειτουργικό σύστημα. Συνεπώς, εμείς θα προσπαθήσουμε να κάνουμε όσο πιο ασφαλές γίνεται το περιβάλλον μας.

6.3. Χρήση συγκεκριμένων component της ASP.Net

Στην υλοποίηση μας έγινε χρήση κάποιων components που μας προσφέρει η ASP.Net για την εύκολη και "έξυπνη" δημιουργία μιας εφαρμογής. Στην συνέχεια ακολουθεί μια μικρή περιγραφή για το καθένα από αυτά καθώς και κομμάτια από την υλοποίηση που δείχνουν τον τρόπο που τα χρησιμοποιήσαμε.

6.3.1. Χρήση StyleSheet CSS

Τα CSS (Cascading Style Sheets), στην κατασκευή ιστοσελίδων, είναι ηλεκτρονικά έγγραφα με ένα σύνολο κανόνων για την μορφοποίηση μιας ιστοσελίδας.

Παλαιότερα, ήταν συνήθεια να διαμορφώνεται κάθε στοιχείο της ιστοσελίδας ξεχωριστά, σε κάθε μία ξεχωριστή σελίδα. Η μεγάλη ευκολία που προσφέρει η χρήση ενός CSS, είναι ότι οι ενδεχόμενες αλλαγές, γίνονται μόνο σε ένα έγγραφο και αυτόματα εφαρμόζονται σε όλες τις σελίδας που το χρησιμοποιούν. Έτσι μία ιστοσελίδα που χρησιμοποιεί CSS, μπορεί πολύ ευκολότερα να αλλάξει όψη.

Για αυτό το λόγο έγινε χρήση CSS για να καθορίσουμε τα χρώματα, το μέγεθος διαφόρων στοιχείων της ιστοσελίδας, το background, το μέγεθος των γραμμάτων κ.α. Πιο κάτω βλέπουμε ένα κομμάτι από το CSS αρχείο που χρησιμοποιήσαμε.

```
/* CSS Document */
body {
background: url(images/bg7.jpg) repeat-x #EAEAEE;
font-family:Verdana, Geneva, sans-serif;
font-size:12px;
}
#main_body {
	background:url(images/blue_bg1.jpg);
	height:410px;
}
#container {
width: 900px;
margin: 0 auto;
overflow: hidden;
}
#banner {
	width: 900px;
	height: 170px;
}
#main_menu {
	float: right;
	text-align:center;
	font-size:12px;
	font-weight:bold;
}
```

Για να ενσωματώσουμε τώρα στις διάφορες σελίδες μας το CSS το μόνο που χρειάζεται να κάνουμε είναι να βάλουμε στο head την πιο κάτω εντολή:

```
<head runat="server">
   <title>Nurse Account</title>
```

```
   <link href="designHomePage.css" rel="stylesheet" type="text/css" />
</head>
```

6.3.2. Το αρχείο web.config

Το αρχείο web.config αποτελεί το αρχείο ρυθμίσεων κάθε ASP.NET εφαρμογής. Η βασική του μορφή δημιουργείται αυτόματα κατά τη δημιουργία του project της εφαρμογής από το Visual Studio. Στην συνέχεια μπορούμε να προσθέσουμε σε αυτό το αρχείο επιπλέον στοιχεία.

Όσον αφορά την εφαρμογή μας, κάναμε μια παρέμβαση στο αρχείο web.config. Συγκεκριμένα θέσαμε το connectionString στο κατάλληλο σημείο του αρχείου. Η ρύθμιση αυτή καθορίζει τη σύνδεση του ιστότοπου με τη βάση δεδομένων μας. Προφανώς λανθασμένη ή ελλιπής πληροφορία έχει ως αποτέλεσμα την μη σύνδεση με τη βάση δεδομένων μας.

Το connectionStrings μπορείς να το προσθέσεις είτε μέσω του SqlDataSource control's wizard είτε προσθέτοντας στο section με τα connectionStrings του web.config τον σωστό κώδικα.

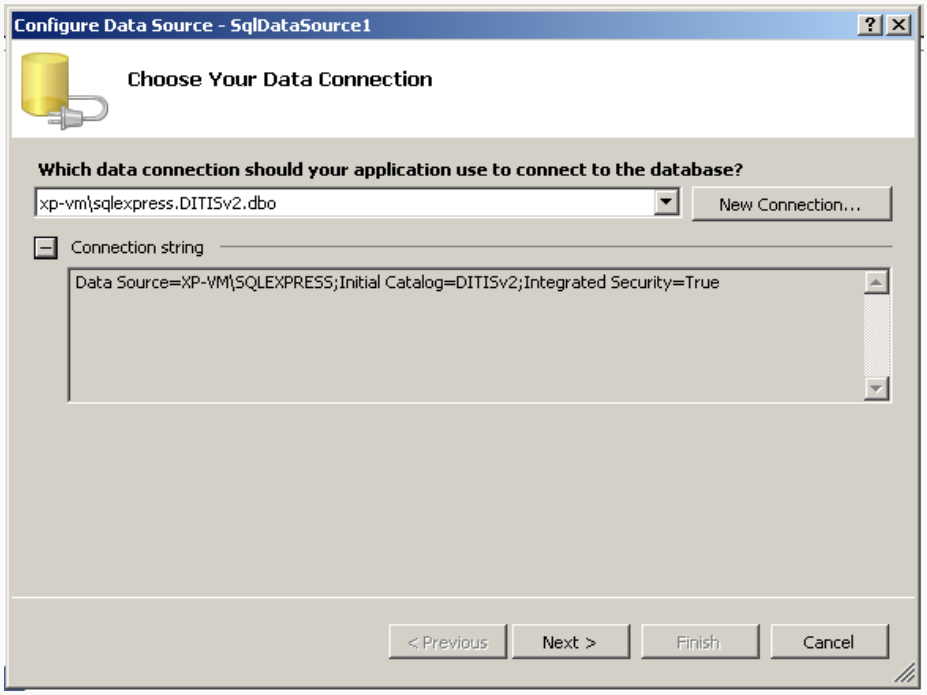
<u>Κώδικας του connection string στο website's configuration file:</u>

```
<appSettings>
<add key="DITISv2ConnectionString" value="Data Source=XP-VM\sqlexpress; Initial Catalog=DITISv2; User ID={0}; Password={1}"/>
</appSettings>
```

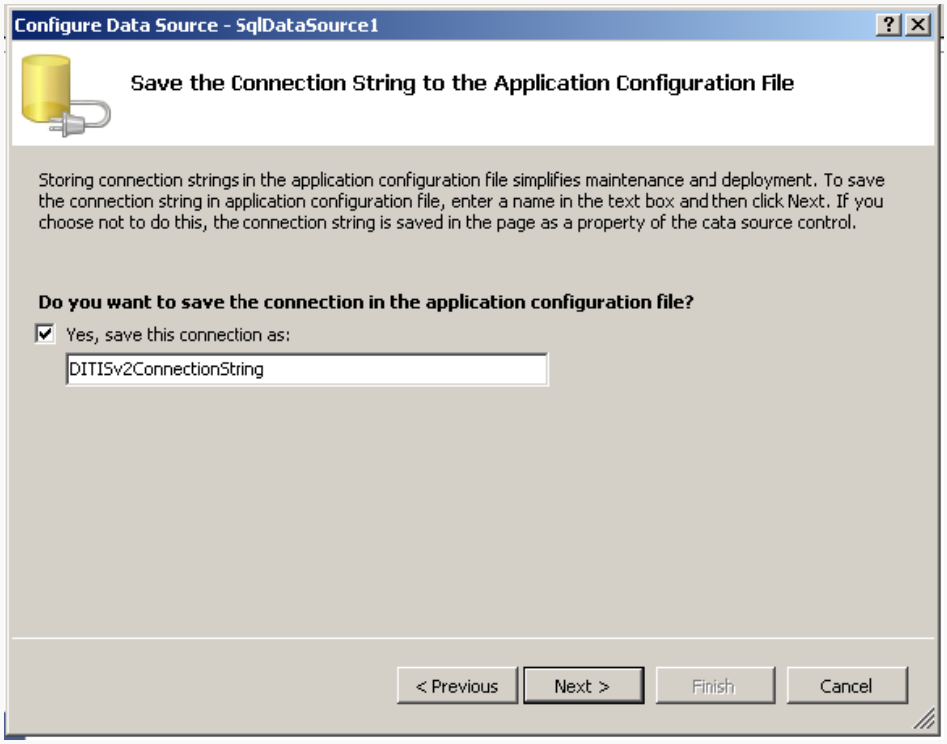
Όπου User ID={0}; Password={1} δύνονται μετά σαν μεταβλητές.

Αποθήκευση του connection string στο website's configuration file μέσω του SqlDataSource control's wizard:

Σχήμα6.3



Σχήμα6.4



Αποθηκεύοντας της πληροφορίες του connection string στο web.config αρχείο έχουμε σαν όφελος ότι προσθέτει ένα επίπεδο indirection στην εφαρμογή μας, η οποία καθιστά την ανταπόκριση στις αλλαγές ευκολότερη. Για παράδειγμα, αν δεν αποθηκεύαμε τις πληροφορίες της σύνδεσης στο web.config αρχείο θα έπρεπε σε κάθε σελίδα ASP.NET που θα δημιουργούσαμε και θα αλληλεπιδρούσε με την βάση να βάζαμε τα στοιχεία αυτά. Στην περίπτωση όμως που το connection string αλλάξει θα χρειαστεί να πάμε σε όλες της σελίδες που το χρησιμοποιήσαμε και να το αλλάξουμε. Με την φύλαξη όμως του connection string στο web.config αρχείο το μόνο που έχουμε να κάνουμε είναι να αλλάζουμε τις πληροφορίες στο συγκεκριμένο αρχείο.

6.3.3. Χρήση Master page

Στις περισσότερες επαγγελματικές εφαρμογές, οι σελίδες που εμφανίζονται στο χρήστη πρέπει να έχουν μια παρόμοια μορφή, για παράδειγμα να υπάρχει ένας πίνακας περιεχομένων σε κάποιο κοινό μέρος ώστε να μπορεί να γίνει πλοήγηση στις άλλες σελίδες, να υπάρχει το λογότυπο της εταιρίας και το όνομα της εφαρμογής, να υπάρχουν κοινές γραμματοσειρές και χρώματα κοκ.

Αυτό αρχικά μπορεί να γίνει με το χέρι, όμως από τη στιγμή που ο αριθμός των σελίδων αυξάνει θα χρειάζεται όλο και περισσότερος χρόνος για να γίνουν τα πιο πάνω. Στην περίπτωση δε, που χρειαστεί να γίνουν αλλαγές στην σχεδίαση της ιστοσελίδας η εργασία που θα χρειαστεί να γίνει θα είναι πάρα πολύ μεγάλη. Για αυτό και υπάρχουν διάφορες τεχνικές οι οποίες υποστηρίζουν κάποιο είδος «template» με τη βοήθεια του οποίου όλες οι σελίδες μπορούν να έχουν κοινά χαρακτηριστικά. Το «template» αυτό στην περίπτωση του ASP.NET ονομάζεται «Master Page».

Μια master page αποτελείται από δύο κομμάτια, τις περιοχές με τα στοιχεία που εμφανίζονται σε όλες τις σελίδες με τον ίδιο τρόπο και τα οποία κληρονομούνται ουσιαστικά από την master page και τις περιοχές που μπορούν να προσαρμοστούν από κάθε σελίδα ξεχωριστά. Το πρότυπο της master page σε ASP.NET επισκιάζει τα απλά HTML-based templates επειδή, όπως και σε οποιαδήποτε σελίδα ASP.NET, μπορούν να περιέχουν όχι μόνο τους ελέγχους σήμανσης HTML, αλλά και Web controls και server-side source κώδικα. Στην master page μπορεί να τοποθετήσουμε Web controls,

User controls, HTML content, and scripts τα οποία μπορούμε να τα βάλουμε σε οποιαδήποτε σελίδα ASP.NET. Μια master page ουσιαστικά είναι μια απλή ASP.NET σελίδα η οποία περιέχει μια σειρά από κανονικά στοιχεία τα οποία θα εμφανίζονται σε όλες τις άλλες σελίδες που την χρησιμοποιούν ως template.

Υπάρχουν τρεις σημαντικές διαφορές μεταξύ μιας master page και μιας κανονικής σελίδας ASP.NET. Κατ' αρχάς, αντίθετα από μια κανονική σελίδα ASP.NET, το όνομα μιας master page πρέπει να τελειώνει με τη ιδική επέκταση .master. Αυτή η επέκταση χαρακτηρίζει τη σελίδα ως master page. Δεύτερον, μια master page περιέχει <%@ Master %> οδηγό αντί του κανονικού <%@ Page %>. Και τέλος μια master page μπορεί να περιέχει μηδέν ή περισσότερους ελέγχους ContentPlaceHolder (<asp:contentplaceholder>) ενώ μια κανονική σελίδα ASP.NET όχι. Αυτά τα content place holders θα χρησιμοποιηθούν από τις σελίδες που χρησιμοποιούν τη master page ώστε να τοποθετήσουν το περιεχόμενο τους εκεί. Έτσι, η δομή όλων των σελίδων θα είναι η ίδια ακριβώς και το μόνο που θα αλλάζει θα είναι το περιεχόμενο του content place holder.

Κάθε σελίδα που χρησιμοποιεί τη master page ως template θα πρέπει να το δηλώνει στην επικεφαλίδα της με κάποια γραμμή της μορφής masterPageFile="MasterPage.master". Για να χρησιμοποιηθεί ένας content place holder χρησιμοποιείται το στοιχείο <asp:content> του οποίου τα περιεχόμενα θα τοποθετηθούν στον αντίστοιχο content place holder.

Η τεχνική αυτή χρησιμοποιήθηκε και εδώ έτσι ώστε να υπάρχει μια ομοιόμορφη δομή σε όλες τις σελίδες που αναπτύχθηκαν και φυσικά να είναι ευκολότερη η ανάπτυξη τους. Στην δική μας ιστοσελίδα χρησιμοποιήσαμε την παρακάτω εικονιζόμενη master page.

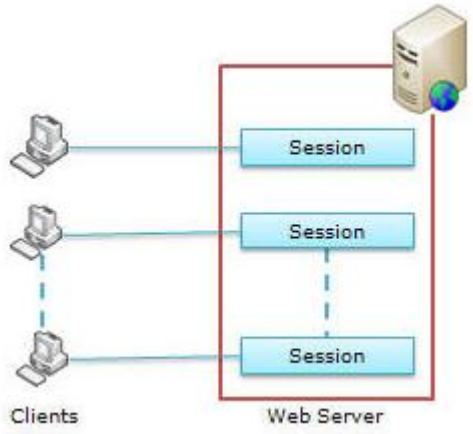
Σχήμα6.5.

6.3.4. Χρήση Sessions

Ο Ιστός είναι κάθε Stateless, αυτό σημαίνει κάθε φορά που ταχυδρομείται η σελίδα στον κεντρικό υπολογιστή ουσιαστικά επαναδημιουργείται. Όπως είναι γνωστό το HTTP είναι ένα stateless πρωτόκολλο, επομένως δεν μπορεί να φυλάσσει τις πληροφορίες των πελατών για τη σελίδα. Εάν ο χρήστης παρεμβάλλει κάποιες πληροφορίες, και κινηθεί προς την επόμενη σελίδα, τότε τα στοιχεία θα χαθούν και ο χρήστης δεν θα μπορεί να ανακτήσει τις πληροφορίες.

Τα Sessions μας παρέχουν την δυνατότητα να αποθηκεύουμε πληροφορίες στην μνήμη του κεντρικού υπολογιστή (server). Μπορεί να υποστηρίξει οποιοδήποτε τύπο αντικειμένου για να τον αποθηκεύσει. Για κάθε client Session τα δεδομένα αποθηκεύονται ξεχωριστά, αυτό σημαίνει ότι κάθε session αποθηκεύεται ξεχωριστά για κάθε πελάτη. Αυτό φαίνεται και στο πιο κάτω σχήμα (Σχήμα 6.5).

Σχήμα 6.6.



Η χρήση Session είναι ένα από τα καλύτερα χαρακτηριστικά γνωρίσματα της asp.net, επειδή είναι ασφαλή, διαφανή από τους χρήστες και μπορούμε να αποθηκεύσουμε σε αυτά οποιοδήποτε είδος αντικειμένου. Μαζί με τα πλεονεκτήματα, κάποιες φορές τα sessions μπορεί να επηρεάσουν την απόδοση για heavy traffic sites επειδή αποθηκεύονται στη μνήμη του κεντρικού υπολογιστή και οι πελάτες διαβάζουν τα δεδομένα από τον κεντρικό υπολογιστή οι ίδιοι.

Ένα session ορίζεται ως η χρονική περίοδος όπου ένας μοναδικός χρήστης αλληλεπιδρά με μια εφαρμογή Ιστού. Όταν ξεκινήσει το session δίνεται στο χρήστη ένα μοναδικό κλειδί. Αυτό το κλειδί φυλάγεται σε ένα HTTP cookie το οποίο στέλνει ο πελάτης στο server σε κάθε request. Ο κεντρικός υπολογιστής μπορεί έπειτα να διαβάσει το κλειδί από το cookie και να βρει το server session state.

6.3.5. Υποστήριξη Πολλών Γλωσσών (Multilanguage Support)

Ένα από τα χαρακτηριστικά του συστήματος είναι η υποστήριξη πολλών γλωσσών. Ο χρήστης μπορεί να επιλέξει την γλώσσα στην οποία θα εμφανίζεται η δικτυακή εφαρμογή και μάλιστα η αλλαγή της γλώσσας γίνεται ακαριαία. Το σύστημα, στην παρούσα του μορφή υποστηρίζει την ελληνική και την αγγλική γλώσσα αλλά η προσθήκη γλώσσας είναι αρκετά εύκολη χάρη στην χρήση της τεχνολογίας των xml.

Κατά την διάρκεια του σχεδιασμού της εφαρμογής, τα μηνύματα που εμφανίζονται στον χρήστη δεν ενσωματώθηκαν στον πηγαίο κώδικα αλλά 'διαβάζονται' από αρχεία xml τα οποία βρίσκονται κάπου στον web server όπου εκτελείται η εφαρμογή.

Συγκεκριμένα το .NET Framework 2.0 καθώς και 1.x έρχονται με ενσωματωμένη υποδομή για τη δημιουργία διεθνών εφαρμογών. Βασικά, το CLR υποστηρίζει έναν μηχανισμό για packaging και deploying των πόρων με οποιοδήποτε τύπο εφαρμογής. Το CLR και η βασική κλάση library του .NET πλαισίου έρχονται με διάφορες κλάσεις για τη διαχείριση και την πρόσβαση των πόρων σε μια εφαρμογή. Αυτές οι κλάσεις βρίσκονται στο System.Resources και το System.Globalization namespaces.

Το Localization υποστηρίζεται στο .Net Framework 2.0 γενικά και στο ASP.Net 2.0 συγκεκριμένα. Συνήθως οι πόροι δημιουργούνται για κάθε πολιτισμό (culture) που η εφαρμογή πρέπει να υποστηρίξει. Πιό συγκεκριμένα, κάθε μορφή σελίδας στον web site πρέπει να έχει τους πόρους για κάθε πολιτισμό , γλώσσα, που υποστηρίζει. Για κάθε σελίδα μας πρέπει να έχουμε δυο αρχεία ένα για κάθε γλώσσα.

Το CLR καθορίζει την συμπεριφορά για την εύρεση της συγκεκριμένης γλώσσας. Κάθε αρχείο καθορίζει σε ποια γλώσσα αντιστοιχεί από το δεύτερο μέρος του ονόματός του (π.χ. Login.aspx.el.resx είναι ένα αρχείο που καθορίζει ελληνικά). Αν δεν καθορίζεται η γλώσσα μέσα στο όνομα τότε είναι το προκαθορισμένο αρχείο.

Στην εφαρμογή μας για την επιλογή της γλώσσας έχουν δημιουργηθεί δυο κουμπιά ένα με την ελληνική σημαία και ένα με την αγγλική σημαία τα οποία αν τα επιλέξουμε θα έχουμε την τροποποιήσει της μεταβλητής "Language" του session που έχει ο χρήστης με τον server και της θέτει σαν τιμή το όνομα της γλώσσας που επέλεξε ο χρήστης.

6.3.6. Χρήση Site Maps

Πριν από την ASP.NET 2.0, ένας πολύ κοινός και πολύ χρονοβόρος στόχος για τους υπεύθυνους ανάπτυξης μιας ιστοσελίδας ήταν να καθορίσουν την δομή της και να δημιουργήσουν site's navigation. Ευτυχώς, αυτές είναι ανησυχίες του παρελθόντος.

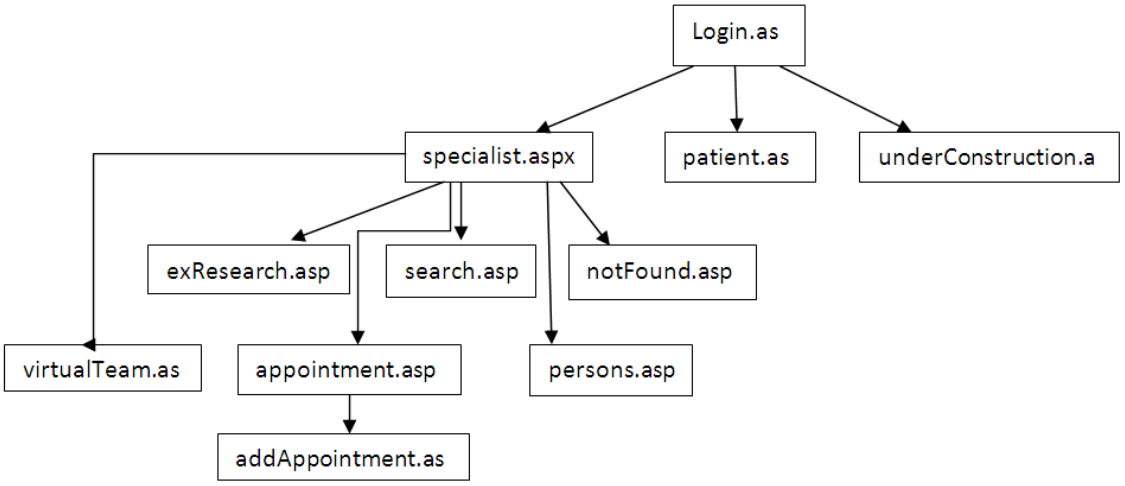
Η ASP.NET παρέχει τα site maps μέσο των οποίων μπορεί κανείς να καθορίσει την δομή της ιστοσελίδας. Ένα site map είναι ουσιαστικά ένα αρχείο XML που καθορίζει τα λογικά τμήματα της ιστοσελίδας και δένει προαιρετικά κάθε τμήμα σε ένα ιδιαίτερο URL. Όταν καθοριστεί ένα site map, διάφοροι νέοι έλεγχοι Ιστού ASP.NET μπορούν να χρησιμοποιηθούν για να ορίσουμε την δομή σε navigation user interface.

Η ASP.NET 2.0 παρέχει τρία navigation Web controls. Το SiteMapPath, το TreeView και το Menu. Στην δική μας εφαρμογή χρησιμοποιήσαμε το SiteMapPath που είναι μια γραμμή κειμένου που παρουσιάζει στο χρήστη την θέση που βρίσκεται κάθε χρονική στιγμή στην ιστοσελίδα. Μέσο αυτού μπορεί ο χρήστης πολύ εύκολα να μεταφερθεί σε προηγούμενη σελίδα με βάση την λογική ιεράρχιση που υπάρχει.

Στην περίπτωση που θελήσουμε να προσθέσουμε νέο τμήμα στην εφαρμογή μας τότε απλά το δημιουργούμε και το προσθέτουμε στο site map και αυτόματα το navigation Web control που χρησιμοποιείται θα συμπεριλαμβάνει και αυτό το τμήμα.

6.3.6.1. Δομή Ιστοσελίδας

Σχήμα6.7



6.3.7. Χρήση Stored Procedures

Ένα stored procedure είναι μια ομάδα SQL statements οι οποίες αποτελούν μια λογική ενότητα για να μπορεί να εκτελεστεί μια συγκεκριμένη εργασία. Χρησιμοποιούνται ακόμη, για να ενώσουμε και να εκτελέσουμε πολλές πράξεις μαζί ή για εκτέλεση ερωτημάτων σε ένα διακομιστή βάσης δεδομένων. Τα stored procedures έχουν πολλά πλεονεκτήματα για αυτό και τα χρησιμοποιούμε στην εφαρμογή μας. Κάποια από αυτά είναι τα ακόλουθα.

6.3.7.1. Συναλλαγές

Αν και ένα stored procedure περιέχει εντολές SQL, μόλις συνταχθεί αυτό θα αλληλεπιδράσει με τον SQL Server πολύ διαφορετικά από τις μεμονωμένες SQL δηλώσεις. Μια από τις βασικές αλλαγές είναι ότι οι εντολές SQL σε ένα stored procedure είτε θα εκτελεστούν όλες ή καμία. Αυτό είναι γνωστό ως atomicity (ατομικότητα).

Ακόμη, αντίθετα από τις τυποποιημένες δηλώσεις SQL, τα stored procedures συντάσσονται και βελτιστοποιούνται από το database server. Αυτή η βελτιστοποίηση περιλαμβάνει τη χρησιμοποίηση των πληροφοριών για τη δομή μιας συγκεκριμένης βάσης δεδομένων την ώρα της εκτέλεσης από ένα stored procedure. Στην περίπτωση που ένα stored procedure καλείται πολλές φορές θα έχουμε εξοικονόμηση χρόνου αφού με τον πιο πάνω τρόπο έχουμε storing execution της πληροφορίας.

6.3.7.2. Ταχύτητα

Ένα ακόμη πλεονέκτημα είναι ότι βελτιώνεται η ταχύτητα αφού τα stored procedures τρέχουν εξ ολοκλήρου στον database server και έτσι δεν χρειάζεται να περάσουμε από SQL κώδικα μέσα ένα δίκτυο. Για μια απλή SELECT δήλωση, μπορεί να μην κάνει μεγάλη διαφορά, αλλά σε περιπτώσεις όπου εκτελούμε μια σειρά βρόχων και υπολογισμών, μπορεί να έχουμε σημαντική βελτίωση στην ταχύτητα.

6.3.7.3. Έλεγχος διεργασιών

Ένα stored procedure μπορεί να εκμεταλλευτεί τις δηλώσεις ροής ελέγχου όπως IF...ELSE, FOR και WHILE loops, οι οποίες δεν είναι διαθέσιμες σε basic SELECT

statement. Αυτό είναι πολύ βοηθητικό γιατί μας επιτρέπει να χειριστούμε μερικές αρκετά σύνθετες λογικές διαδικασίες μέσα από τον κώδικα SQL. Χωρίς stored procedure σε μια τέτοια περίπτωση, θα είχαμε μεγάλο network traffic λόγο του μεγάλου αριθμού αρχείων που θα πρέπει να υποβληθεί σε επεξεργασία. Με την χρήση τους όμως επιστρέφονται μόνο τα στοιχεία που απαιτούνται από τον πελάτη/χρήστη.

Η χρήση των δηλώσεων ροής ελέγχου είναι βασική σε οποιαδήποτε γλώσσα προγραμματισμού, και με την εφαρμογή αυτής της λειτουργίας, ο SQL Server γεφυρώνει το χάσμα μεταξύ του κώδικά μας και της βάσης δεδομένων.

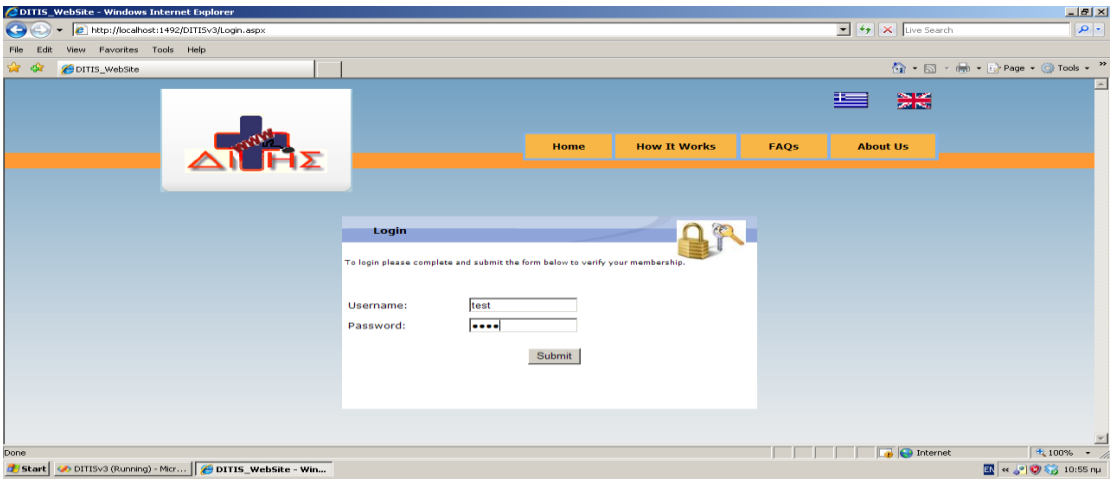
6.3.7.4. Συντήρηση / Επαναχρησιμοποίηση

Τα Stored procedures επιτρέπουν την ευκολότερη συντήρηση. Είναι συγκεντρωμένα, έτσι μπορούμε να επαναχρησιμοποιήσουμε τα υπάρχοντα Stored procedures σε όλο το σύστημα. Έχουμε βασικά εύκολη μεταφορά σε άλλη γλώσσα προγραμματισμού εφόσον όλη η λογική της βάσης υλοποιείται εσωτερικά και όχι στην γλώσσα προγραμματισμού. Είναι ευκολότερο να έχουμε πρόσβαση σε αυτά, να τα συντηρούμε, να τα επαναχρησιμοποιούμε και τέλος να κάνουμε ανά πάσα στιγμή αλλαγές σε αυτά.

Τα stored procedures που θα χρησιμοποιήσουμε να αναφερθούν πιο κάτω κατά την περιγραφή της κάθε σελίδας και των λειτουργιών της.

6.4. Περιγραφή υλοποίησης για κάθε σελίδα

6.4.1. Login σελίδα



Η πρώτη σελίδα της εφαρμογής μας είναι η Login.aspx. Η γλώσσα που θα εμφανιστεί αυτή η σελίδα εξαρτάτε από την γλώσσα που έχει σαν επιλογή στον φυλλομετρητή του ο χρήστης. Τόσο σε αυτή την οθόνη όσο και στις ακόλουθες στο πάνω δεξί μέρος υπάρχουν οι σημαίες για την επιλογής γλώσσας. Όταν ο χρήστης πατήσει σε κάποια από τις δυο τότε η μεταβλητή Session("Language") αλλάζει αναλόγως ("el" για ελληνικά και "en-US" για αγγλικά) και ο χρήστης πλέον βλέπει στις οθόνες του και την ανάλογη γλώσσα.

Στην πρώτη σελίδα συγκεκριμένα ο χρήστης πρέπει να δώσει τα σωστά στοιχεία για να προχωρήσει στην εφαρμογή. Ακόμη από το μενού μπορούμε να ενωθούμε με την σελίδα http://www.ditis.ucy.ac.cy/ .

Για να ελέγξουμε αν τα στοιχεία που έδωσε ο χρήστης είναι σωστά, χρησιμοποιούμε τρείς μεταβλητές. Αρχικά ελέγχουμε ότι δόθηκαν τόσο το username όσο και το password. Στην περίπτωση που κάποιο από τα δυο ή και τα δυο δεν δόθηκε βγαίνουν τα κατάλληλα προειδοποιητικά μηνύματα στον χρήστη. Ο επόμενος έλεγχος που γίνεται αφορά την ορθότητα των στοιχείων αυτών. Ουσιαστικά το username και το password είναι τα sql στοιχεία που χρειάζεται ο χρήστης για να κάνει login στον sql server. Επομένως αν δεν μπορέσουμε να ενωθούμε με τον sql server σημαίνει ότι κάποιο από τα στοιχεία που δόθηκαν είναι λάθος και βγαίνει το κατάλληλο μήνυμα στον χρήστη.

Αφού βεβαιωθήκαμε για την ορθότητα των στοιχείων με το παραπάνω τρόπο ορίζουμε μια μεταβλητή session, Session("islogin") = True για να μπορούμε ανά πάσα στιγμή στην συνέχεια να ελέγχουμε μέσο αυτής ότι δεν χάθηκε για κάποιο λόγο η σύνδεση του χρήστη. Αν για κάποιο λόγο διακοπή η σύνδεση τότε ο χρήστης αυτομάτως οδηγείτε στη σελίδα Login.aspx για να κάνει ξανά σύνδεση. Ακόμη διαβάζουμε το connectionString από το αρχείο Web.Config αντικαθιστούμε τις δυο μεταβλητές και αποθηκεύουμε στο Session("connStr") το connectionString μας για να μπορούμε να ενωνόμαστε με την βάση μας στην συνέχεια χρησιμοποιώντας το. Ο κώδικας που χρησιμοποιήσαμε για αυτό είναι ο ακόλουθος:

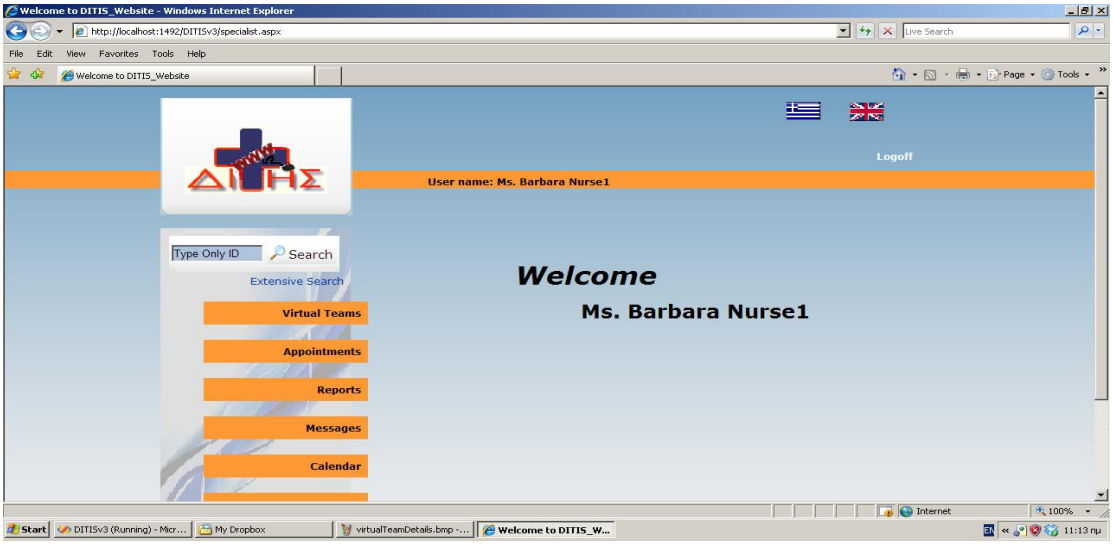
```
Dim connStr As String  = System.Configuration.ConfigurationManager.AppSettings("DITISv2ConnectionString")
connStr = String.Format(connStr, Session("user"), Session("pass"))
Session("connStr") = connStr
```

Τέλος, για να μπορέσουμε να βρούμε το ID του person , τον ρόλο, το όνομα και το uid του χρήστη καλλούμε τα stored procedures sp_fc_USER_PROFILE και sp_mia_FindUserRole. Στην συνέχεια ανάλογα με τον ρόλο οδηγούμαστε και στην επόμενη οθόνη.

6.4.2. Σελίδες specialists

6.4.2.1. Σελίδα Welcome

Σχήμα6.9.



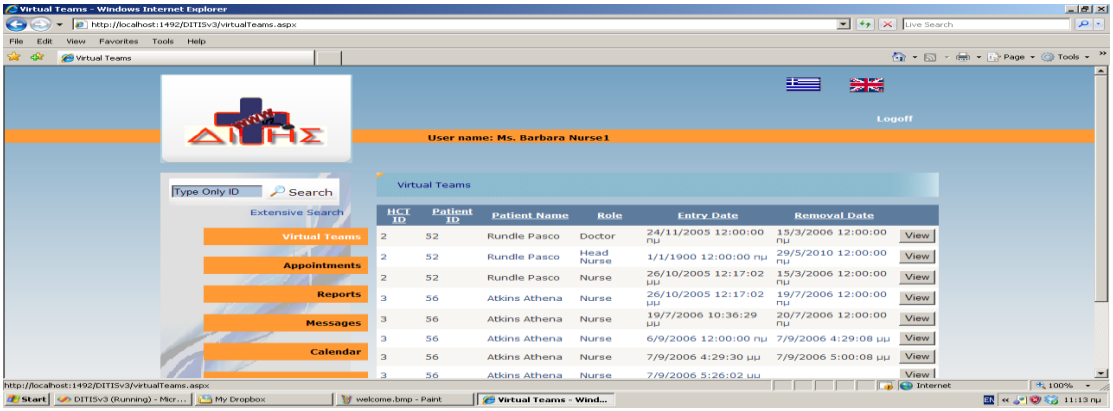
Ακολουθεί η οθόνη καλωσορίσματος όπου καλωσορίζεται ο χρήστης και μπορεί να δει το μενού με τις λειτουργίες που μπορεί να εκτελέσει ανάλογα με τον ρόλο του. Η συγκεκριμένη οθόνη παρουσιάζεται σε όλους τους specialist.

Σε όλες τις σελίδες που ακολουθούν και γίνεται χρήση Grid View, ο χρήστης μπορεί να κάνει τόσο κανονική όσο και αντίστροφη ταξινόμηση πάνω σε οποιαδήποτε από τις στήλες. Αυτό γίνεται για διευκόλυνση του χρήστη να βρει κάτι συγκεκριμένο.

Ακόμη ο χρήστης μπορεί ανά πάσα στιγμή να κάνει Logoff πατώντας στο link Logoff που βρίσκεται και αυτό στο πάνω δεξί μέρος κάθε οθόνης.

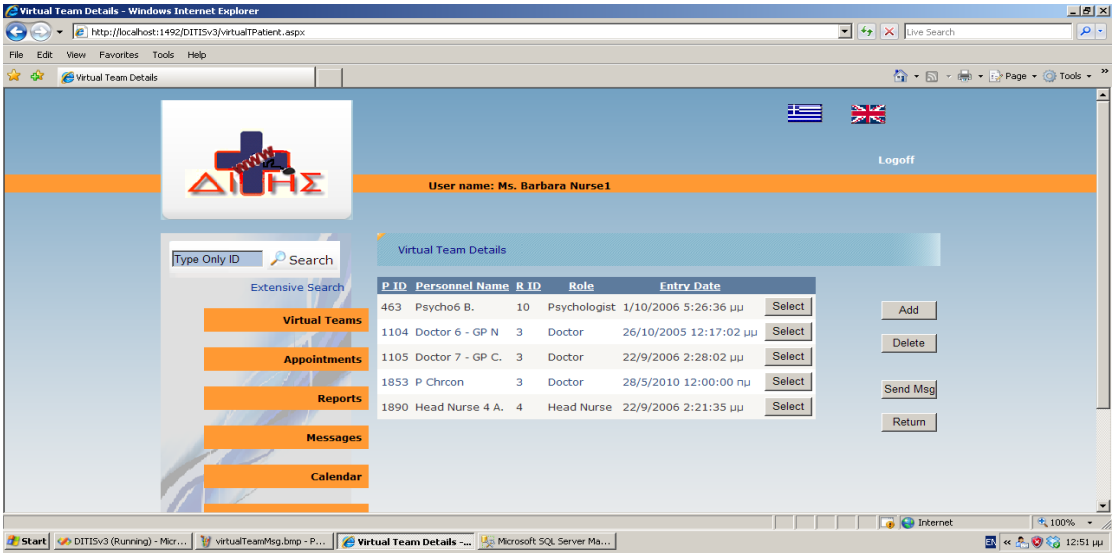
6.4.2.2. *Σελίδες Virtual Teams*

Σχήμα6.10.



Αν ο χρήστης επιλέξει την λειτουργία Virtual Teams οδηγείτε σε μια σελίδα στην οποία μπορεί να δει σε ποιές εικονικές ομάδες ανήκει ο ίδιος. Για να γίνει αυτό χρησιμοποιούμε ένα Grid View και το stored procedure sp_fc_select_HCT_BY_PERSONNEL. Στην συνέχεια μπορεί να διαλέξει μια συγκεκριμένη και να δει περισσότερα στοιχεία πατώντας το κουμπί View που βρίσκετε διπλά από κάθε regord. Αυτόματα μόλις το πατήσουμε φυλάγουμε σε μεταβλητές session το ID του ασθενή που θέλουμε να δούμε περισσότερες λεπτομέρειες για την εικονική του ομάδα και το ID της εικονικής ομάδας τα οποία θα χρειαστούμε στις επόμενες σελίδες.

Σχήμα6.11.



Αν επιλέξει ο χρήστης να δει περισσότερες λεπτομέρειες τότε ο οδηγείτε στην σελίδα VirtualTPatient.aspx στην οποία μπορεί να δει μέσο ενός Grid View που καλεί το

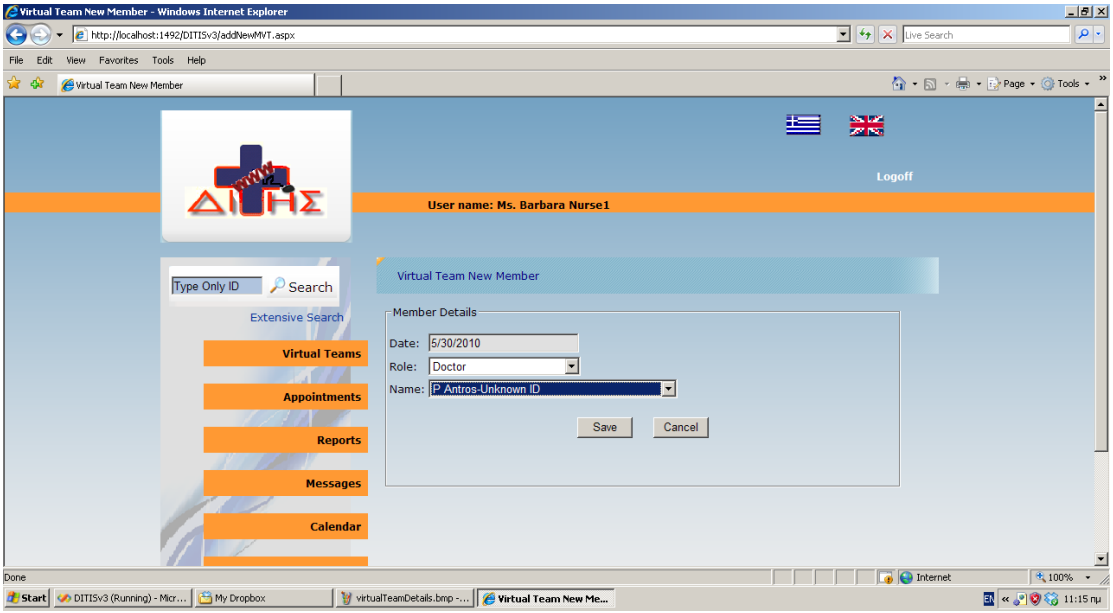
stored procedure sp_fc_select_HCT_ENTRIE την εικονική ομάδα του ασθενή που επέλεξε.

Ακόμη μέσο αυτής της σελίδας μπορεί είτε να εισάγει ένα νέο μέλος στην εικονική ομάδα, να διαγράψει κάποιο που υπάρχει, να στείλει μήνυμα σε κάποιο από τα άλλα μέλη ή να επιστρέψει στην προηγούμενη οθόνη.

Αν ο χρήστης πατήσει στην επιλογή Delete χωρίς να επιλέξει πρώτα κάποιον personnel βγαίνει το κατάλληλο προειδοποιητικό μήνυμα στην οθόνη ότι πρέπει να επιλέξει πρώτα κάποιο regord. Στην συνέχεια βγαίνει ένα μήνυμα που ρωτά τον χρήστη αν πράγματι θέλει να διαγράψει τον συγκεκριμένο personnel από την ομάδα. Αν πάρουμε θετικέ επιβεβαίωση από τον χρήστη τότε διαγράφεται ο συγκεκριμένος personnel χρησιμοποιώντας το stored procuder sp_fc_delete_HCT_ENTRIES ενώ παράλληλα ενημερώνεται αυτόματα από το σύστημα αποστέλλοντας του ένα μήνυμα ότι έχει διαγραφή από την πιο πάνω ομάδα. Αυτό το επιτυγχάνουμε χρησιμοποιώντας το stored procuder sp_fc_insert_MESSEGES και το sql query SELECT [PERSON_NAME]FROM [DITISv2].[dbo].[view_PERSON_NAMES]WHERE [PERSON_ID] = Session("VTPatientID"). Στο τέλος ενημερώνεται ο χρήστης για την ολοκλήρωση της λειτουργίας.

Πατώντας το κουμπί Add οδηγούμαστε στην σελίδα addNewMVT.aspx ενώ όταν πατήσουμε το κουμπί Send Msg οδηγείτε στην σελίδα sendMsgVT.aspx.

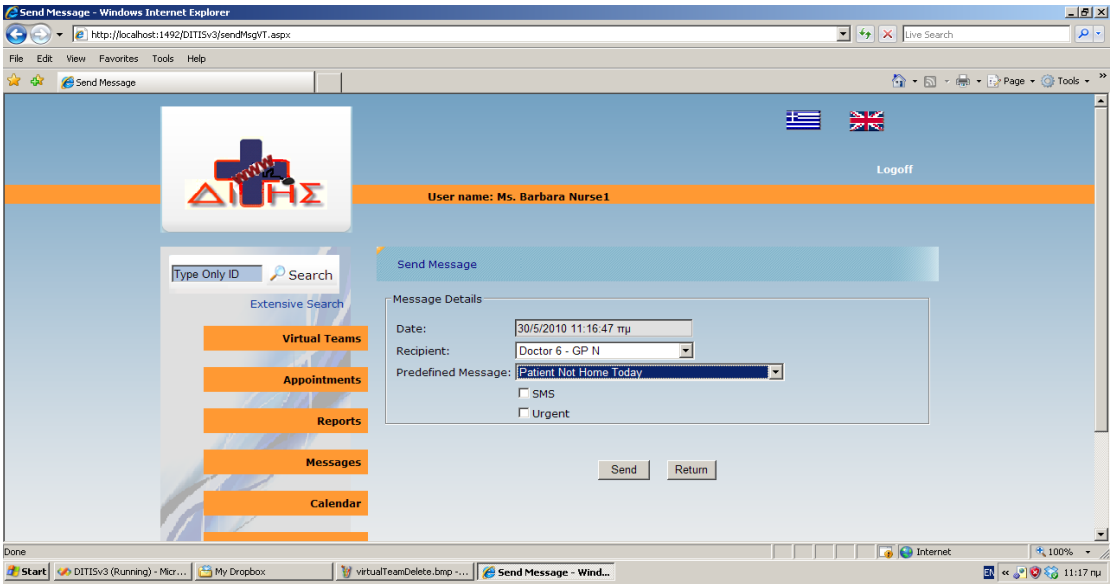
Σχήμα6.12.



Στην σελίδα addNewMVT.aspx ουσιαστικά ο χρήστης το μόνο που έχει να κάνει είναι να διαλέξει το ρόλο του μέλους που θέλει να εισάγει και αυτόματα(γίνεται χρήση του sp_fc_select_PROFESSION) θα εμφανιστούν μόνο τα άτομα που είναι εγγεγραμμένα στο σύστημα με τον συγκεκριμένο ρόλο(γίνεται χρήση του sp_fc_select_PERSONS_BY_ROLE) ευκολύνοντας έτσι τον χρήστη. Η ημερομηνία δίνεται αυτόματα και δεν μπορεί ο χρήστης να την αλλάξει.

Αν ο χρήστης επιλέξει το κουμπί Save ή Cancel τότε ζητείτε από αυτόν η επιβεβαίωση της επιλογής του μέσο μηνύματος. Στην περίπτωση του Cancel τότε θα επιστρέψει στην σελίδα virtualTPatient.aspx, ενώ στην περίπτωση της επιλογής Save ελέγχεται από το σύστημα αν ο νέος personnel είναι είδη μέλος της συγκεκριμένης ομάδας με χρήση του stored procedure sp_mia_FindIfMemberVTEx το οποίο επιστρέφει τον αριθμό τον record. Στην περίπτωση που είδη υπάρχει απλά βγαίνει ενημερωτικό μήνυμα στο χρήστη και δεν ξαναεισάγεται, αν δεν υπάρχει τότε προστίθεται στην ομάδα και στέλνεται αυτόματα μήνυμα στον νέο personnel για ενημέρωση όσον αφορά την εισαγωγή του στην εικονική ομάδα του συγκεκριμένου ασθενή. Για να γίνουν αυτά χρησιμοποιούνται τόσο τα stored procedures sp_fc_insert_HCT_ENTRIES και sp_fc_insert_MESSAGES όσο και το sql query SELECT [PERSON_NAME]FROM [DITISv2].[dbo].[view_PERSON_NAMES]WHERE [PERSON_ID] = Session("VTPatientID"). Στο τέλος ενημερώνεται ο χρήστης για την ολοκλήρωση της λειτουργίας.

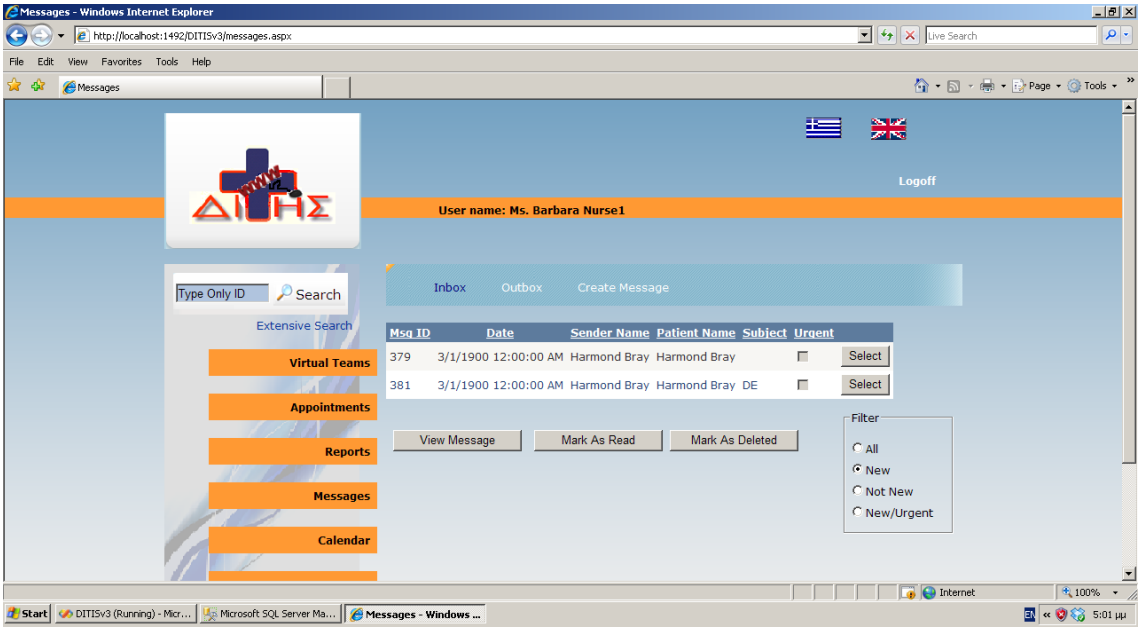
Σχήμα6.13.



Η τελευταία σελίδα που αφορά τις εικονικές ομάδες είναι η sendMsgVT.aspx η οποία μοιάζει αρκετά εμφανισιακά με την addNewMVT.aspx. Έχουμε και εδώ μια ημερομηνία που μπαίνει αυτόματα και δεν μπορεί να αλλάξει ο χρήστης που αντιπροσωπεύει βασικά την ημερομηνία αποστολής του μηνύματος, και δυο drop down list. Στο ένα διαλέγουμε τον παραλήπτη του μηνύματος ανάμεσα στα μέλη της ομάδας και το δεύτερο το αυτόματο μήνυμα που θέλουμε να στείλουμε και με την χρήση του sp_fc_insert_MESSAGES αποθηκεύουμε την αλλαγή. Στο τέλος ενημερώνεται ο χρήστης για την ολοκλήρωση της λειτουργίας.

6.4.2.3. Σελίδες Messages

Σχήμα6.14.



Στην σελίδα messages.aspx ουσιαστικά ο χρήστης μπορεί να δει τα εισερχόμενα μηνύματα του φιλτράροντας τα. Μπορεί να τα δει όλα, τα αδιάβαστα, τα διαβασμένα και τα επείγον ανάλογα με την επιλογή που θα κάνει. Βασικά χρησιμοποιούμε ένα grid view και το sp_mia_FindInboxMsg που επιστρέφει ανάλογα με την επιλογή του χρήστη τα μηνύματα στην οθόνη.

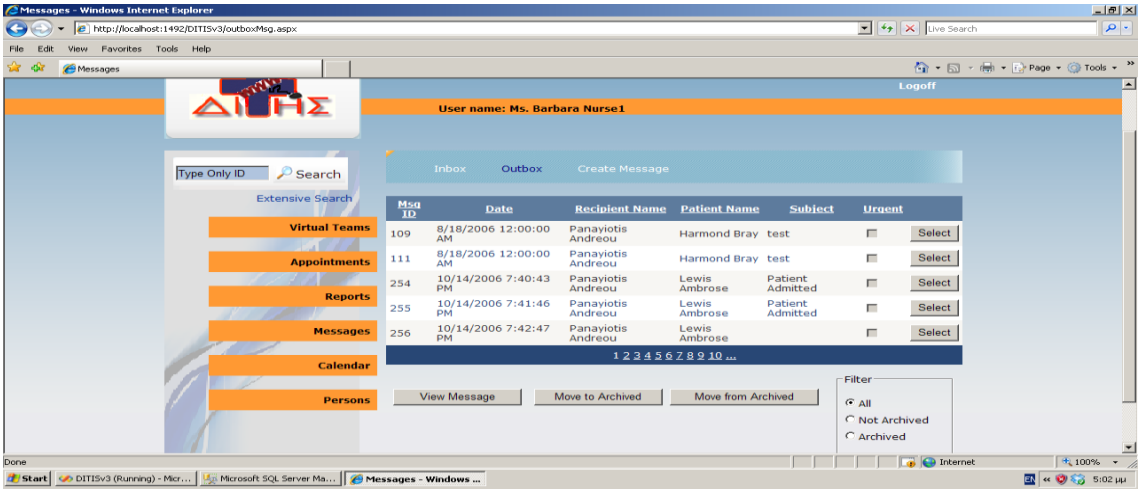
Ακόμη μπορεί μέσο αυτής της σελίδας να πάει στην σελίδα outboxMsg.aspx όπου μπορεί να δει τα εξερχόμενα μηνύματα του και στην σελίδα createMsg.aspx.

Αν κτυπήσουμε κάποιο από τα κουμπιά View Message, Mark As Read και Mark As Deleted χωρίς να έχουμε επιλέξει κάποιο από τα μηνύματα, τότε βγαίνει στην οθόνη ένα μήνυμα που παροτρύνει τον χρήστη να επιλέξει πρώτα κάποιο μήνυμα.

Αν έγινε επιλογή μηνύματος και πατήσαμε το κουμπί Mark As Read τότε το επιλεγόμενο μήνυμα τίθεται σαν διαβασμένο και για να το δει ο χρήστης θα πρέπει να πάει στα Not New μηνύματα. Αυτόματα εμφανίζεται μήνυμα για ενημέρωση του χρήστη όσον αφορά την ολοκλήρωση της λειτουργίας.

Αν τώρα έγινε επιλογή μηνύματος και πατήσαμε το κουμπί Mark As Deleted τότε εμφανίζεται μήνυμα για επιβεβαίωση από τον χρήστη της διαγραφής και αν είναι θετική τότε το επιλεγόμενο μήνυμα τίθεται σαν διαγραμμένο. Αυτόματα εμφανίζεται μήνυμα για ενημέρωση του χρήστη όσον αφορά την ολοκλήρωση της λειτουργίας.
Και για τις δυο αυτές λειτουργίες χρησιμοποιούμε το stored procedure sp_fc_update_MESSAGES και το sql query select F_DELETED, F_READ, F_ARCHIVE FROM [dbo].[MESSAGES] WHERE [MESSAGE_ID]= Session("InBoxMsg") για να δούμε την τιμή τον άλλον flags. Με την επιλογή View Message οδηγούμαστε στην σελίδα viewMsg.aspx και τίθεται η μεταβλητή Session("InboxOrOutbox") ίση με 1.

Σχήμα6.15.



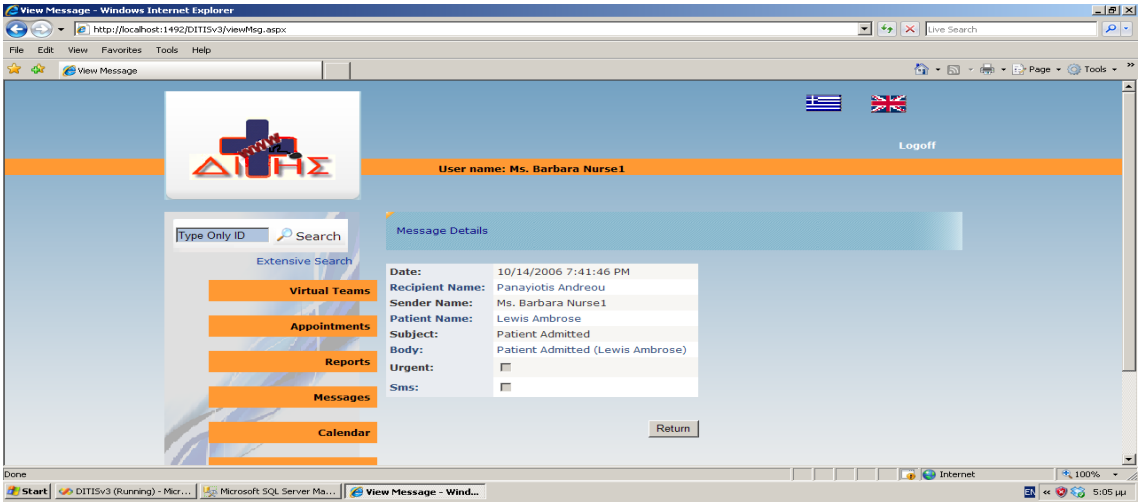
Εδώ ο χρήστης μας μπορεί να δει τα εξερχόμενα του μηνύματα πάλι είτε όλα, είτε μόνο τα archived, είτε μόνο τα not archived ανάλογα με το τι θα επιλέξει στο filter. Σε αυτή την περίπτωση όμως γίνεται χρήση του sp_mia_FindOutboxMsg

Και πάλι στην περίπτωση που πατήσει σε κάποιο κουμπί χωρίς να γίνει επιλογή του μηνύματος εμφανίζεται μήνυμα στο χρήστη. Με την επιλογή View Message

οδηγούμαστε στην σελίδα viewMsg.aspx αφού πρώτα η μεταβλητή Session("InboxOrOutbox") πάρει την τιμή 2.

Αν έγινε επιλογή μηνύματος και πατήσαμε το κουμπί Move to Archived τότε το επιλεγόμενο μήνυμα τίθεται σαν Archive και για να το δει ο χρήστης θα πρέπει να πάει στα Archived μηνύματα. Αυτόματα εμφανίζεται μήνυμα για ενημέρωση του χρήστη όσον αφορά την ολοκλήρωση της λειτουργίας. Με τον ίδιο τρόπο γίνεται και η λειτουργία Move from Archived. Για να κάνουμε αυτές τις αλλαγές χρησιμοποιούμε το stored procedure sp_fc_update_MESSAGES και αλλάζουμε το ανάλογο flag αφού πρώτα καλέσουμε το sql query select F_DELETED, F_READ FROM [dbo].[MESSAGES] WHERE [MESSAGE_ID]= Session("OutBoxMsg") για να δούμε την τιμή τον άλλον flags.

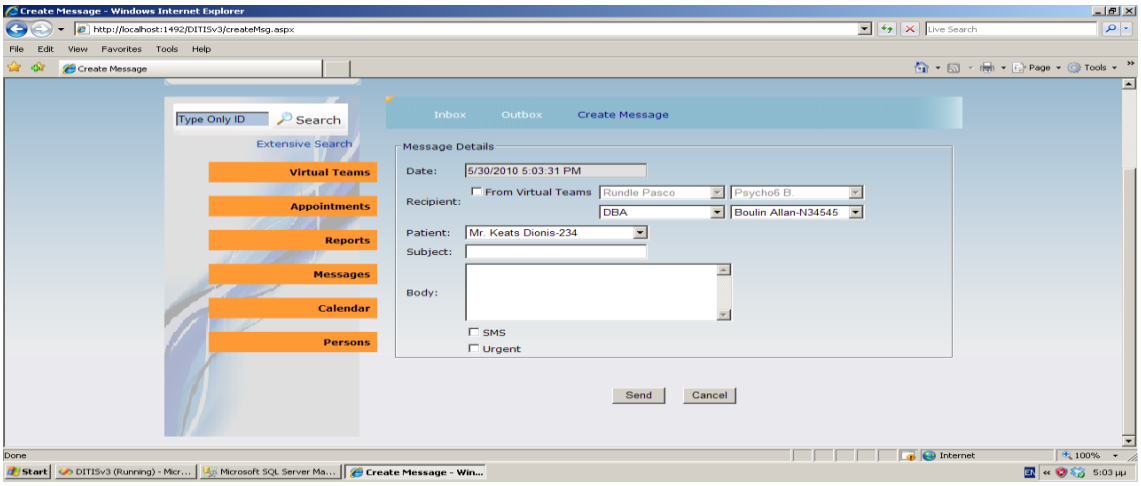
Σχήμα6.16.



Στην σελίδα ViewMsg.aspx βασικά βλέπουμε με την βοήθεια του sql query SELECT [MESSAGE_DATE], a.[PERSON_NAME] Sender_name, b.[PERSON_NAME] Patient_name, [SUBJECT], [F_URGENT], [F_SMS], [BODY], c.[PERSON_NAME] Recipient_name FROM [MESSAGES] inner join [view_PERSON_NAMES] a on [MESSAGES].[FROM_ID] = [a].[PERSON_ID] INNER JOIN [view_PERSON_NAMES] b ON [MESSAGES].[PATIENT_ID] = [b].[PERSON_ID] INNER JOIN [view_PERSON_NAMES] c ON [MESSAGES].[TO_ID] = [c].[PERSON_ID] WHERE [MESSAGE_ID] = @msgID "

βλέπουμε τις λεπτομέρειες του μηνύματος. Όταν πατήσει ο χρήστης το κουμπί Return ανάλογα με την τιμή που έχει η μεταβλητή Session(“InboxOrOutbox”) οδηγούμαστε και στη σελίδα από την οποία ήρθαμε.

Σχήμα6.17.



Η τελευταία σελίδα που αφορά τα messeges είναι η createMsg.aspx στην οποία ουσιαστικά ο χρήστης μας μπορεί να δημιουργήσει και να αποστείλει μήνυμα σε οποιοδήποτε άλλο εγγεγραμμένο χρήστη στου συστήματος διαλέγοντας πρώτα τον ρόλο και μετά το άτομο και διαλέγοντας τον ασθενή για τον οποίο μιλάμε. Αυτό γίνεται με την χρήση των stored procedures sp_fc_select_PROFESSION και sp_fc_select_PERSONS_BY_ROLE.

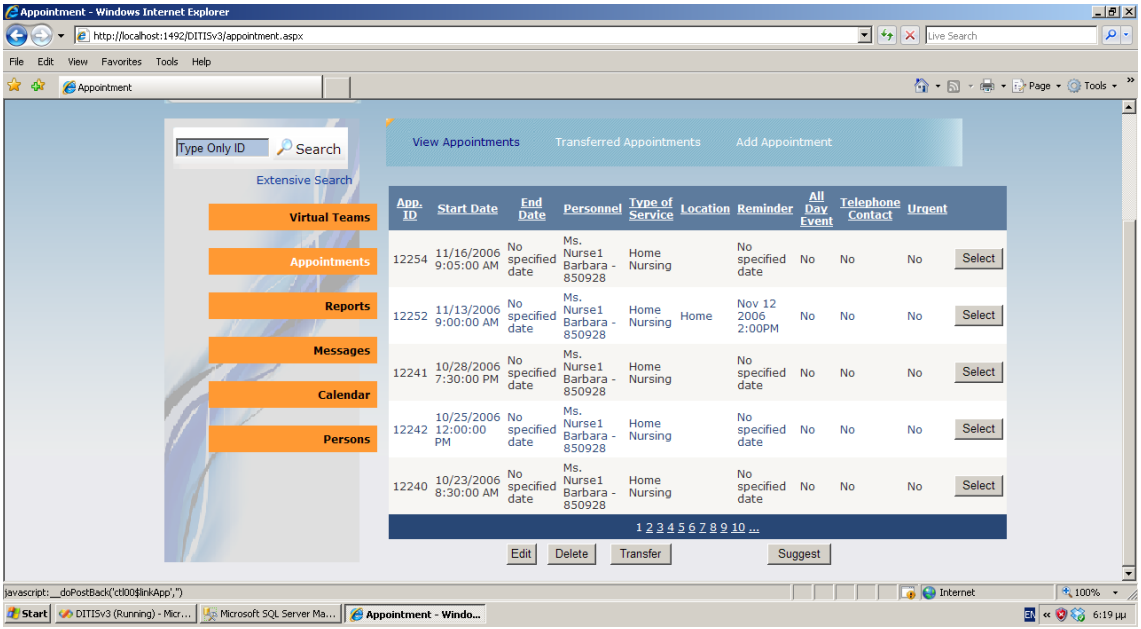
Μπορεί όμως να διαλέξει την επιλογή From Virtual Teams και αμέσως ενεργοποιούνται οι άλλες δυο επιλογές που φαίνονται στην οθόνη. Μέσο τον οποίων μπορεί ο χρήστης να επιλέξει τον ασθενή και μετά ένα άλλο μέλος από την εικονική ομάδα του συγκεκριμένου ασθενή για να του αποστείλει μήνυμα. γίνεται με την χρήση των stored procedures sp_fc_select_HCT_BY_PERSONNEL και sp_fc_select_HCT_ENTRIES. Ακόμη ο χρήστης μπορεί να ορίσει το μήνυμα αυτό ως επείγον ή να το στείλει σε sms στον παραλήπτη του.

Πατώντας το κουμπί Clear και Send βγαίνει μήνυμα προς τον χρήστη για επιβεβαίωση της επιλογής του. Στην συνέχεια ο χρήστη με την εκτέλεση της λειτουργίας Clear οδηγείτε στην σελίδα messages.aspx. Όσον αφορά την επιλογή Send με την βοήθεια του sp_fc_insert_MESSAGES αποστέλλουμε το μήνυμα τον παραλήπτη ενώ στην

οθόνη του χρήστη βγαίνει μήνυμα για ενημέρωση της επιτυχούς ολοκλήρωσης της λειτουργίας αυτής.

6.4.2.4. Σελίδες Appointments

Σχήμα6.18.



Όταν ο χρήστης επιλέξει από το μενού την επιλογή appointments τότε μεταφέρεται στη οθόνη appointment.aspx όπου εμφανίζονται τα ραντεβού του συγκεκριμένου χρήστη. Από αυτή την σελίδα μπορεί αλλάξει, να διαγράψει και να μεταφέρει ένα ραντεβού που υπάρχει ενώ παράλληλα μπορεί να κάνει νέο ραντεβού, να προτείνει ένα νέο ραντεβού σε κάποιο και να δει τα transferred appointments.

Στην περίπτωση που οδηγηθούμε σε αυτή την σελίδα από το search δηλαδή από τα στοιχεία ενός ασθενή τότε εμφανίζεται και ένα κουμπί Return το οποίο πατώντας το επιστρέφουμε στην σελίδα patientInfoGeneral.aspx από οπού και οδηγηθήκαμε σε υτή την οθόνη.

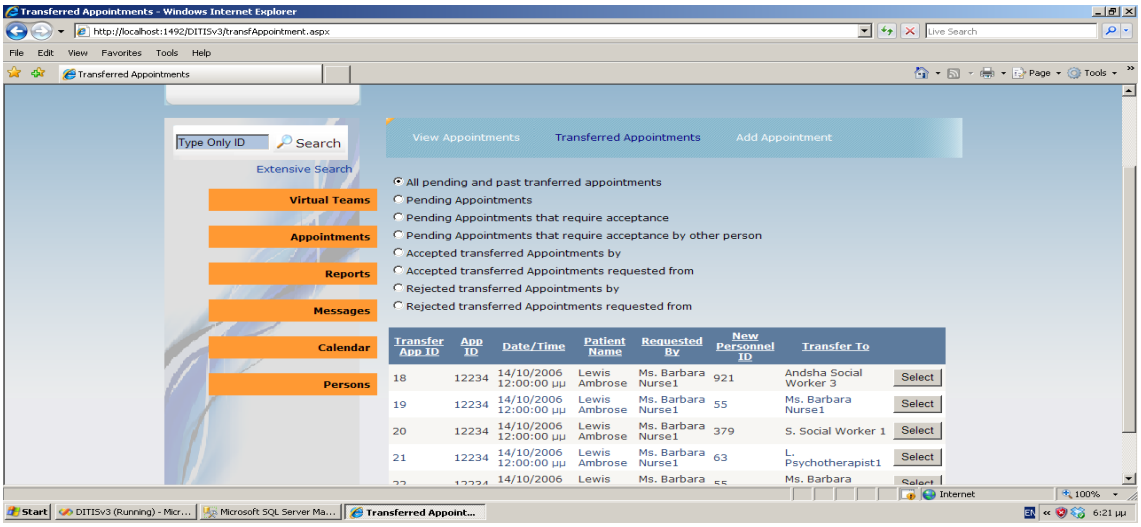
Συγκεκριμένα για να διαγράψει ένα ραντεβού πρέπει πρώτα να το επιλέξει αλλιώς βγαίνει προειδοποιητικό μήνυμα. Ακολούθως βγαίνει μήνυμα για επιβεβαίωση της λειτουργίας αυτής και στην περίπτωση που είναι θετική η απάντηση τότε διαγράφεται ο

ραντεβού μέσο της χρήσης του sp_fc_delete_APPOINTMENT. Στο τέλος ενημερώνεται ο χρήστης για την ολοκλήρωση της λειτουργίας.

Για νε μεταφέρουμε ένα ραντεβού σε κάποιο άλλο τότε επιλέγουμε το ραντεβού και πατάμε το κουμπί Transfer. Με την επιλογή αυτή οδηγούμαστε στην σελίδα transferApp.aspx. Το ίδιο συμβαίνει και με το κουμπί Edit. Δηλαδή αν δεν επιλέξουμε ραντεβού βγαίνει μήνυμα και στην συνέχεια αν επιλεγεί οδηγούμαστε στην σελίδα editAppointment.aspx.

Με την επιλογή Suggest απλά μεταφερόμαστε σε μια άλλη σελίδα την suggestApp.aspx. Το ίδιο και με την επιλογή transferred appointments και add appointments όπου μεταφερόμαστε στις σελίδες transfAppointment.aspx και addAppointment.aspx αντίστοιχα.

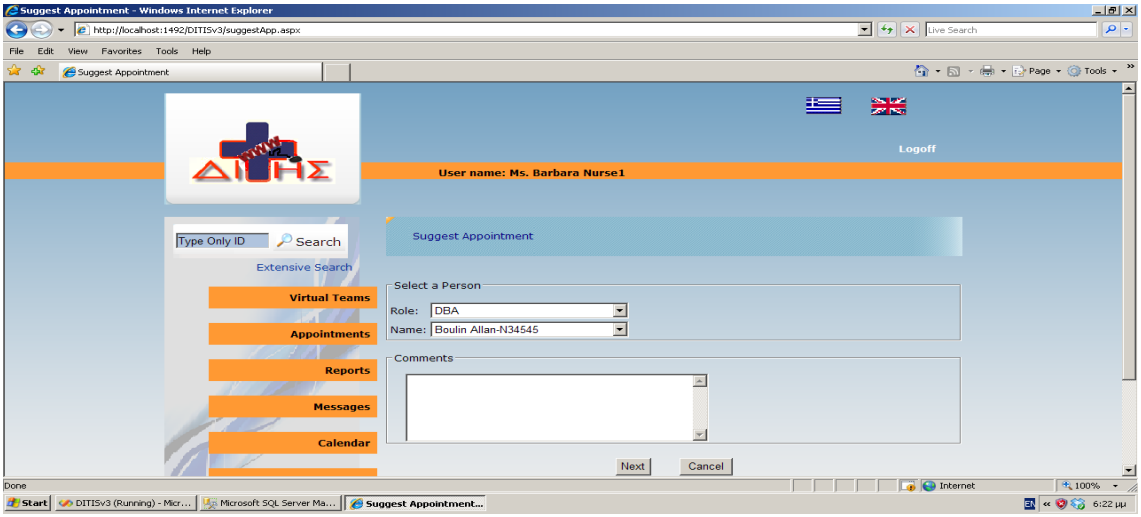
Σχήμα6.19.



Στην πιο πάνω σελίδα ουσιαστικά μπορούμε να επιλέξουμε μια από τις οχτώ επιλογές και ανάλογα θα μας εμφανιστούν και τα κατάλληλα αποτελέσματα. Αυτό γίνεται μέσω του sp_fc_select_TRANSFER_APPOINTMENTS. Στην περίπτωση που έχουμε ραντεβού που περιμένουν την αποδοχή ή την απόρριψη τότε επιλέγοντας το Pending Appointments that require acceptance θα εμφανιστούν δυο κουμπιά, Accept και Reject. Επιλέγοντας επομένως ένα ραντεβού θα μπορούμε να το απορρίψουμε ή να τα αποδεχτούμε, αυτό γίνεται μέσο του sp_fc_update_TRANSFER_APPOINTMENTS.

Και σε αυτή την περίπτωση γίνονται οι κατάλληλη ελέγχει και ενημερώσεις προς τον χρήστη.

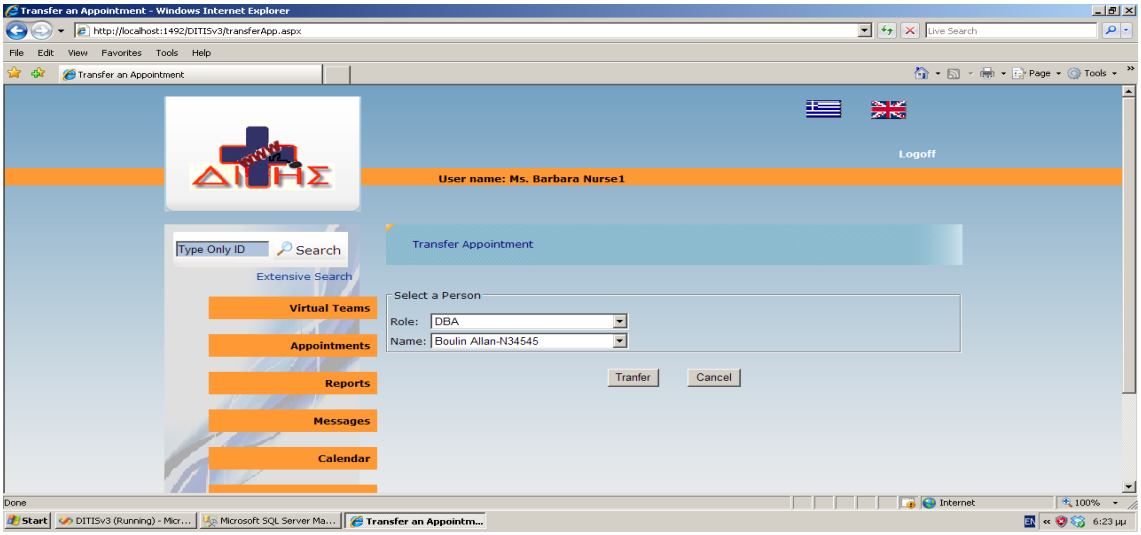
Σχήμα6.20.



Στην σελίδα suggestApp.aspx επιλέγουμε το άτομο που θέλουμε να του προτείνουμε κάποιο ραντεβού. Αυτό γίνεται με την χρήση των sp_fc_select_PROFESSION και sp_fc_select_PERSONS_BY_ROLE.

Έχουμε τα κουμπιά Next και Cancel. Με το Next οδηγούμαστε ουσιαστικά την σελίδα addAppointment.aspx ενώ με την επιλογή Cancel πάμε πίσω στην σελίδα appointment.aspx. Και στις δυο περίπτωσης αρχικά ζητούμε την επιβεβαίωση του χρήστη μέσω ενός μηνύματος.

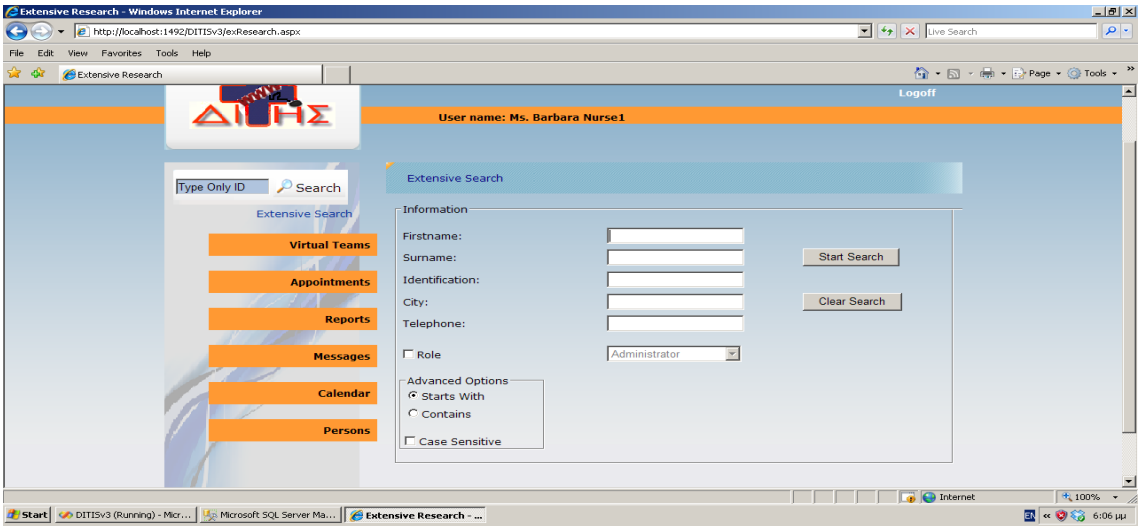
Σχήμα6.21.



Για να μεταφέρουμε ένα ραντεβού το μόνο που έχουμε να κάνουμε είναι να διαλέξουμε τον εγγεγραμμένο χρήστη(γίνεται χρήση των sp_fc_select_PROFESSION και sp_fc_select_PERSONS_BY_ROLE) στον οποίο θέλουμε να μεταφέρουμε το ραντεβού και να πατήσουμε το κουμπί Transfer. Και σε αυτή την σελίδα τόσο στο κουμπί Transfer όσο και στο Cancel ζητούμε από τον χρήστη την επιβεβαίωση του αιτήματος του.

6.4.2.5. Σελίδες Extensive Search

Σχήμα6.22.



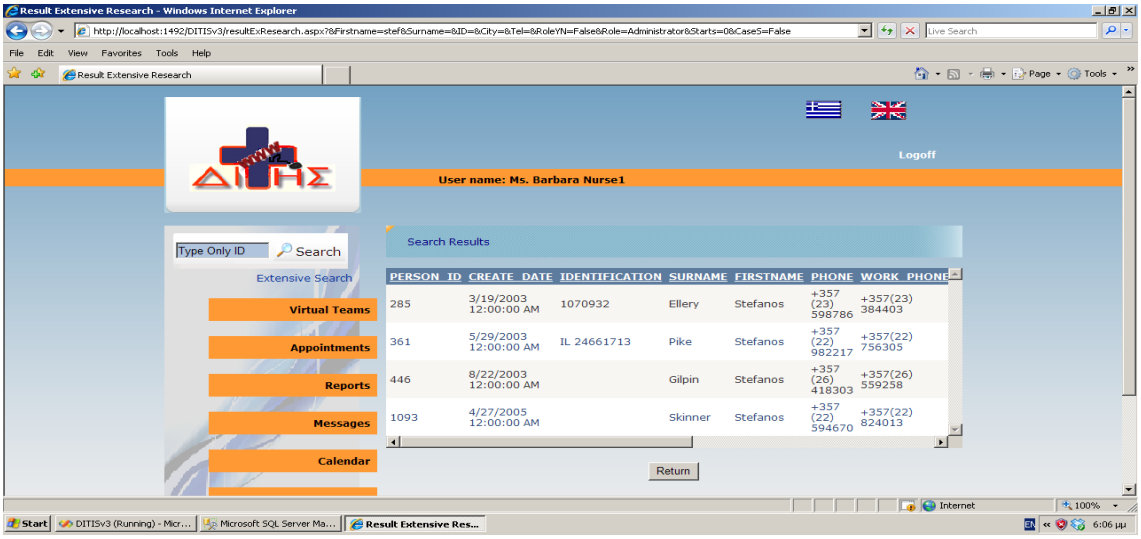
Ένας χρήστης για να μπορέσει να βρει στοιχεία άλλου εγγεγραμμένου χρήστη είτε ασθενή είτε κάποιου άλλου ιδικού μπορεί να χρησιμοποιήσει είτε την επιλογή Extensive Search, είτε την επιλογή Search είτε την επιλογή Persons.

Την επιλογή Extensive Search θα την χρησιμοποιήσει στην περίπτωση που δεν γνωρίζει την ταυτότητα του ατόμου που θέλει να βρει αλλά γνωρίζει κάποιο άλλο στοιχείο όπως όνομα ή μέρος αυτού κ.α.(βλέπουμε στην πάνω οθόνη).

Όταν κάνει τις κατάλληλες επιλογές στην σελίδα αυτή τότε μπορεί να πατήσει το κουμπί Start Search για να δει αν η ερευνά του έχει κάποιο αποτέλεσμα και θα μεταφερθεί στην σελίδα resultExResearch.aspx. Καθώς καλούμε την σελίδα αυτή στέλνουμε σαν παραμέτρους στο url και όλα τα στοιχεία που έδωσε ο χρήστης. Στην περίπτωση που θέλει να καθαρίσει τα στοιχεία που έγραψε τότε πρέπει να πατήσει το

κουμπί Clear Search. Στην περίπτωση που δεν δοθεί κανένα στοιχείο και ο χρήστης πατήσει το κουμπί Start Search βγαίνει το κατάλληλο μήνυμα στην οθόνη.

Σχήμα6.23.



Αν υπάρχει κάποιο αποτέλεσμα από την έρευνα τότε στην σελίδα resultExResearch.aspx θα εμφανιστούν τα αποτελέσματα με την βοήθεια ενός Grid View και του stored procedure sp_fc_SEARCH το οποίο παίρνει σαν παραμέτρους της μεταβλητές που ήρθαν με το request url.

Πατώντας το κουμπί Return μπορεί ο χρήστης να επιστέψει στην σελίδα exResearch.aspx για να κάνει κάποια άλλη αναζήτηση αν το επιθυμεί.

6.4.3. Σελίδες patients

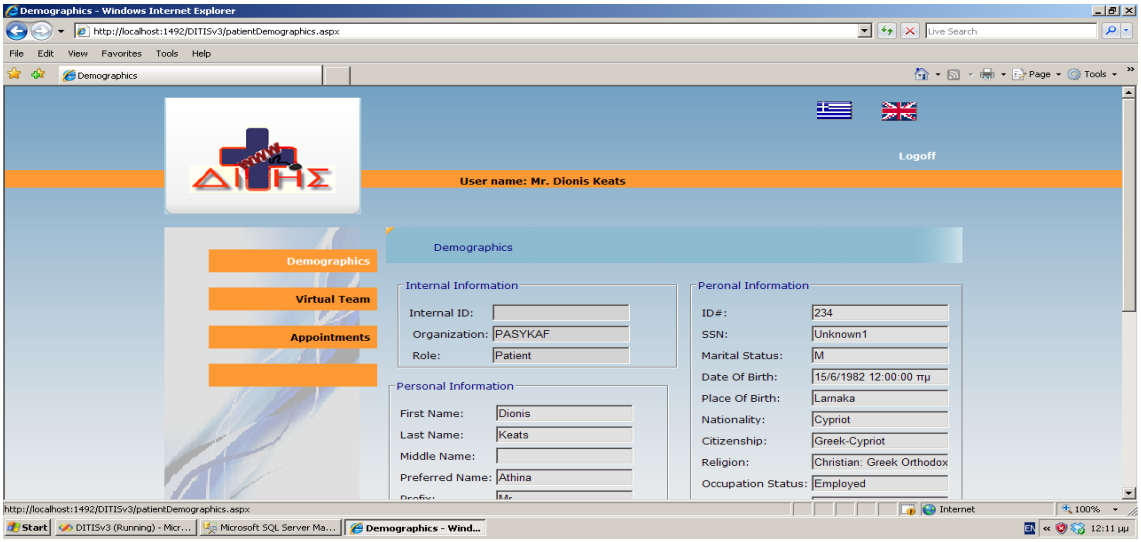
6.4.3.1. Σελίδα Welcome

Σχήμα6.24.

Όταν ο χρήστης που θα συνδεθεί στο σύστημα μας είναι ασθενής οδηγείτε σε μια σελίδα που καλωσορίζει τον χρήστη και από εκεί μπορεί να διαλέξει από το μενού να δει είτε την εικονική ομάδα που είναι υπεύθυνη για αυτόν μέσο της σελίδας patientVT.aspx. Ακόμη μπορεί να δει τα ραντεβού του από την σελίδα patientApp.aspx και τέλος μπορεί να δει τα δημογραφικά του στοιχεία και να τα αλλάξει.

6.4.3.2. Σελίδα Demographics

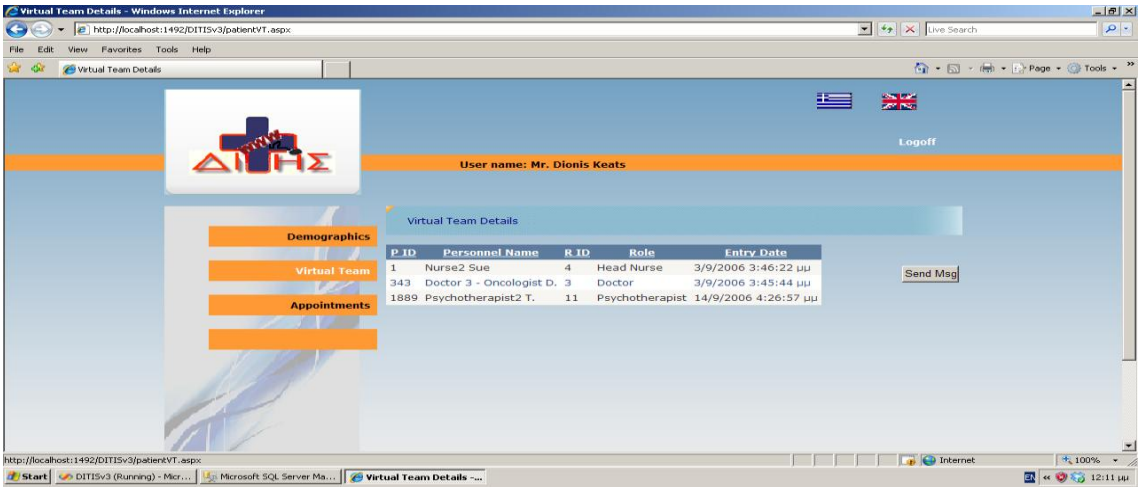
Σχήμα6.25.



Στην σελίδα αυτή ο χρήστης μπορεί να δει τα προσωπικά του στοιχεία και μπορεί να τα τροποποιήσει. Αυτό γίνεται με την βοήθεια των stored procedures sp_fc_select_PERSON_SINGLE και sp_fc_rpt_select_PERSON.

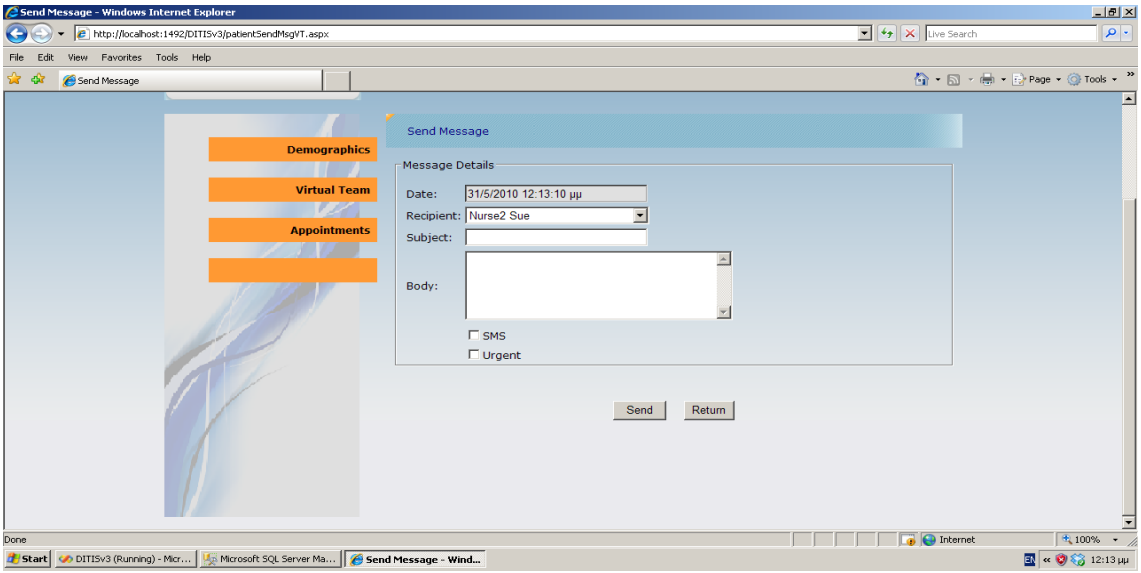
6.4.3.3. Σελίδες Virtual Team

Σχήμα6.26.



Ακόμη ένας ασθενής μπορεί να δει τα μέλη από τα οποία απαρτίζετε η εικονική του ομάδα(γίνεται χρήση του sp_fc_select_HCT_ENTRIES). Πατώντας το κουμπί Send Msg ο χρήστης μεταφέρετε στην σελίδα patientSendMsg.aspx.

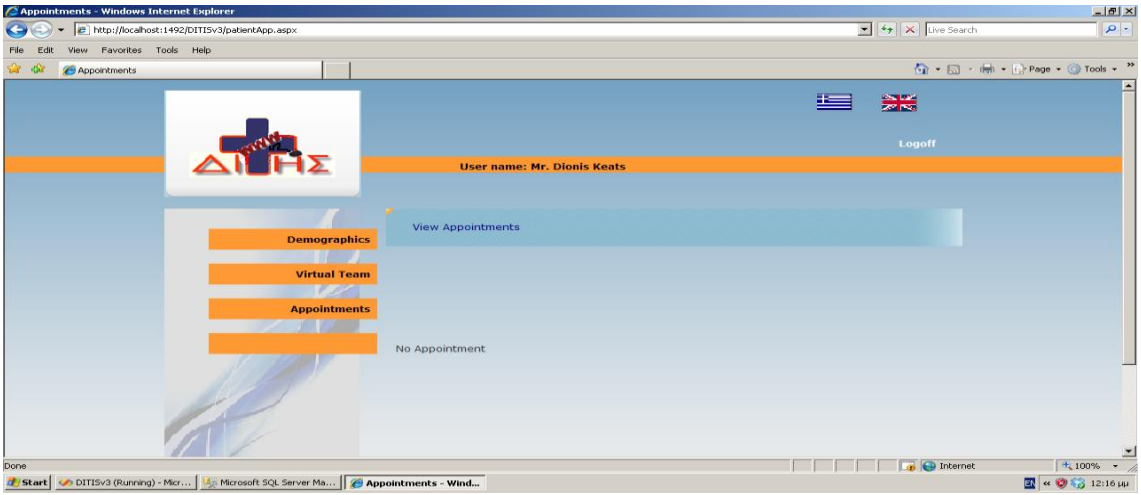
Σχήμα6.27.



Στην σελίδα αυτή ο χρήστης μπορεί να στήλη μήνυμα σε κάποιο από τα μέλη της εικονικής του ομάδας. Μετά την ολοκλήρωση της λειτουργίας αυτή το σύστημα βγάζει ενημερωτικό μήνυμα προς τον χρήστη. Πατώντας το κουμπί Return αλλά και μετά την ολοκλήρωση της αποστολής ο χρήστης μεταφέρετε στην προηγούμενη σελίδα.

6.4.3.4. Σελίδα Appointments

Σχήμα6.28.



Στην σελίδα patientApp.aspx ο χρήστης μπορεί να δει αν έχει κάποιο ραντεβού.

Κεφάλαιο 7

Αξιολόγηση

7.1. Αξιολόγηση 87

7.2. Πίνακες Αξιολόγησης 88

7.1. Αξιολόγηση

Η αξιολόγηση μιας διαδικτυακής εφαρμογής είναι η καταμέτρηση της επίδοσης, της αποδοτικότητας και της ευχρηστίας. Ένας οργανισμός, όταν αναπτύσσει μια διαδικτυακή εφαρμογή, προτού την δημοσιεύσει στο διαδίκτυο, πρέπει να είναι σε θέση να γνωρίζει αν η εφαρμογή αυτή ανταποκρίνεται στους στόχους του οργανισμού.

Όσον αφορά την αξιολόγηση της δικής μας εφαρμογής δεν θα δώσουμε έμφαση στην ικανοποίηση του χρήστη όσον αφορά τις λειτουργίες που του παρέχει η εφαρμογή άλλα θα επικεντρώσουμε την προσοχή μας στην ευχρηστία.

Ευχρηστία είναι η δυνατότητα ενός προϊόντος/ συστήματος ή υπηρεσίας που χρησιμοποιείται από καθορισμένους χρήστες με καθορισμένους στόχους, υπό καθορισμένες συνθήκες χρήσης να είναι αποτελεσματικό (effectiveness), αποδοτικό (efficiency) και να παρέχει υποκειμενική ικανοποίηση (satisfaction) στους χρήστες του.

Η ευχρηστία είναι σημαντική παράμετρος στην αξιολόγηση εφαρμογών διαδικτύου. Δυστυχώς όμως συχνά δεν λαμβάνει την δέουσα προσοχή, αν και είναι σημαντική όσο αφορά την επιτυχία μιας διαδικτυακής εφαρμογής. Η καλή ευχρηστία του συστήματος είναι ζωτικής σημασίας όσο αφορά την επιτυχία μιας διαδικτυακής εφαρμογής.
Ο σχεδιαστής ενός συστήματος είναι φυσικό να χρησιμοποιεί με ευκολία το σύστημα, να λατρεύει τον σχεδιασμό και να καταλαβαίνει τον σκοπό και στόχο του συστήματος. Ένας χρήστης όμως;

Επομένως πριν αναρτήσουμε μια διαδικτυακή εφαρμογή σε κοινή χρήση πρέπει να γίνεται αξιολόγηση της ευχρηστίας. Τα αποτελέσματα αυτής της αξιολόγησης μπορούν να αποκαλύψουν προβλήματα που έχουν να κάνουν με ελλειπή ή πολύπλοκο σχεδιασμό. [11]

Για να αξιολογήσουμε επομένως την εφαρμογή μας εφαρμόσαμε κάποιες δοκιμές ευχρηστίας(τεχνικές). Συγκεκριμένα δώσαμε στους χρήστες που χρησιμοποίησαν την εφαρμογή μας ένα ερωτηματολόγιο(Παράρτημα Α) που αφορά την διαπροσωπεία για να συμπληρωθεί μετά από την πρώτη επαφή με το σύστημα. Ακόμη βάλαμε τους χρήστες να εκτελέσουν συγκεκριμένες λειτουργίες και μετρήσαμε τον χρόνο που χρειάστηκε ο κάθε χρήστης για την υλοποίηση τους.

7.2. Πίνακες Αξιολόγησης

Α. Σχετικά με τον αξιολογητή

Ερωτήσεις	Αξιολ 1	Αξιολ 2	Αξιολ 3	Αξιολ 4	Αξιολ 5	Αξιολ 6	Αξιολ 7	Αξιολ 8	Αξιολ 9	Αξιολ 10
1.	15λ	10λ	20λ	15λ	10λ	15λ	8λ	20λ	12λ	24λ
2.	ναι	ναι	όχι	όχι	ναι	ναι	ναι	όχι	ναι	όχι
3.	ναι	ναι	όχι	όχι	ναι	ναι	ναι	όχι	ναι	όχι
4.	όχι	όχι	όχι	όχι	όχι	όχι	όχι	όχι	όχι	όχι

Β. Αποδοτικότητα

Ερωτήσεις	Αξιολ 1	Αξιολ 2	Αξιολ 3	Αξιολ 4	Αξιολ 5	Αξιολ 6	Αξιολ 7	Αξιολ 8	Αξιολ 9	Αξιολ 10
1.	4	3	4	4	4	3	4	3	3	4
2.	3	2	4	3	3	4	4	3	4	3
3.	-	-	-	-	-	-	-	-	-	-

Όσον αφορά την ερώτηση 3 οι χρήστες μας δεν μπορούσαν να γνωρίζουν αφού η εφαρμογή όταν την αξιολόγησαν έτρεχε στον τοπικό web server που προσφέρει το Visual Studio 2005 και μπορεί να χρησιμοποιηθεί για να φιλοξενηθούν ASP.NET εφαρμογές κατά την ανάπτυξη και δοκιμή τους. ο φυλλομετρητής που χρησιμοποίησαν όλη είναι ο Internet Explorer.

Γ. Πλοήγηση

Ερωτήσεις	Αξιολ 1	Αξιολ 2	Αξιολ 3	Αξιολ 4	Αξιολ 5	Αξιολ 6	Αξιολ 7	Αξιολ 8	Αξιολ 9	Αξιολ 10
1.	5	5	4	5	5	5	4	5	5	4
2.	4	3	3	4	3	4	4	3	4	4
3.	5	5	4	5	4	4	5	4	3	3

Δ. Σχεδίαση Ιστοσελίδας

Ερωτήσεις	Αξιολ 1	Αξιολ 2	Αξιολ 3	Αξιολ 4	Αξιολ 5	Αξιολ 6	Αξιολ 7	Αξιολ 8	Αξιολ 9	Αξιολ 10
1.	4	5	4	4	5	5	5	5	5	4
2.	4	3	4	3	3	4	5	5	4	3
3.	5	5	5	4	5	5	4	5	5	5
4.	5	5	5	5	5	5	5	5	5	5
5.	1	1	1	1	1	1	1	1	1	1

Ε. Φόρμες

Ερωτήσεις	Αξιολ 1	Αξιολ 2	Αξιολ 3	Αξιολ 4	Αξιολ 5	Αξιολ 6	Αξιολ 7	Αξιολ 8	Αξιολ 9	Αξιολ 10
1.	3	3	4	4	4	4	4	3	5	4
2.	5	4	5	5	5	4	4	4	5	5
3.	5	5	5	5	4	4	4	4	4	5

Ζ. Λειτουργίες Εφαρμογής

Ερωτήσεις	Αξιολ 1	Αξιολ 2	Αξιολ 3	Αξιολ 4	Αξιολ 5	Αξιολ 6	Αξιολ 7	Αξιολ 8	Αξιολ 9	Αξιολ 10
1.	4	4	4	5	4	4	4	4	4	4
2.	4	5	4	5	4	4	5	4	5	4
3.	1	1	1	1	1	1	1	1	1	1
4.	5	5	5	5	5	5	5	5	5	5
5.	4	5	5	4	4	4	5	5	5	4
6.	4	4	4	3	4	4	4	4	4	4

Η. Συνολική εκτίμηση του συστήματος

Γενικά τόσο η αξιολογητές που έχουν εμπειρία με ανάπτυξη οποιασδήποτε εφαρμογής όσο και αυτοί που δεν έχουν πιστεύουν ότι είναι αρκετά καλή και προσεγμένη δουλεία. Όμως πολλά πράγματα όπως την ταξινόμηση δεν γνώριζαν την ύπαρξη τους με αποτέλεσμα να μην εκμεταλλευτούν τα προτερήματα αυτών των λεπτομερειών όσον αφορά την υλοποίηση.

Η ικανοποίηση του χρήστη είναι αρκετή από ότι μας είπαν τα άτομα που αξιολόγησαν την εφαρμογή μας αφού τα χρώματα είναι ζωντανά χωρίς όμως να ενοχλούν τα μάτια τους με την πολύ ώρα. Ακόμη μας αναφέρουν πως πολλές λειτουργίες είναι εύκολες στην χρήση τους αφού ο χρήστης μόνο ένα κουμπί έχει να πατήσει.

Κάποιοι μας επισυνάπτουν πως είναι εύκολη ακόμη και για κάποιον που δεν είναι γνώστης του θέματος αλλά καλό θα ήταν να υπάρχει help για κάθε οθόνη για να μπορεί ο χρήστης να βλέπει τον ακριβή τρόπο λειτουργίας χωρίς να χρειάζεται να τον ανακαλύπτει από μόνος του.

Στα πλαίσια της αξιολόγησης θεωρήθηκε σημαντικό να πάρουμε και την άποψη ατόμων από το PASYKAF. Τα άτομα αυτά επειδή έχουν έρθει σε επαφή με το DITISv1και τον χρησιμοποιούν θα μπορούν να μας δώσουν μια πιο ακριβή αξιολόγηση και γιατί όχι να εντοπίσουν πιο εύκολα και μειονεκτήματα της εφαρμογής μας.

Συγκεκριμένα συναντήσαμε την κυρία Βαρβάρα, ένα άτομο αρκετά έμπειρο όσον αφορά την χρήση του συστήματος DITIS. Με την βοήθεια της εντοπίσαμε κάποιες αλλαγές που χρειάστηκε να γίνουν στην εφαρμογή, ενώ παράλληλα σκιαγραφήσαμε μελλοντικές εργασίες που θα βοηθήσουν τον οργανισμό σε μεγάλο βαθμό.

Οι αλλαγές που έγιναν αφορούν τις λειτουργίες. Μια από αυτές είναι η προθήκη στην σελίδα sendMsgVT.aspx της επιλογής να μπορεί ο χρήστης να στείλει αυτόματα σε όλα τα μέλη της εικονικής ομάδας ένα κοινό μήνυμα.

Όσον αφορά το interface(τα χρώματα, μέγεθος γραμμάτων κ.α.) πήραμε θετικά σχόλια, πράγμα που δείχνει ότι μπορεί να χρησιμοποιηθεί η εφαρμογή μας στο μέλλον φτάνει να ενσωματωθούν σε αυτή οι λειτουργίες που τα άτομα του οργανισμού θεωρούν σημαντικές και απαραίτητες για την εργασία τους.

Κεφάλαιο 8

Συμπεράσματα

8.1. Συμπεράσματα 91

8.2. Μελλοντικές Εργασίες 92

8.1. Συμπεράσματα

Στα πλαίσια της διπλωματικής εργασίας, έγινε μια διαδικτυακή εφαρμογή για εξυπηρέτηση ασθενών και ιατρικών ομάδων στην μεταξύ τους ανταλλαγή πληροφοριών, για την διαχείριση της περίθαλψης ευπαθών ομάδων από απόσταση, χωρίς την μετακίνηση του ασθενή από το σπίτι του.

Κατά την διάρκεια αυτής της εφαρμογής βρέθηκαν και διορθώθηκαν διάφορα σχεδιαστικά λάθη τα οποία είναι αναμενόμενα κατά την ανάπτυξη οποιασδήποτε εφαρμογής, ενώ συνάμα τροποποιήθηκαν διάφορες λειτουργίες του συστήματος που αποδείχτηκαν ότι επιβράδυναν την λειτουργία του.

Συνάμα, κατά την διάρκεια της υλοποίησης δόθηκε ιδιαίτερη προσοχή στο να διατηρήσει η εφαρμογή έναν γενικό χαρακτήρα ο οποίος να μπορεί να ικανοποιήσει όλους τους χρήστες ανεξαρτήτως ηλικίας, εμπειρίας με την εφαρμογή ή παρόμοιες εφαρμογές κ.α. Επίσης έγινε προσπάθεια το σύστημα που ανεπτύχθητε να είναι όσο πιο ευέλικτο και φιλικό προς το χρήστη γίνεται ώστε να μπορεί ο χρήστης να το χρησιμοποιεί με ευκολία.
Η τελική μορφή του προγράμματος της παρούσης διπλωματικής εργασίας μπορούμε να ισχυριστούμε ότι αποτελεί ένα αρκετά εύχρηστο εργαλείο το οποίο βοηθάει στον συντονισμό και στην καλή λειτουργία μιας ιατρικής ομάδας.

Η αξιολόγηση που έγινε στο σύστημα μας έδειξε ότι η προσπάθεια είχε αρκετά καλά αποτελέσματα, μιας και οι απαντήσεις που πήραμε από τα ερωτηματολόγια ήταν κυρίως θετικές.

Από την άλλη μεριά όμως, φάνηκε πως υπάρχουν και κάποιες κοινές παρατηρήσεις(μειονεκτήματα) που εντόπισαν οι αξιολογητές. Αυτά τα προβλήματα θα μπορούν να διορθωθούν σε μελλοντικές εργασίες για την διόρθωση και ανάπτυξη του παρόντος συστήματος έτσι ώστε να μπορεί να δημοσιευθεί και να χρησιμοποιηθεί επίσημα.

Η όλη εργασία ήταν μια πραγματική εμπειρία που βοήθησε στην πρακτική εφαρμογή του θεωρητικού μέρους των ακαδημαϊκών μαθημάτων που έχω διδαχθεί μέσα στα πλαίσια του ακαδημαϊκού προγράμματος σπουδών μου και που σίγουρα με έμαθε να ξεπερνώ πιθανά εμπόδια που μπορεί να συναντήσω.

Για την διεκπεραίωση του έργου χρησιμοποιήθηκαν κάποια εργαλεία όπως : Microsoft Visual Studio 2005, ASP.NET, Sql Server 2005, Visual Basic 2005, Microsoft word κ.α.

8.2. Μελλοντική εργασία

Οι επεκτάσεις και οι βελτιώσεις που μπορούν να γίνουν στο σύστημα μας, έτσι ώστε αυτό να γίνει ολοκληρωμένο και να μπορεί να ανταγωνιστεί άλλες υλοποιήσεις παρόμοιων συστημάτων που βρίσκονται τόσο στο διαδίκτυο όσο και σε μορφή λογισμικού είναι αρκετές. Μπορούμε ακόμη να κάνουμε πρωτοποριακές επεκτάσεις που θα κάνουν την εφαρμογή να ξεχωρίζει και να έχει προβάδισμα από άλλες. Ποιο κάτω θα δώσουμε κάποιες μελλοντικές εργασίες που μπορούν να γίνουν.

Η πρώτη αλλαγή που είναι αρκετά σημαντικό να γίνει είναι η εγγραφή ακόμη ενός css αρχείου το οποίο θα χρησιμοποιούμε για τις συσκευές με μικρότερη οθόνη από 600px πλάτος, όπως τα κινητά.

Το σύστημα όπως προαναφέρθηκε μπορεί να επεκταθεί. Μια ακόμη μελλοντική επέκταση του συστήματος θα ήταν να ενσωματωθεί ένας μηχανισμός encryption των δεδομένων που έρχονται από τη βάση, ώστε να μην μπορεί κάποιος να τα υποκλέψει.

Ακόμη καλό θα ήταν όπως φάνηκε από τις αξιολογήσεις να ενσωματωθεί στο σύστημα οδηγός χρήσης για κάθε σελίδα.

Μια έξυπνη και πρωτοποριακή επέκταση του συστήματος θα ήταν η ενσωμάτωση αισθητήρων στο σπίτι του ασθενή για καταγραφή π.χ. καρδιογραφήματος.

Όσον αφορά τις λειτουργίες μένει στο σύστημα να ενσωματωθούν τα Reports και Calendar. Με την ενσωμάτωση των λειτουργιών αυτών θα προσφέρουμε στον χρήστη μεγάλη άνεση και επιπλέον ευκολία αφού θα μπορεί να εκτυπώνει ανά πάσα στιγμή εκτύπωσης οποιουδήποτε παραστατικού, και θα υλοποιείτε από το σύστημα για ότι θέλει. Ακόμη θα πρέπει να ενσωματωθούν οι λειτουργίες που αφορούν τα metigation και τα equipments.

Τέλος, μετά την συνάντηση μας με την κυρία Βαρβάρα εντοπίστηκαν και κάποιες άλλες μελλοντικές εργασίες που θεωρούμε σημαντικές. Ουσιαστικά χρειάζεται να ενσωματωθεί μια λειτουργία που να αφορά τις συνταγές των γιατρών. Μέσο αυτής θα είναι κατοχυρωμένοι οι υπόλοιποι specialists ότι δεν έδωσαν κάτι άλλο στον ασθενή.

Ακόμη, θα πρέπει να ενσωματωθεί στην εφαρμογή stock control όσον αφορά τα equipments. Μέσο αυτού θα μπορεί ο χρήστης να δει αν υπάρχει στον οργανισμό κάποιο equipment(σε οποιαδήποτε πόλη) και αν χρειαστεί να μπορεί να το κρατήσει.

Μια άλλη έξυπνη λειτουργία θα ήταν η ενσωμάτωση GPS. Με αυτό τον τρόπο θα μπορεί εύκολα το προσωπικό να βρει που μένει ο ασθενής, ενώ παράλληλα θα μπορεί και ο οργανισμός να μετρήσει την απόσταση που διάνυσε το προσωπικό για τον υπολογισμό της βενζίνης.

Βιβλιογραφία

[1] Δημήτρης Κουτσούρης & Κωνσταντίνα Νικήτα, "Εργαστήριο Βιοϊατρικής Τεχνολογίας", Εθνικό Μετσόβιο Πολυτεχνείο, σελ. 81-115

[2] Michael A. Kittel & Geoffrey T. LeBlond, "ASP.NET Cookbook", O'REILLY, 2004

[3] Michael Halvorson, "Microsoft Visual Basic 2005 Βήμα Βήμα", Κλειδάριθμος, 2006

[4] Panayiotis Andreou, "DITIS Component Description", 2006

[5] Panayiotis Andreou, "A secure XML-based synchronization scheme for mobile devices", 2006

[6] Stephen Walther, "ASP.NET", SAMS, 2004

[7] Scott Mitchell, "Teach Yourself ASP.NET 2.0 in 24 Hours", SAMS, 2006

[8] Jesudas Chinnathampi (Das), Fabio Claudio Ferrachiati, James Greenwood, John Kauffman, Brian Matsik, Eric N. Mintz, Jan D. Narkiewicz, Kent Tegels, John West & Donald Xie, "Beginning ASP.NET Databases using VB.NET", Wrox, 2002

[9] http://www.webcredible.co.uk/user-friendly-resources/web-usability/basics.shtml

[10] http://tc.eserver.org/19272.html

[11] http://www.eeei.gr/interbiz/articles/useit.htm

[12] http://msdn.microsoft.com, http://www.microsoft.com

[13] http://en.wikipedia.org/wiki/ASP.NET

[14] http://en.wikipedia.org/wiki/Microsoft_Visual_Studio

[15] http://en.wikipedia.org/wiki/Microsoft_SQL_Server

[16] www.asp.net

Παράρτημα Α

Α1. Ερωτηματολόγιο διαπροσωπείας χρήστη

Παρακαλώ όπου θεωρείτε απαραίτητο αξιολογήστε με βάση την κλίμακα 1(καθόλου) - 5(πολύ)

Α. Σχετικά με τον αξιολογητή

1. Για πόσο χρόνο συνολικά ήλθατε σε επαφή με το σύστημα;……
2. Προσωπική εμπειρία ανάπτυξης παρόμοιας εφαρμογής ……..…….
3. Προσωπική εμπειρία ανάπτυξης λογισμικού γενικότερα……
4. Προσωπική εμπειρία χρήσης παρόμοιας εφαρμογής ……..……

Β. Αποδοτικότητα

1. Φορτώνεται γρήγορα η διαδικτυακή εφαρμογή(ιστοσελίδα) στον φυλλομετρητή; ……..
2. Χρειάζεται να μετακινηθείτε κατά μήκος της σελίδας για να τη δείτε ολόκληρη; (scroll down) ……
3. Η διαδικτυακή εφαρμογή ανοίγει σε όλους τους φυλλομετρητές που χρησιμοποιείται;……

Γ. Πλοήγηση

1. Υπάρχουν σε κάθε σελίδα οι επιλογές πλοήγησης;……
2. Βρίσκετε εύκολα πώς να εκτελέσετε την λειτουργία που θέλετε;……..
3. Οι συνδέσμοι για τις επιλογές πλοήγησης είναι ξεκάθαροι; …….

Δ. Σχεδίαση Ιστοσελίδας

1. Οι συμβολοσειρές γραμμάτων και τα μεγέθη που χρησιμοποιούνται είναι εύκολα στο να διαβαστούν; ……..…….....

2. Υπάρχει καλή ποιότητα εικόνων και φωτογραφιών;……..
3. Κατά πόσο τα χρώματα που χρησιμοποιήθηκαν είναι ξεκούραστα για το χρήστη;……
4. Ο συνδυασμός του background και των χρωμάτων είναι βοηθητικός για το χρήστη;……
5. Ήταν υπερβολική η χρήση χρώματος (αριθμός χρωμάτων) που χρησιμοποιήθηκαν;…….

Ε. Φόρμες

1. Υπάρχει διάκριση μεταξύ προαιρετικών και υποχρεωτικών πεδίων;…….
2. Υπάρχει λογική διάταξη πεδίων;…….
3. Υπάρχει έγκαιρος έλεγχος εσφαλμένων τιμών;……..

Ζ. Λειτουργίες Ιστοσελίδας

1. Πόσο εύκολα μπορείτε να βρείτε πως εκτελείτε μια λειτουργία; …….
2. Η δομή και η οργάνωση των λειτουργιών είναι λογική;……....
3. Υπάρχουν οδηγίες για τη χρήση κάθε μιας;……..
4. Η πρόσθεση, διαγραφή ή τροποποίηση γίνεται εύκολα;……...
5. Γενικά η εκμάθηση της λειτουργίας του συστήματος είναι εύκολη ή δύσκολη;…….
6. Η εκμάθηση προχωρημένων-ειδικών λειτουργιών είναι εύκολη;…….

Η. Συνολική εκτίμηση του συστήματος

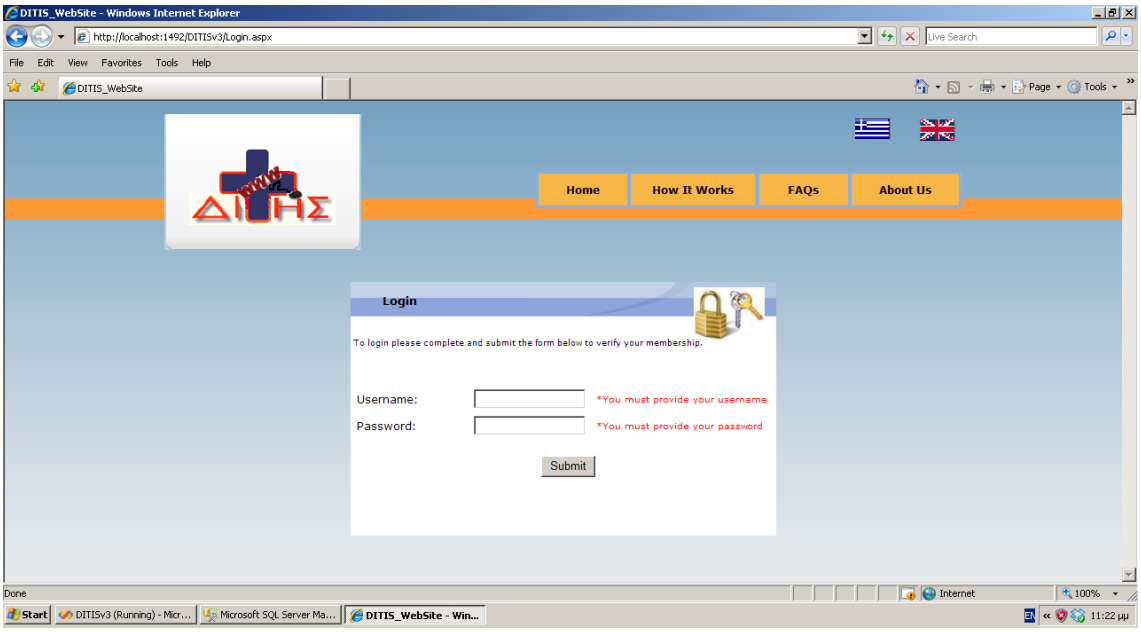
1. Γενική εκτίμηση:……..
2. Ικανοποίηση που παρέχει η χρήση του:……..
3. Πόσο εύκολη θα είναι κατά τη γνώμη σας η χρήση του;…….

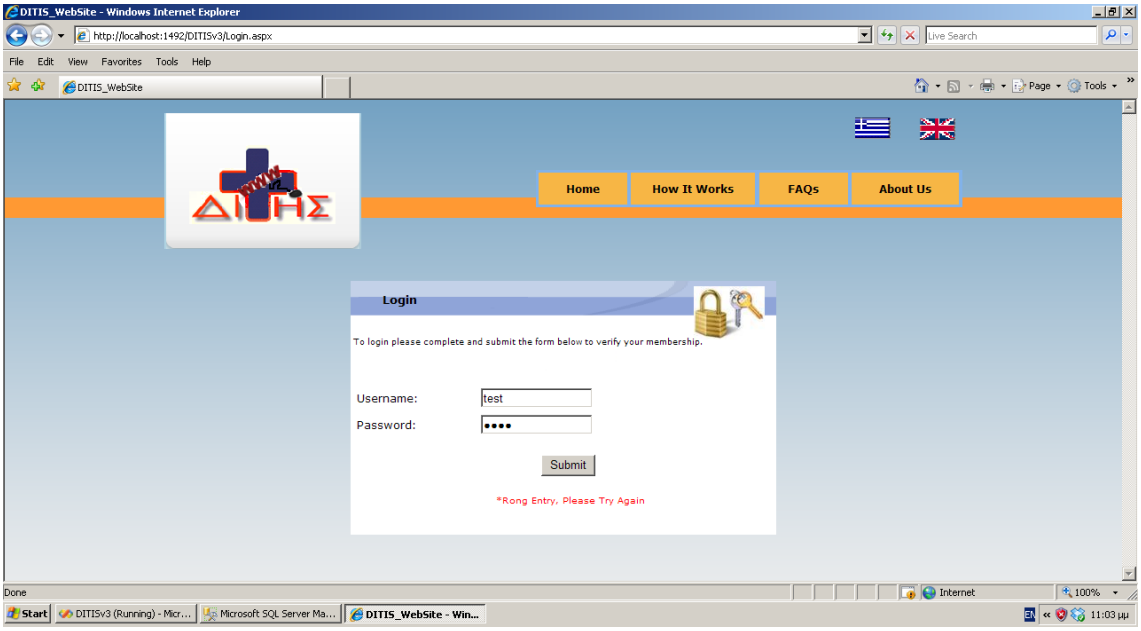
Παράρτημα Β

Β.1. Σενάριο χρήσης

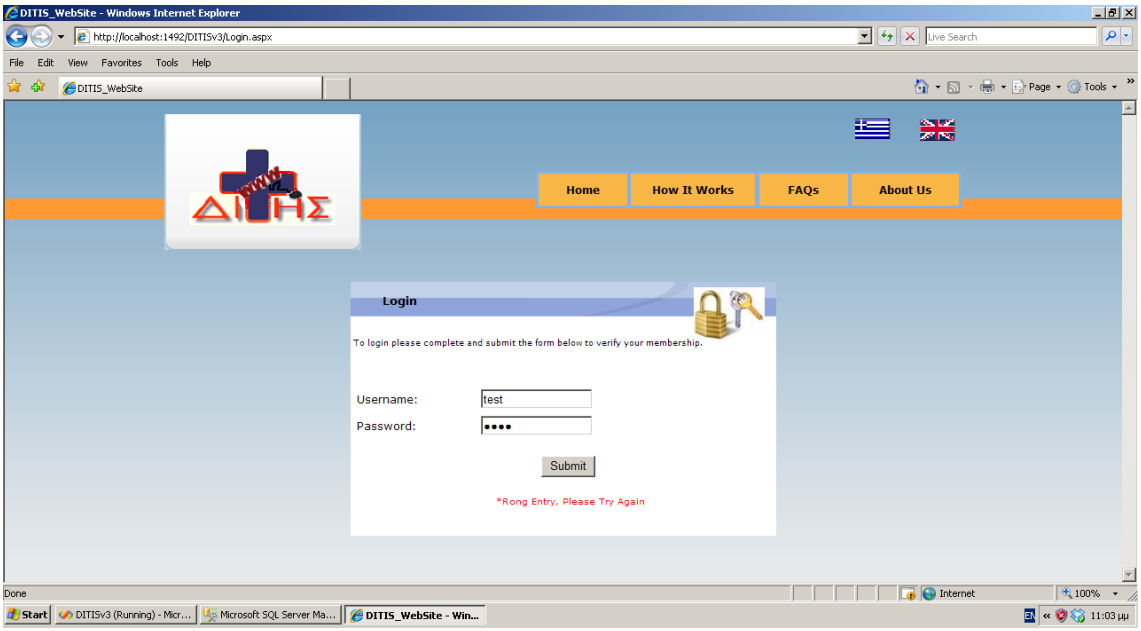
Έλεγχος και αποστολή μηνυμάτων

Αρχικά ο χρήστης θα πρέπει να κάνει login στο σύστημα δίνοντας τα σωστά στοιχεία (username, password). Στην περίπτωση που παραλείψει κάποιο από αυτά ή δώσει λάθος στοιχεία εισαγωγής τότε βγαίνουν στην οθόνη μηνύματα σφάλματος.

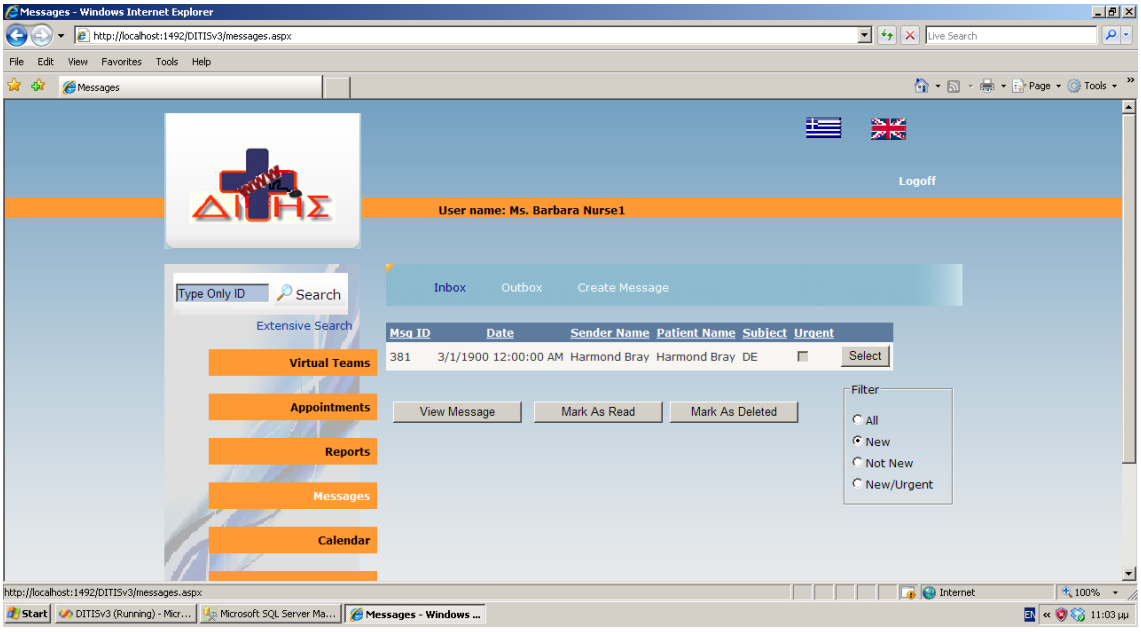




Μόλις ο χρήστης δώσει τα σωστά στοιχεία τότε γίνεται εισαγωγή του στο σύστημα. Και στην συνέχεια μπορεί να διαλέξει από την οθόνη ποια λειτουργία επιθυμεί να εκτελέσει.



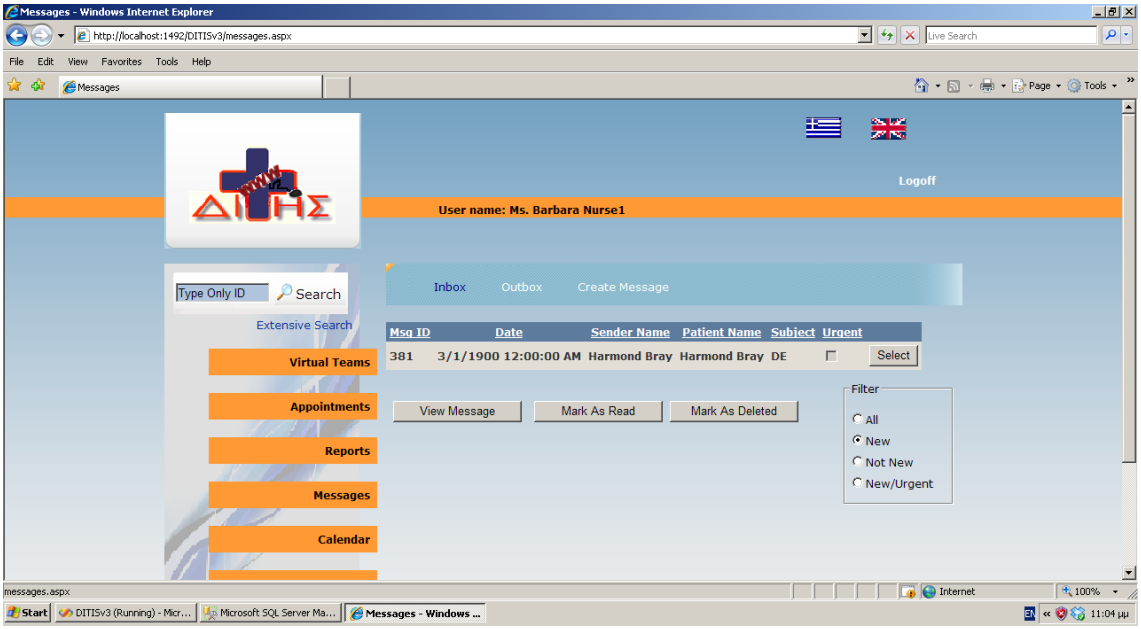
Στην περίπτωση που θέλει να δει τα μηνύματα του, τότε θα πατήσει το Messages από το μενού επιλογών και θα εμφανιστεί η πιο κάτω οθόνη.



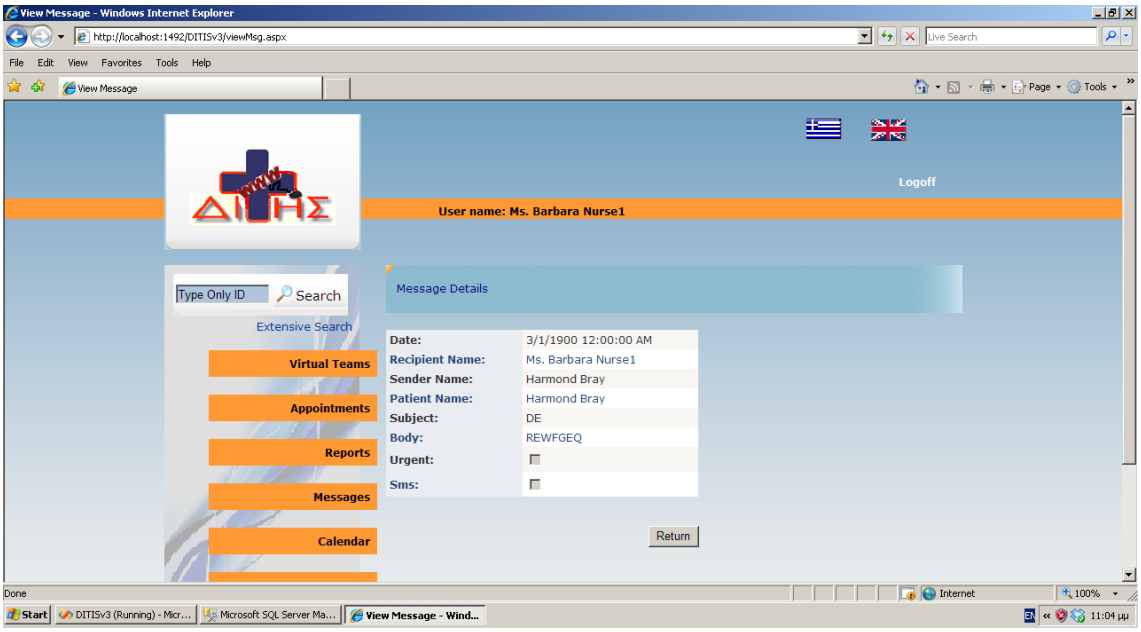
Στα Messages μπορεί ο χρήστης να δει τόσο τα εισερχόμενα μηνύματα του όσο και τα εξερχόμενα. Ακόμη μπορεί να δημιουργήσει και να αποστείλει καινούργιο μήνυμα.

Οι επιλογές που έχει στα εισερχόμενα είναι, να δει όλα τα μηνύματα, τα καινούργια μόνο, τα διαβασμένα μόνο και τέλος τα καινούργια ή τα επείγον μηνύματα.

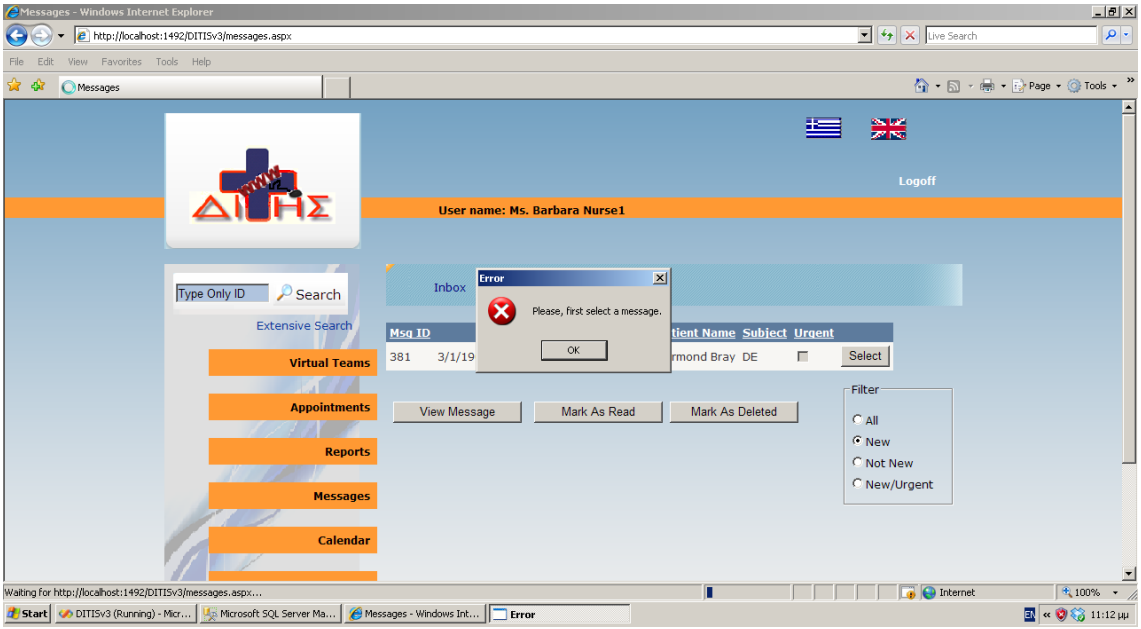
Στο σενάριο αυτό εμείς θέλουμε να δούμε τα καινούργια μηνύματα και να διαβάσουμε ένα από αυτά. Έτσι κάνουμε Filter/New και στην συνέχεια επιλέγουμε το μήνυμα που θέλουμε να διαβάσουμε. Τέλος πατούμε το κουμπί View Message.



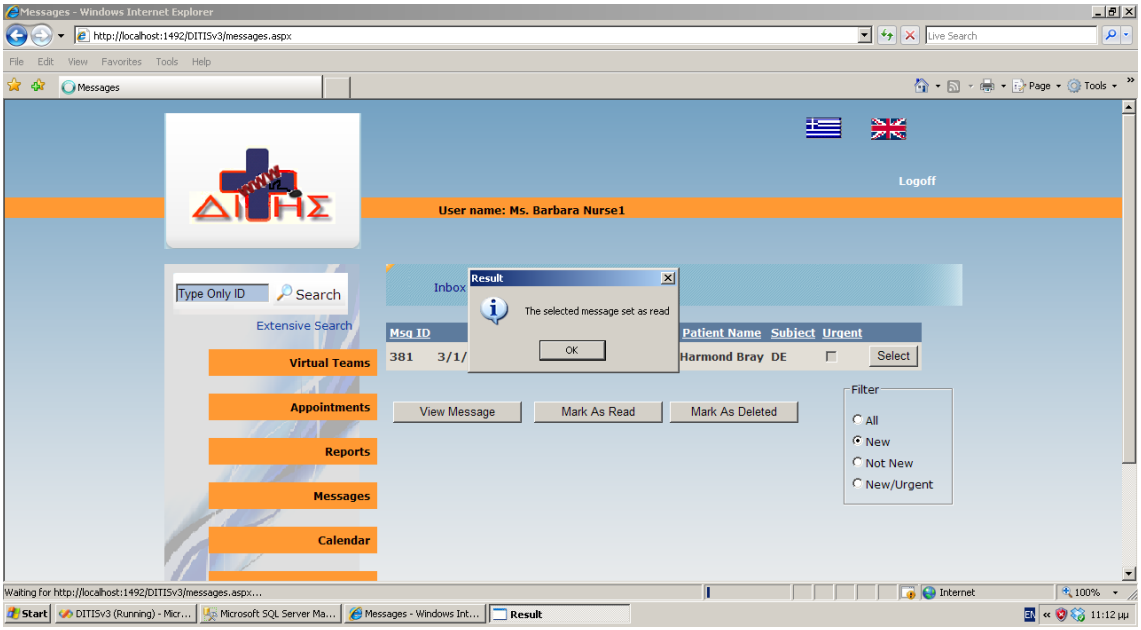
Στην συνέχεια εμφανίζεται μια οθόνη με τα στοιχεία του μηνύματος που επιλέξαμε να δούμε. Για να επιστρέψουμε πίσω ή θα πατήσουμε το κουμπί Return ή θα πάμε πάλι από το μενού και την επιλογή Messsages.



Πατώντας το κουμπί Return επιστρέφουμε στην προηγούμενη φόρμα. Εκεί μπορούμε να θέσουμε το μήνυμα που μόλις είδαμε ως διαβασμένο, για να μην εμφανίζεται στα νέα μηνύματα μας πλέων.

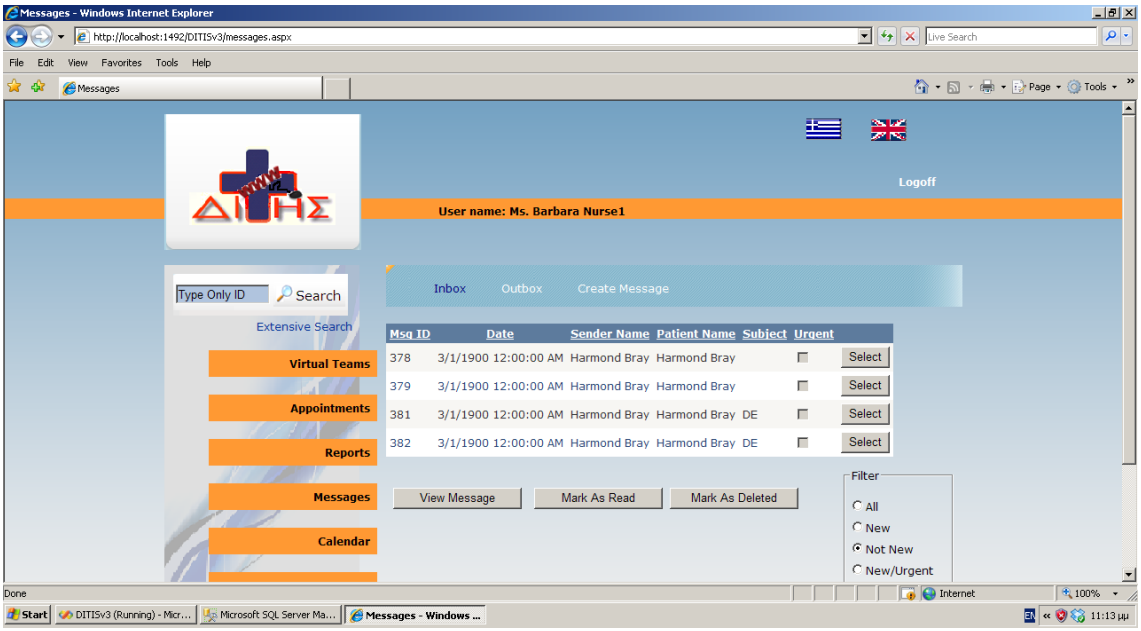


Στην περίπτωση που δεν επιλέξουμε μήνυμα και πατήσουμε οποιοδήποτε από τα κουμπιά βγαίνει ένα μήνυμα λάθους. Έτσι και στην πιο πάνω περίπτωση που πατήσαμε το κουμπί Mark As Read βγήκε αυτό το μήνυμα.



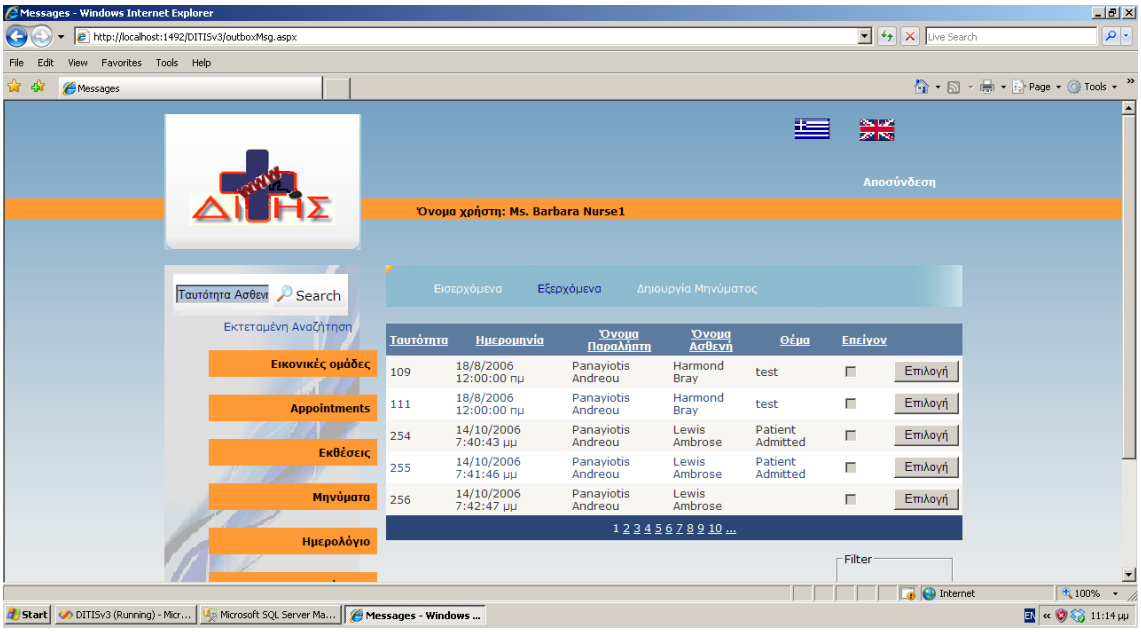
Όταν επιλέξουμε ένα μήνυμα και μετά το κουμπί Mark As Read το επιλεγμένο μήνυμα τίθεται αυτόματα σαν διαβασμένο και ένα μήνυμα ενημέρωσης εμφανίζεται στον χρήστη.

Για να δούμε αν πράγματι εκτελέστηκε σωστά μπορούμε να πάμε στα Filter/Not New μηνύματα και θα δούμε ότι τώρα εμφανίζεται εκεί.

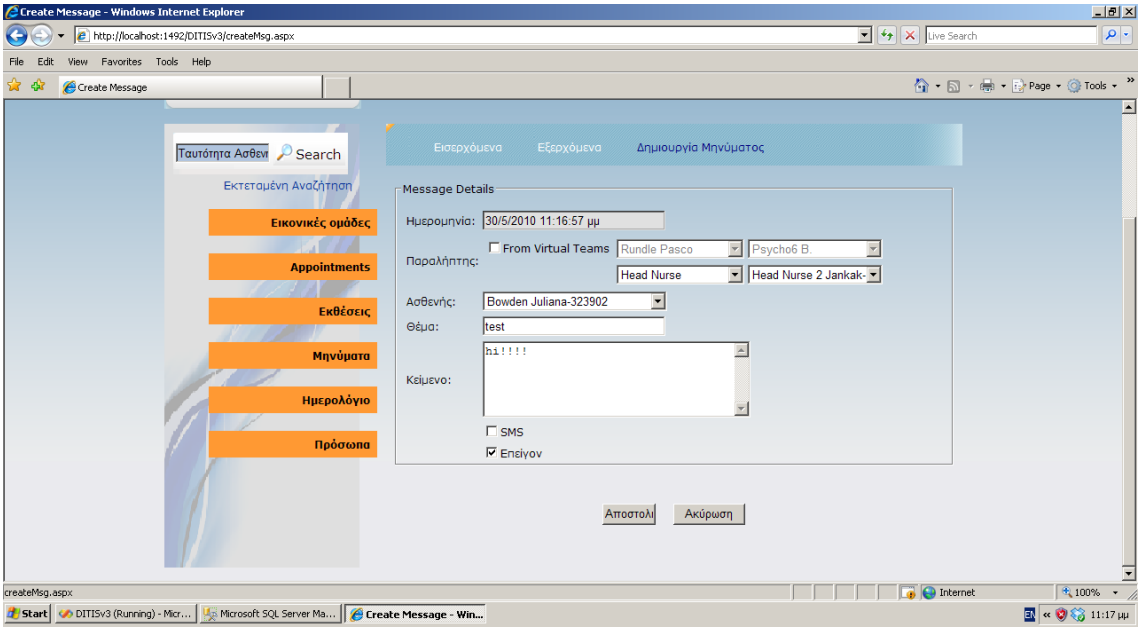


Στην περίπτωση που θέλουμε να αλλάξουμε γλώσσα σε όποιο σημείο της εφαρμογής μας και αν βρισκόμαστε μπορούμε να πατήσουμε στο αντίστοιχο σημαιάκι και τότε αυτόματα θα αλλάξει η γλώσσα τόσο για την οθόνη που βρισκόμαστε αλλά και για όλη την συνέχεια.

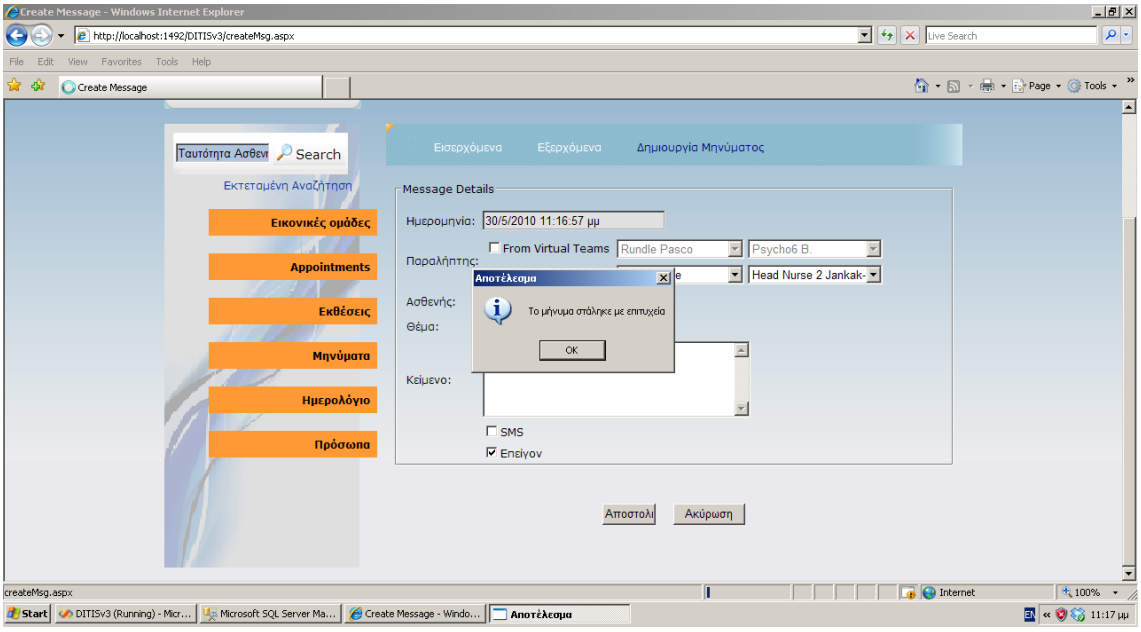
Πριν τη δημιουργία νέου μηνύματος πατήσαμε το ελληνικό σημαιάκι για να αλλάξουμε την γλώσσα από αγγλικά σε ελληνικά. Έτσι τώρα βλέπουμε στην οθόνη μας ελληνικά.



Πάμε στην επιλογή Δημιουργία Μηνύματος και ακολούθως δημιουργούμε ένα μήνυμα.



Πατώντας το κουμπί Αποστολή το μήνυμα μας στέλνεται και στην οθόνη μας εμφανίζεται ένα μήνυμα ενημέρωσης προς τον χρήστη.



Με τα πιο πάνω βήματα ολοκληρώσαμε με επιτυχία την λειτουργεί δημιουργίας και αποστολής μηνύματος.