



**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Εφαρμογή Κοινωνικού Περιεχομένου-
Ηλεκτρονική Συνταγογράφηση Φαρμάκων**

**Σάμιος Π. Θεόδωρος
Α.Μ 38026**

**Βουρδουμπας Β. Νικόλαος
Α.Μ 38267**

Εισηγητής: Δρ Παναγιώτης Γιαννακόπουλος, Καθηγητής

**ΑΘΗΝΑ
Φεβρουάριος 2017**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Εφαρμογή Κοινωνικού Περιεχομένου-
Ηλεκτρονική Συνταγογράφηση Φαρμάκων**

**Σάμιος Π. Θεόδωρος
Α.Μ. 38026**

**Βουρδουμπάς Β. Νικόλαος
Α.Μ. 38267**

Εισηγητής:

Δρ Παναγιώτης Γιαννακόπουλος, Καθηγητής

Εξεταστική Επιτροπή:

**Δρ Έλληνας Ιωάννης, Καθηγητής
Δρ Βελώνη, Αναστασία, Καθηγητής**

Ημερομηνία εξέτασης:

10 / 02 / 2017

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Οι κάτωθι υπογεγραμμένοι *Βουρδουμπάς Νικόλαος & Σάμιος Θεόδωρος*, του *Ευαγγέλου & Παναγιώτη (αντίστοιχα)* με αριθμούς μητρώου 38267 & 38026 (*αντίστοιχα*) φοιτητές του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μας, δηλώνουμε ότι ενημερωθήκαμε για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Πριν από όλα θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή μας Δρ. Γιαννακόπουλο Παναγιώτη που μας έδωσε την δυνατότητα να ασχοληθούμε με ένα τόσο ενδιαφέρον θέμα. Θα θέλαμε επίσης να τον ευχαριστήσουμε για την βοήθεια, την καθοδήγηση και τις χρήσιμες συμβουλές που μας παρείχε.

Ένα μεγάλο ευχαριστώ επίσης οφείλουμε στις οικογένειες μας που μας στήριξαν με κάθε τρόπο για την ολοκλήρωση της πτυχιακής μας εργασίας αλλά και σε όλη τη διάρκεια των σπουδών μας.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει σαν σκοπό την ανάπτυξη μίας καινοτόμου εφαρμογής μέσω της οποίας θα πραγματοποιείται μια ηλεκτρονική συνταγογράφηση. Η εφαρμογή αυτή μπορεί να εκτελεστεί από οποιαδήποτε κινητή συσκευή αλλά και με την χρήση ηλεκτρονικού υπολογιστή. Η κατασκευή της πραγματοποιήθηκε με τη χρήση του προγράμματος Visual Studio 2015 και της πλατφόρμας .NET, ενώ η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η VISUAL BASIC.

ABSTRACT

This paper has as a purpose the development of an innovative application through which will be made an electronic prescription. The application's construction has been held by the use of the Visual Studio 2015 program and the .NET platform. Also VISUAL BASIC has been used as a programming language.

ΠΕΡΙΕΧΟΜΕΝΑ

1	Εισαγωγή	14
2	Σκοπός και στόχοι εργασίας.....	15
2.1	Περιγραφή εφαρμογής.....	15
3	Δομή εργασίας.....	16
4	Ιστορική Αναδρομή.....	16
4.1	Microsoft.....	16
5	Η συνεισφορά της ανοικτής αρχιτεκτονικής της IBM.....	17
6	Γλώσσες προγραμματισμού	19
7	Κατηγοριοποίηση γλωσσών προγραμματισμού	20
7.1	Κατηγοριοποίηση σύμφωνα με τα επίπεδα που δουλεύουν στον υπολογιστή.....	21
7.1.1	Γλώσσες μηχανής.....	21
7.1.2	Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου.....	21
7.1.3	Γλώσσες υψηλού επιπέδου.....	22
7.1.4	Γλώσσες 4ης γενιάς	22
7.2	Κατηγοριοποίηση σύμφωνα το τρόπο οργάνωση προγράμματος.....	23
7.3	Κατηγοριοποίηση με βάση τον στόχο που έχει η γλώσσα.....	23
7.4	Κατηγοριοποίηση με βάση τον τρόπο που περιγράφουν το ζητούμενο αποτέλεσμα ..	24
8	Ιστορία γλωσσών προγραμματισμού	25
8.1	FORTRAN.....	25
8.2	COBOL.....	26
8.3	BASIC.....	26
8.4	PASCAL	27
8.5	C.....	27
8.6	JAVA	28
9	Δομημένος Προγραμματισμός	28
10	Προγραμματισμός οδηγούμενου από το γεγονός προγραμματισμού (object driven programming) και του οπτικός προγραμματισμός (visual programming).	29
11	Αντικειμενοστραφής Προγραμματισμός.....	30
12	Σύγκριση μεταξύ δομημένου και αντικειμενοστραφούς προγραμματισμού	31
13	Ανάλυση Αντικειμενοστραφούς Προγραμματισμού	31
14	Χαρακτηριστικές ιδιότητες αντικειμενοστραφούς προγραμματισμού	32
	Σάμιος Θεόδωρος	
	Βουρδουμπάς Νικόλαος	11

15	Εργαλεία δημιουργίας εφαρμογών στο λειτουργικό σύστημα Microsoft Windows.....	33
15.1	Τι είναι το .NET Framework?	33
15.2	Αρχιτεκτονική.....	34
16	Microsoft Visual Basic	35
17	Ολοκληρωμένο Περιβάλλον Εργασίας (Integrated Development Environment).....	35
18	Λύσεις και Έργα (Solutions and Projects).....	35
19	Εργαλεία Δοκιμών (Testing Tools).....	36
20	Σχεδιασμός και Μοντελοποίηση (Design and Modeling).....	36
21	Microsoft Visual Studio SDK (Software Development Kit)	36
22	Τύποι Δεδομένων στην Visual Basic	37
22.1	Δομές Ελέγχου	39
22.2	Δομές Επανάληψης	39
22.3	Η δομή for.....	39
22.4	Η δομή while	39
23	Πλατφόρμα Ανάπτυξης.....	40
24	Visual Studio 2015.....	40
25	Βάση Δεδομένων SQL2012.....	43
26	Πίνακες	43
26.1	Πίνακας Πελατών(Dbo.Customers).....	43
26.2	Πίνακας Γιατρών (Dbo.Iatroi).....	44
26.3	Πίνακας Φαρμάκων.....	44
26.4	Πίνακας Φαρμακείων.....	44
26.5	Πίνακας Αρίθμησης Συνταγών	45
26.6	Βασικός Πίνακας Συνταγών.....	45
26.7	Αναλυτικός Πίνακας Συνταγών.....	45
27	Γενικές Παρατηρήσεις Δεδομένων.....	46
27.1	Application DataSetEgasia	47
28	Γενικά σχόλια για τον κώδικα και τα CONTROLS στις φόρμες.....	48
29	Εφαρμογή για Mobile, Tablet & Desktop σε C# (Windows 10 μέσω του Store).....	49
29.1	Αρχική φόρμα	49
29.2	Φόρμα Ιατρού.....	50
29.3	Φόρμα Φαρμακοποιού	53
30	Εφαρμογή για Desktop σε Visual Basic (Windows 7 και προγενέστερες εκδόσεις).....	54

30.1	ΛΕΙΤΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ	54
30.2	Φόρμα συνταγογράφησης	56
30.3	Εκτέλεση Συνταγών	60
31	Διαδικτυακή Εφαρμογή για οποιαδήποτε συσκευή σε HTML – ASP.NET	62
31.1	ΑΡΧΙΚΗ WEB PAGE START.HTML	62
31.2	ΑΝΑΖΗΤΗΣΗ ΓΙΑΤΡΩΝ.....	63
31.3	ΑΝΑΖΗΤΗΣΗ ΦΑΡΜΑΚΕΙΩΝ ΜΕ ΠΕΡΙΟΧΗ.....	64
31.4	ΑΝΑΖΗΤΗΣΗ ΠΕΛΑΤΩΝ ΚΑΙ ΣΥΝΤΑΓΩΝ.....	65
32	ΠΑΡΑΡΤΗΜΑ Ι – ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ (C#)	67
32.1	Αρχική φόρμα – MainPage.xaml.cs.....	67
32.2	Φόρμα Ιατρού – DoctorPage.xaml.cs.....	68
32.3	Φόρμα Φαρμακοποιού – DrugstorePage.xaml.cs	71
33	ΠΑΡΑΡΤΗΜΑ Ι – ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ (VISUAL BASIC)	74
33.1	Πίνακας Κοινών Μεταβλητών	74
33.2	Custom Tables Queries	74
33.2.1	Insert Queries	74
33.2.2	Update Queries	74
33.2.3	ReceiptsMain Queries	74
33.2.4	ReceiptsDetails Queries	74
33.3	Συνολικός Κώδικας Αρχικής Φόρμας	75
33.4	Συνολικός Κώδικας Φόρμας Συνταγογραφήσεων	76
33.5	Συνολικός Κώδικας εκτέλεσης συνταγών.....	83
33.6	Επιμέρους Κώδικας Επιλογής Ιατρών	84
33.7	Καταχώρηση νέας συνταγής	86
34	ΠΑΡΑΡΤΗΜΑ Ι – ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ (HTML-ASP.NET)	87
34.1	GENERAL SCRIPT	87
34.2	ΑΝΑΖΗΤΗΣΗ ΓΙΑΤΡΩΝ ΜΕ ΟΝΟΜΑ.....	87
34.3	ΑΝΑΖΗΤΗΣΗ ΓΙΑΤΡΩΝ ΜΕ ΕΙΔΙΚΟΤΗΤΑ.....	88
34.4	ΑΝΑΖΗΤΗΣΗ ΦΑΡΜΑΚΕΙΩΝ ΜΕ ΠΕΡΙΟΧΗ.....	88
34.5	ΑΝΑΖΗΤΗΣΗ ΠΕΛΑΤΩΝ ΚΑΙ ΣΥΝΤΑΓΩΝ.....	89
34.6	ΕΚΤΕΛΕΣΗ ΣΥΝΤΑΓΗΣ	90

1 Εισαγωγή

Η εποχή μας μπορεί να χαρακτηριστεί μεταβιομηχανική. Οι επιτυχίες της Βιομηχανικής Επανάστασης μέσω της οποίας καταπολεμήθηκαν στον ανεπτυγμένο κόσμο η φτώχεια και οι κακουχίες, παύουν πια να συναρπάζουν. Πλέον όλος ο πλανήτης είναι ενθουσιασμένος με τα επιτεύγματα της σύγχρονης εποχής, μέρος της οποίας αποτελεί η πληροφορική.

Παρότι θα μπορούσαμε να πούμε πως δεν έχουμε ισομερή κατανομή των ευεργετημάτων, με κάποιες εταιρίες ή ακόμα και φυσικά πρόσωπα να εκμεταλλεύονται εξαιρετικά τις νέες συνθήκες δημιουργίας πλούτου, δεν μπορούμε να αμφισβητήσουμε την αλλαγή που έχει σημειωθεί. Η ανάπτυξη οφείλεται στο γεγονός ότι η νέα τεχνολογία προσφέρει τη δυνατότητα να πραγματοποιούνται κάθε είδους συναλλαγές με μεγάλη ταχύτητα και με σχετικά μικρό κόστος. Με αυτό τον τρόπο οι αγορές της πώλησης αγαθών και υπηρεσιών έχουν παγκοσμιοποιηθεί, με το αποτέλεσμα να μπορεί να συγκεντρώνεται μεγάλη ποσότητα πλούτου σε εταιρίες οι οποίες ξεχωρίζουν για τις καινοτομίες τους.

Επίσης, αξίζει να αναφερθεί πως η πληροφορική έχει μπει σε όλα τα σπίτια. Πολλές οικογένειες έχουν πρόσβαση στις τεχνολογίες του διαδικτύου, με άμεση πρόσβαση σε πληροφορίες, οι οποίες παλαιότερα ήταν απλησίαστες. Θα μπορούσαμε να ισχυριστούμε πως θεωρητικά έχουμε τις προϋποθέσεις για την απόλυτη δημοκρατία, δεδομένου ότι, ακόμα και ο χαμηλότερος εισοδηματικά νέος, μπορεί να γίνει εξειδικευμένος με θαυμαστές γνώσεις οι οποίες μπορούν να του αποφέρουν σημαντικές απολαβές στην παγκοσμιοποιημένη αγορά εργασίας.

Ένα ακόμα επίτευγμα της σύγχρονης εποχής αποτελούν τα κοινωνικά δίκτυα, μέσω των οποίων επιτυγχάνεται η ηλεκτρονική κοινωνικοποίηση των ατόμων. Παρόλη τη κριτική που θα μπορούσε να ασκήσει κάποιος σε αυτή την αλλαγή, η οποία απομακρύνει την επικοινωνία από την άμεση διαπροσωπική σχέση, δεν μπορεί να παραβλεφθεί πως άμεσα και χωρίς κόστος μπορούν να δημιουργηθούν σχέσης και δικτύωσης ή ακόμα και επιχειρηματικές επαφές.

2 Σκοπός και στόχοι εργασίας

Η εργασία αυτή είχε σαν σκοπό της την ανάπτυξη μίας καινοτόμου εφαρμογής μέσω της οποίας θα πραγματοποιείται ηλεκτρονική συνταγογράφηση. Η ιδέα για την ανάπτυξή της μας δημιουργήθηκε από την προτροπή του οικογενειακού μας ιατρού, ο οποίος ήταν απογοητευμένος από τη λειτουργία της επίσημης εφαρμογής ηλεκτρονικής συνταγογράφησης. Μας έδωσε όλη την απαιτούμενη πληροφόρηση, ενώ ταυτόχρονα πλοηγηθήκαμε στην υφιστάμενη εφαρμογή και ελέγξαμε το στήσιμο της. Η εφαρμογή αυτή κατασκευάστηκε με τη χρήση της πλατφόρμας .NET ενώ η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η VISUAL BASIC

2.1 Περιγραφή εφαρμογής

Η εφαρμογή έχει ως στόχο να εγκατασταθεί σε όλους τους γιατρούς του Ιατρικού Συλλόγου Αθηνών και με τη χρήση της να μπορεί ο καθένας από αυτούς να συνταγογραφήσει σε όλους τους δυνητικούς ασθενείς που θα έχει. Το αποτέλεσμα της συνταγογράφησης για την αποφυγή της περιττής γραφειοκρατίας δεν θα εκτυπώνεται, αλλά αποστέλλεται στο κινητό του πελάτη. Αυτός θα πηγαίνει στο πλησιέστερο φαρμακείο, και το φαρμακείο θα βρίσκει τη συνταγή που του έχει συνταγογραφηθεί, δίνοντάς του τα φάρμακα, και κλείνοντας την μη εκτελεσμένη συνταγή.

Για την υποστήριξη τις περιγραφόμενης μεθοδολογίας δεν χρησιμοποιήσαμε την υφιστάμενη εφαρμογή, αλλά φτιάξαμε μία εξολοκλήρου καινούργια standalone εφαρμογή. Τα δεδομένα που χρησιμοποιήσαμε πήραμε από ανοιχτές μορφές πληροφόρησης και διαφυλάξουμε τα ευαίσθητα στα προσωπικά δεδομένα.

Μιλώντας σε καθαρά θεωρητικό επίπεδο μπορούμε διαπιστώσουμε ότι η δική μας εφαρμογή μπορεί να λειτουργήσει και μόνη της σε περιπτώσεις όπου δεν υπάρχει πρόσβαση στην αντίστοιχη εφαρμογή του υπουργείου και να λειτουργήσει αυτή εναλλακτικά.

3 Δομή εργασίας

Στα επόμενα κεφάλαια θα αναλύσουμε τη δομή εργασίας. Την χωρίσαμε σε τμήματα με τέτοιο τρόπο έτσι ώστε να είναι πιο εύκολη η ανάγνωση και η εκτίμησή της. Τα τμήματα που έχουμε χωρίσει την εργασία μας είναι απαραίτητα για την καλύτερη κατανόηση της εργασίας μας, όμως δεν ήταν ίδια με τη δομή με την οποία την υλοποιήσαμε. Η συγκεκριμένη δομή βόλευε περισσότερο τον τρόπο της παρουσίασης της, οπότε την επιλέξαμε, ενώ στην ανάπτυξη της εφαρμογής ακολουθήσαμε διαφορετικά μονοπάτια τα οποία μας οδήγησαν στην εύρεση των τεχνολογιών που τελικώς χρησιμοποιήσαμε .

4 Ιστορική Αναδρομή

4.1 Microsoft

Η Microsoft είναι η εταιρεία κολοσσός λογισμικού που εδρεύει στις Ηνωμένες Πολιτείες της Αμερικής. Ιδρύθηκε το 1975 από τον Μπιλ Γκέιτς και τον Πωλ Άλεν, μέσω της ανάπτυξης της γλώσσας Basic.

Η Microsoft αναπτύσσει, κατασκευάζει, αδειοδοτεί και υποστηρίζει ένα ευρύ φάσμα προϊόντων λογισμικού για διάφορες υπολογιστικές συσκευές. Τα δημοφιλέστερα προϊόντα της είναι τα λειτουργικά συστήματα Microsoft Windows και η σειρά λογισμικού γραφείου Microsoft Office οι κονσόλες παιχνιδιών Xbox, το λογισμικό για κινητά τηλέφωνα (παλιά: Windows Mobile , Windows CE), και τώρα Windows Phone. Επίσης παράγει επίσης λογισμικό για επιχειρήσεις με σημαντικότερο τον SQL Management Studio ενώ παρέχει και υπηρεσίες ηλεκτρονικού ταχυδρομείου, διαδικτυακής αποθήκευσης, κοινωνικής δικτύωσης. Η εταιρεία έχει περίπου 9300 υπαλλήλους και έχει παρουσία σε 102 χώρες.

Κάποιοι θα μπορούσαν να πουν πως όλη η επιτυχημένη ιστορία βασίστηκε σε ένα τυχαίο γεγονός.

5 Η συνεισφορά της ανοικτής αρχιτεκτονικής της IBM

Όλη η επιτυχημένη ιστορία της Microsoft ξεκίνησε από την επιλογή της IBM να υποστηρίξει την ανοικτή αρχιτεκτονική στα προϊόντα που πουλούσε. Μέσω αυτής, δόθηκε με αυτό τον τρόπο το έναυσμα στην Microsoft να αποκτήσει την πελατειακή βάση της IBM και ταυτόχρονα να καταστήσει τα προϊόντα της γνωστά σε όλο τον κόσμο. Ας ξετυλίξουμε το κουβάρι αυτής της διάσημης ιστορίας.

Η Microsoft αρχικά είχε δημιουργήσει λογισμικό για τη γλώσσα προγραμματισμού BASIC, το οποίο και πώλησε στην Micro Instrumentation and Telemetry Systems (MITS) του Νέου Μεξικού, που τότε κατασκεύαζε τους υπολογιστές Altair. Το 1976 εμφανίζεται το λειτουργικό σύστημα CP/M, δημιούργημα του Γκάρι Κίλνταλ (Gary A. Kildall), το οποίο και καθιερώνεται ως πρότυπο το 1977. Ο Κίλνταλ δημιουργεί την εταιρεία Digital Research μαζί με τη σύζυγό του Ντόροθι, μέσω της οποίας αδειοδοτεί στις κατασκευάστριες εταιρίες το CP/M.

Η Microsoft λοιπόν δημιούργησε το λογισμικό για τη γλώσσα προγραμματισμού basic την οποία αρχικά έχει πουλήσει στο νέο Μεξικό στη συνέχεια όταν η IBM ήθελα ένα ανοιχτό λειτουργικό, η Microsoft το έδωσε με χαρά γνωρίζοντας είναι μία πολύ επιτυχημένη εμπορική στρατηγική.

Η IBM, η οποία ήταν μεγάλη εταιρεία εκείνη την εποχή, αποφάσισε να κατασκευάσει ένα προσωπικό υπολογιστή. Για το σκοπό αυτό χρειαζόταν ένα απλό και σταθερό λειτουργικό σύστημα. Απευθύνθηκε, όπως ήταν φυσικό, στον Κίλνταλ και στην εταιρεία του, αλλά δεν συμφώνησαν, οπότε τελικά απευθυνθήκανε στην Microsoft η οποία παρέδωσε στην IBM το λειτουργικό, που η εταιρία ονόμασε PC-DOS.

Η επιτυχία του IBM PC υπήρξε τεράστια και, όπως ήταν φυσικό, αυτό έδωσε τεράστια ώθηση στη Microsoft, η οποία, έχοντας το προηγούμενο της αποκλειστικής διάθεσης της BASIC στη MITS, φρόντισε να κρατήσει τα δικαιώματα του DOS, μη εκχωρώντας αποκλειστικότητα στην IBM.

Έτσι, όταν εμφανίσθηκαν οι "κλώνοι" του IBM PC, κυρίως λόγω της ανοικτής αρχιτεκτονικής που επέλεξε η IBM, η Microsoft ήταν σε θέση να εφοδιάζει τους κατασκευαστές με λειτουργικό σύστημα (με ορισμένες διαφοροποιήσεις από αυτές του PC-DOS), το οποίο επονόμασε MS-DOS (ακρωνύμιο των λέξεων MicroSoftDisk Operating System).

Οι περισσότεροι υποστηρίζουν κάτι το οποίο κατά την άποψή μας φαίνεται λογικό: Η IBM επέλεξε αυτή την στρατηγική για να επεκτείνει την πελατειακή βάση μέσω από τις αντιγραφές των προϊόντων της. Με αυτόν τον τρόπο κατάφερε να αντιμετωπίσει πολλούς ανταγωνιστές της, να καθιερωθούν ουσιαστικά όλοι οι υπολογιστές της δεκαετίας του 90 ώστε να είναι συμβατή με την IBM.

Ένα ακόμα σημείο το οποίο αξίζει να αναφερθεί είναι ότι η εταιρεία του Bill Gates διατήρησε την κυριότητα, δηλαδή τα πνευματικά δικαιώματα, το νέο λογισμικό. Αν προσπαθήσουμε να κάνουμε μία αξιολόγηση αυτής της στρατηγικής θα πούμε ότι ήταν ξεκάθαρα επιτυχημένη.

Οι παλιοί γνωστές πληροφορικής επισημαίνουν ότι για την εποχή το λογισμικό της ,δεν ήταν ότι καλύτερο. Ανταγωνιστές της, είχαν να επιδείξουν καλύτερα ή μάλλον αρτιότερα λογισμικά, όμως επέλεξαν να μην δώσουν το συγκεκριμένο λογισμικό στην IBM, προφανώς γιατί θέλαμε πολλά περισσότερα ως οικονομικά ανταλλάγματα. Βλέπουμε λοιπόν ότι η ανάπτυξη της Microsoft έγινε της IBM.

Επιστρέφοντας στη σημερινή εποχή οφείλουμε να παραδεχτούμε ότι η Microsoft είναι από τις μεγαλύτερες αν όχι η μεγαλύτερη εταιρεία παραγωγής λογισμικού στον κόσμο. Τα προϊόντα της αφορούν όλο τον πληθυσμό της γης είτε είναι προγραμματιστές είτε είναι απλοί χρήστες εφαρμογών γραφείου. Τα πιο γνωστά προϊόντα της είναι το λειτουργικό σύστημα Windows, το Microsoft Office και το x-box. Επίσης δεν θα πρέπει να αγνοήσουμε τα σημαντικά προϊόντα τα οποία έχει για την ανάπτυξη λογισμικού. Η ναυαρχίδα της εταιρείας για την ανάπτυξη λογισμικού είναι το Visual Studio με τη χρήση του οποίου αναπτύξαμε τη συγκεκριμένη εφαρμογή.

6 Γλώσσες προγραμματισμού

Γλώσσα προγραμματισμού καλείται μία τεχνητή γλώσσα που μπορεί να χρησιμοποιηθεί για τον έλεγχο ενός υπολογιστή. Είναι μία συστηματική απεικόνιση μέσω της οποίας περιγράφονται υπολογιστικές διεργασίες. Οι υπολογιστικές διεργασίες δεν είναι τίποτα διαφορετικό από μία ακολουθία βημάτων, σε γλώσσα κατανοητή από τον υπολογιστή, η οποία εκτελείται για να λύσει ένα πρόβλημα.

Συνήθως μία γλώσσα προγραμματισμού χαρακτηρίζεται από ένα σύνολο συντακτικών κανόνων, μεταβλητών, εντολών και παραστάσεων που ορίζουν τη δομή και το νόημα των προτάσεων της γλώσσας. Για κάποιες γλώσσες που έχουν χρησιμοποιηθεί για αρκετό χρονικό διάστημα όπως για παράδειγμα η C, υπάρχουν ειδικοί φορείς τυποποίησης, οι οποίοι κάποιες φορές επεκτείνουν τις αρχικές τυπικές της προδιαγραφές.

Κάποιες φορές ο ορισμός γλώσσα προγραμματισμού περιορίζεται μόνο για τις αλγοριθμικές γλώσσες. Όπως θα φανεί στη συνέχεια, αυτός είναι ένας περιορισμός που δεν επικροτούμε.

Υπάρχουν πάρα πολλές διαφορετικές γλώσσες προγραμματισμού, κυρίως γιατί κάθε νέα που δημιουργείται καλύπτει μία καινούργια ανάγκη που δεν είχε προβλεφτεί στο παρελθόν.

7 Κατηγοριοποίηση γλωσσών προγραμματισμού

Στην προσπάθεια κατηγοριοποίησης των γλωσσών προγραμματισμού έχουν προταθεί διάφορα μοντέλα.

Δυστυχώς καμία προσπάθεια δεν στέφθηκε από απόλυτη επιτυχία. Ο κύριος λόγος που συνέβη αυτό είναι το γεγονός ότι οι περισσότερες γλώσσες έχουν κοινά χαρακτηριστικά ενώ παράλληλα και χρήση τους πολλές φορές γίνεται για τον ίδιο σκοπό.

Επιπρόσθετα ένα ακόμα χαρακτηριστικό που δυσκολεύει διαχωρισμό αποτελεί το γεγονός ότι κάθε γλώσσα παίρνει, η καλύτερα κληρονομεί κάποια επιτυχημένα χαρακτηριστικά από άλλες, οπότε αυτό δυσχεραίνει τον διαχωρισμό.

Τέλος, η κατηγοριοποίηση είναι ακόμα δυσκολότερη γιατί πολλές γλώσσες μπορούν να συμπεριληφθούν σε παραπάνω από μία κατηγορίες. Για αυτό το λόγο είναι γεγονός ότι πολλές γλώσσες ανήκουν σε παραπάνω από μία κατηγορίες. Για παράδειγμα, η Java είναι τόσο αντικειμενοστραφής όσο και παράλληλη γλώσσα, εφόσον υποστηρίζεται και την οργάνωση δεδομένων ως αντικείμενα, δηλαδή είναι αντικειμενοστραφής, και ταυτόχρονα μπορεί να λειτουργεί με τη δημιουργία νημάτων (threads) που εκτελούνται παράλληλα.

Λαμβάνοντας υπόψη τους παραπάνω περιορισμούς θα προσπαθήσουμε να τις κατηγοριοποιήσουμε σύμφωνα με τα παρακάτω:

7.1 Κατηγοριοποίηση σύμφωνα με τα επίπεδα που δουλεύουν στον υπολογιστή

7.1.1 Γλώσσες μηχανής

Αρχικά για να μπορέσει ο υπολογιστής να εκτελέσει μία οποιαδήποτε λειτουργία, έπρεπε να δοθούν κατευθείαν οι κατάλληλες ακολουθίες από 0 και 1, δηλαδή εντολές σε μορφή κατανοητή από τον υπολογιστή, αλλά σχετικώς μη εύκολα κατανοητές από τον άνθρωπο. Ο τρόπος αυτός ήταν επίπονος και ελάχιστοι μπορούσαν να τον υλοποιήσουν, αφού απαιτούσε βαθιά γνώση του υλικού και της αρχιτεκτονικής του υπολογιστή. Ο πρώτος υπολογιστής ο περίφημος ENIAC για να “προγραμματιστεί”, ώστε να εκτελέσει κάποιους υπολογισμούς, έπρεπε να αλλάξουν θέση εκατοντάδες διακόπτες. Ο “προγραμματισμός” των πρώτων αυτών υπολογιστών, δεν ήταν ουσιαστικά προγραμματισμός με τη σημερινή έννοια του όρου, αλλά είναι η τροποποίηση των κυκλωμάτων μίας μηχανής που εκτελεί υπολογιστικές διεργασίες.

7.1.2 Συμβολικές γλώσσες ή γλώσσες χαμηλού επιπέδου

Στην συνέχεια έγιναν προσπάθειες για τη δημιουργία μίας συμβολικής γλώσσας, η οποία θα έχει και κάποια έννοια για τον άνθρωπο με τη χρήση απλών λέξεων. Αυτές οι «λέξεις» θα μετατρέπονται στους υπολογιστές σε αντίστοιχες ακολουθίες από 0 και 1. Ουσιαστικά αυτές οι γλώσσες προγραμματίζουν ψηφιακά κυκλώματα τα οποία με τη σειρά τους είναι η απεικόνιση λογικών κυκλωμάτων. Η μετατροπή σε εκτελέσιμο πρόγραμμα γίνεται με τη χρήση **συμβολομεταφραστή (assembler)**.

Το μεγαλύτερο μειονέκτημα αυτής της προσέγγισης αποτελεί το γεγονός πως ήμασταν πάρα πολύ δεμένοι με την αρχιτεκτονική του κάθε υπολογιστή, οπότε η μετατρεψιμότητα των προγραμμάτων ήταν πάρα πολύ δύσκολη ως αδύνατη σε κάποιες περιπτώσεις.

7.1.3 Γλώσσες υψηλού επιπέδου

Τα παραπάνω προβλήματα οδήγησαν στην ανάπτυξη των πρώτων γλωσσών Προγραμματισμού υψηλού επιπέδου. Οι γλώσσες FORTRAN, COBOL, C αποτελούν παραδείγματα γλωσσών προγραμματισμού υψηλού επιπέδου. Ποια χαρακτηριστικά τους όμως τις κατατάσσουν ως γλώσσες «υψηλού επιπέδου»;

- a. Η ανεξαρτησία από το υλικό του υπολογιστή. Δηλαδή επιτυγχάνεται η τόσο σημαντική **μετατρεψιμότητα** των δημιουργημένων προγραμμάτων
- b. Η σχετικά εύκολη διόρθωση και η συντήρηση τους
- c. Η ευκολία στην εκμάθησή τους, δεδομένου ότι ακολουθούν κανόνες οι οποίοι είναι κοντά στην ανθρώπινη αλγοριθμική σκέψη

7.1.4 Γλώσσες 4ης γενιάς

Παρότι οι γλώσσες υψηλού επιπέδου έγιναν κοινά αποδεκτές λόγω των πλεονεκτημάτων που συγκέντρωναν, είχαν ένα σημαντικό μειονέκτημα: απευθύνονταν μόνο σε προγραμματιστές.

Επιπρόσθετα, για να υλοποιήσει κάποιος ένα αρκετά σύνθετο πρόγραμμα απαιτείται πολύς χρόνος, μέρος του οποίου αναλώνεται στις ιδιαιτερότητες της γλώσσας προγραμματισμού που χρησιμοποιεί.

Για να ξεπεραστούν αυτά τα προβλήματα, αλλά και να δοθούν στους χρήστες ενός υπολογιστή κάποιες προγραμματιστικές δυνατότητες, κυρίως μέσω της εκτέλεσης κάποιων ερωτημάτων (queries) δημιουργήθηκαν γλώσσες που εφοδιάστηκαν με εργαλεία προγραμματισμού που αποκρύπτουν πολλές λεπτομέρειες από τις τεχνικές υλοποίησης.

Οι γλώσσες αυτές καλούνται 4ης γενιάς με τις περισσότερο χαρακτηριστικές να είναι η **SQL** και η **ABAP** (χρησιμοποιείται στο SAP).

7.2 Κατηγοριοποίηση σύμφωνα το τρόπο οργάνωση προγράμματος

Με βάση την οργάνωση του προγράμματος έχουμε τις εξής κατηγορίες

- **Διαδικαστικές γλώσσες (procedural)**. Μία μεγάλη κατηγορία τέτοιων γλωσσών είναι οι βασιζόμενες σε διαδικασίες, οι οποίες στη συνέχεια καλούνται σε άλλα μέλη του προγράμματος. Χαρακτηριστικό παράδειγμα αποτελεί η Pascal και η C.
- **αντικειμενοστραφείς γλώσσες**: σύμφωνα με τις οποίες όλα τα πρόγραμμα οργανώνονται σε αντικείμενα. Η περιγραφή των αλγορίθμων αφορά συσχετίσεις στα συγκεκριμένα αντικείμενα, ενώ όλοι οι αλγόριθμοι είναι αλληλεπιδράσεις σε αντικείμενα. Παραδείγματα αντικειμενοστραφών γλωσσών είναι η Java ή η C++.
- μία άλλη σημαντική κατηγορία είναι **συναρτησιακές** γλώσσες προγραμματισμού συνήθως αφορούν προγραμματισμό σημαντικών μαθηματικών εφαρμογών. Μία τέτοια χαρακτηριστική γλώσσα είναι Lisp

7.3 Κατηγοριοποίηση με βάση τον στόχο που έχει η γλώσσα

Η κατηγοριοποίηση των γλωσσών προγραμματισμού μπορεί να γίνει με βάση το στόχο που έχει η γλώσσα, οπότε έχουμε:

- **Γλώσσες γενικής χρήσης**. Σε αυτήν την κατηγορία λαμβάνονται γλώσσες που δημιουργήθηκαν κυρίως για γενικές εφαρμογές ενώ επίσης περιλαμβάνονται χθες που έχουν αναπτυχθεί για ανάπτυξη και εκπαίδευση νέων προγραμματιστών όπως η Pascal.
- **Γλώσσες προγραμματισμού συστημάτων**, οι οποίες κυρίως χρησιμοποιούνται για την για τον προγραμματισμό λειτουργικών συστημάτων και την υποστήριξη drivers. Ο κυρίαρχος σε αυτή την κατηγορία είναι η γλώσσα προγραμματισμού C.
- **Γλώσσες σεναρίων (scripting)**, οι γλώσσες αυτές βρίσκουν εκτεταμένη χρήση στην ανάπτυξη μικρών προγραμμάτων κατά τα οποία ο χρόνος του προγραμματιστή είναι πιο πολύτιμος την ταχύτητα της εκτέλεσης προγράμματος. Τέτοιες γλώσσες σεναρίων είναι python ή perl κάποια και κελυφή του λειτουργικού συστήματος unix (δηλαδή shells)

- **Γλώσσες ειδικών εφαρμογών,** στην κατηγορία αυτή ανήκουν οι γλώσσες οι οποίες έχουν αναπτυχθεί για μία συγκεκριμένη λειτουργία ή μία συγκεκριμένη εφαρμογή. Για παράδειγμα το matlab είναι σχεδιασμένο για επεξεργασία πινάκων και σημαντικές μηχανολογικές εφαρμογές , ενώ το mathematica όπως λέει και το όνομα του είναι αναπτυγμένο για εφαρμογές μαθηματικών.

- **Παράλληλες ή κατανεμημένες γλώσσες,** εδώ έχουμε κατανεμημένες τις γλώσσες τις οποίες υποστηρίζουν την παράλληλη επεξεργασία προγραμμάτων. Ο βασικός στόχος είναι να τρέξει το τελικό εκτελέσιμο πρόγραμμα γρηγορότερα. Η επεξεργασία γίνεται παράλληλα μέσω νημάτων. Παραδείγματα γλωσσών που ανήκουν (και) σε αυτή την κατηγορία είναι η Go, η Java, η Erlang, η MultiLisp.

7.4 Κατηγοριοποίηση με βάση τον τρόπο που περιγράφουν το ζητούμενο αποτέλεσμα

- **Προστακτικές γλώσσες προγραμματισμού (imperative)**

οι προστακτικές γλώσσες προγραμματισμού, εκτελούν το πρόγραμμα τους μέσω κάποιων εντολών. Οι εντολές- προσταγές δίνουν το ζητούμενο αποτέλεσμα. Είναι οι γλώσσες που περιγράφουν το ζητούμενο αποτέλεσμα κατασκευαστικά, δίνοντας μια σειρά εντολών που όταν εκτελεστούν παράγουν το ζητούμενο αποτέλεσμα. Τέτοιες γλώσσες είναι η C, η Java αλλά και η OCaml.

- **Δηλωτικές γλώσσες προγραμματισμού (declarative)**

είναι γλώσσες οι οποίες περιγράφουν το ζητούμενο αποτέλεσμα χωρίς να προσεγγίζουν απόλυτα τον τρόπο με τον οποίο παράγονται. Παραδείγματα γνωστόν αποτελεί SQL και η Prolog.

8 Ιστορία γλωσσών προγραμματισμού

Ακολουθεί μία ιστορική αναδρομή της δημιουργίας των γλωσσών προγραμματισμού η οποία θα μας βοηθήσει να αξιολογήσουμε τα προγραμματιστικά εργαλεία που χρησιμοποιήσαμε.

8.1 FORTRAN

Το 1957 η IBM ήταν πρώτη οποία ανέπτυξε γλώσσα υψηλού επιπέδου τη FORTRAN. Το όνομα FORTRAN προέρχεται από τις λέξεις FORmula TRANsalte. Η FORTRAN αναπτύχθηκε αρχικά ως γλώσσα κατάλληλη για την επίλυση δύσκολων μαθηματικών και επιστημονικών προβλημάτων.

Όταν ένα πρόγραμμα έχει γραφτεί στη γλώσσα FORTRAN αλλά και σε οποιαδήποτε άλλη γλώσσα υψηλού επιπέδου άμεσα μεταφράζεται σε γλώσσα μηχανής του υπολογιστή με τη βοήθεια ενός ειδικού προγράμματος το οποίο καλείται **μεταγλωττιστής**. Το σημαντικό είναι ότι η συγκεκριμένη υλοποίηση, δηλαδή το συγκεκριμένο πρόγραμμα μπορεί να μεταφερθεί και να τρέχει σε έναν οποιοδήποτε άλλο υπολογιστή, αρκεί να υπάρχει ο αντίστοιχος μεταγλωττιστής

Η FORTRAN παρά τα πολλά τα πολύ δυνατά χαρακτηριστικά που είχε ενσωματώσει αλλά ταυτόχρονα και τις συχνές αλλαγές οι οποίες γινόντουσαν για να ανταποκριθεί στις νέες απαιτήσεις, δεν μπόρεσε να καλύψει όλο το εύρος των αναγκών του προγραμματισμού. Κάποιοι ισχυρίζονταν, και πιθανότατα είχαν δίκιο, ότι υστερούσε σημαντικά σε εμπορικές εφαρμογές. Το κενό αυτό έμελε να το καλύψει η ακριβώς για επόμενη γλώσσα που εξετάζουμε, η COBOL.

Η γλώσσα FORTRAN πέρασε στην ιστορία ως η πρώτη γλώσσα προγραμματισμού υψηλού επιπέδου

8.2 COBOL

στα τέλη της δεκαετίας του 50 αναπτύχθηκε μία άλλη γλώσσα η οποία απέκτησε ευρύτατη αποδοχή σε εμπορικές εφαρμογές: ήταν η περίφημη COBOL. Η COBOL όπως δηλώνει και το όνομα της (COmmon Business Oriented Language -Κοινή γλώσσα προσανατολισμένη στις επιχειρήσεις) ήταν κατάλληλα σχεδιασμένη για την ανάπτυξη εμπορικών εφαρμογών, κυρίως λόγω του γεγονότος ότι χειριζόταν με μεγάλη ευκολία τα αρχεία.

Η COBOL καθιερώθηκε ως η κυρίαρχη προγραμματιστική επιλογή για επιχειρήσεις και εκατομμύρια εμπορικές εφαρμογές αναπτύχθηκαν σε αυτή, τόσο στις επιχειρήσεις όσο και στο δημόσιο τομέα. Πιθανότατα, ακόμα και σήμερα υπάρχουν εμπορικές εφαρμογές οι οποίες συντηρούνται.

Η COBOL έκανε δυνατή τη ουσιαστική αξιοποίηση των υπολογιστών από τις επιχειρήσεις και τους οργανισμούς παρέχοντας-ισχυρότατες δυνατότητες διαχείρισης αρχείων δεδομένων.

Δύο σημαντικότερες γλώσσες γενικού σκοπού, οι οποίες αναπτύχθηκαν τη δεκαετία του 60 αλλά χρησιμοποιούνται πάρα πολύ στις ημέρες μας, είναι η BASIC και η PASCAL.

8.3 BASIC

Η γλώσσα BASIC δημιουργήθηκε το 1964 στο Dartmouth College από τους καθηγητές Kemeny και Kurtz. Ο βασικός σκοπός των δημιουργών της ήταν η υλοποίηση μιας απλής γλώσσας προγραμματισμού κυρίως για εκπαιδευτικό σκοπό. Παρόλα αυτά η ιστορία έμελλε να ήταν διαφορετική. Η πάρα πολύ μεγάλη διάδοση της έγινε όταν ο Bill Gates και οι συνεργάτες του υλοποίησαν ένα διερμηνευτή για το IBM PC.

Η BASIC λοιπόν, γνώρισε εκπληκτική διάδοση με την εμφάνιση των μικροϋπολογιστών (1975). Ο πρώτος διερμηνευτής της γλώσσας σε 8K ROM ήταν έργο των Bill Gates και Paul Allen. Ας σημειωθεί ότι, η εταιρία Microsoft στήριξε την ανάπτυξή της για την εμπορική εκμετάλλευση αυτού του διερμηνευτή.

Η έκδοση GWBASIC παρουσιάζεται το 1981 με τον IBM PC. Εξέλιξη της γλώσσας είναι οι εκδόσεις QuickBASIC και Visual BASIC.

Η ανάπτυξη όμως των μικροϋπολογιστών αλλά και οι συνεχείς εκδόσεις της γλώσσας βοήθησαν στην εξάπλωσή της, τόσο ώστε να γίνει ίσως η δημοφιλέστερη γλώσσα στους προσωπικούς υπολογιστές.

Η τυποποίηση της δε από τη Microsoft, με τις εκδόσεις QuickBasic και κύρια με τη Visual Basic, καθιέρωσε τη γλώσσα ως πρότυπο για ανάπτυξη εφαρμογών σε προσωπικούς υπολογιστές.

8.4 PASCAL

Η γλώσσα PASCAL (δημιούργημα του καθηγητή Niklaus Wirth) έφερε επανάσταση στις τεχνικές προγραμματισμού. Αναπτύχθηκε σαν μια γλώσσα γενικής χρήσης, ενώ έδωσε την ώθηση για **δομημένο προγραμματισμό**. Ο δομημένος προγραμματισμός είναι μία προγραμματιστική τεχνική κατά την οποία χρησιμοποιούνται μόνο τρεις δομές για την ανάπτυξη των προγραμμάτων: η ακολουθιακή δομή, η δομή επιλογής και η δομή επανάληψης.

Η PASCAL έχει γνωρίσει αρκετά μεγάλη εξάπλωση και θεωρείται η καλύτερη γλώσσα για να διδαχθεί κάποιος προγραμματισμό. Στα σύγχρονα προγραμματιστικά περιβάλλοντα μπορεί να τη συναντήσει κάτω από την πλατφόρμα της Delfi, και χρησιμοποιείται ακόμα και σήμερα σχεδόν για όλων των ειδών τις εφαρμογές.

8.5 C

Μία ακόμα εξαιρετικά σημαντική γλώσσα για τον προγραμματισμό αποτελεί η γλώσσα C. Η C αναπτύχθηκε στα εργαστήρια της εταιρείας BELL και χρησιμοποιήθηκε για την ανάπτυξη του λειτουργικού συστήματος Unix.

Έχοντας πολλά κοινά χαρακτηριστικά με την PASCAL, κυρίως αυτά που αφορούν την αναγκαιότητα ύπαρξης δομημένου προγραμματισμού, είχε συνάμα ενσωματώσει χαρακτηριστικά τα οποία την κάνουν ξεχωριστή.

Το σημαντικότερο από αυτά είναι η ύπαρξη pointers, μέσω των οποίων μπορεί να στοχεύει και να επεξεργάζεται θέσεις μνήμης στον υπολογιστή στον οποίο τρέχει η εφαρμογή που έχει αναπτυχθεί. Επίσης και η λιτότητα στην σύνταξη προγραμμάτων, η οποία την κάνει εξαιρετικά γρήγορη είναι ένα ακόμα καλό χαρακτηριστικό.

Όπως ειπώθηκε και στην αρχή, η C είναι άρρηκτα συνδεδεμένη με το λειτουργικό σύστημα unix και τον διάδοχό του το linux. Παρόλα αυτά συμπεριφέρεται εξίσου καλά και σε άλλες πλατφόρμες. Δεν είναι τυχαίο, ότι το λειτουργικό σύστημα windows έχει χτιστεί με την γλώσσα προγραμματισμού C, ενώ και πολλές γλώσσες scripting, δηλαδή τέταρτης γενιάς, όπως για παράδειγμα ο SQL τρέχουν βασισμένοι στην C ώστε να εκτελούνται πιο γρήγορα τα ερωτήματα τα οποία θέλει ο χρήστης.

Η C επίσης εξελίχτηκε στη γλώσσα C++, που είναι αντικειμενοστραφής.

8.6 JAVA

Με την ανάπτυξη του internet δημιουργήθηκε η αναγκαιότητα να υπάρχει μία γλώσσα η οποία να χειρίζεται καλά τα διαφορετικά γραφικά περιβάλλοντα που υπήρχαν σε κάθε υπολογιστή. Την αναγκαιότητα αυτή κατάφερε να την ικανοποιήσει τη γλώσσα προγραμματισμού java η οποία μέσω των java applets κατάφερε να είναι ανεξάρτητη από το υλικό κάθε υπολογιστή και ταυτόχρονα να εκτελεί πλήρως καθορισμένα βήματα.

Όπως μπορεί πολύ εύκολα κάποιος να υποθέσει, η γλώσσα προγραμματισμού java είναι πάρα πολύ σημαντική ανάπτυξη εφαρμογών στο διαδίκτυο. Πολλοί προγραμματιστές ξεκινούν τη καριέρα τους στην ανάπτυξη λογισμικού από αυτή.

Αν θέλαμε κάνουμε μία σύγκριση με άλλες γλώσσες που έχουμε ήδη περιγράψει, java είναι πολύ πιο κοντινή C++ η οποία είναι και αντικειμενοστραφής, και ταυτόχρονα έχει λίγο-πολύ την ίδια σύνταξη.

Το κύριο χαρακτηριστικό που την διαφοροποιεί είναι τα java applets.

9 Δομημένος Προγραμματισμός

Όπως αναφέρθηκε προηγουμένως, ο δομημένος προγραμματισμός είναι προγραμματιστική τεχνική κατά την οποία γίνονται αποδεκτές μόνο τρεις δομές: η δομή ακολουθίας, η δομή επιλογής και η δομή επανάληψης.

Καμία άλλη δομή ή καμία άλλη εντολή όπως για παράδειγμα το goto δεν γίνεται αποδεκτή. Αν θέλουμε πάμε στα πλαίσια του δομημένου προγραμματισμού, ο δομημένος προγραμματισμός εγγυάται ότι το πρόγραμμά μας θα είναι κατάλληλα διαμορφωμένο και εύκολα συντηρήσιμο. Έχει αποδειχθεί ότι δεν χρειάζεται οποιαδήποτε άλλη δομή για να υλοποιήσουμε οποιοδήποτε πρόγραμμα.

Μία κριτική που θα μπορούσαμε να ασκήσουμε στην προγραμματιστική τεχνική με δομημένο προγραμματισμό, αποτελεί η ίδια η αυστηρή δομή του, η οποία απαγορεύει δομές ελέγχου οι οποίες λειτουργούν ως έξοδοι από το πρόγραμμα. Τέτοιες προγραμματιστικές τεχνικές μπορούν να έχουν άμεση χρησιμότητα στα περιβάλλοντα τα οποία τα οποία είναι απαραίτητο να κλείνουν σε κάποιες περιπτώσεις ή να μεταβαίνουν με μη δομημένο τρόπο σε άλλο σημείο του κώδικα. Ένα τέτοιο παράδειγμα είναι το άνοιγμα ή το κλείσιμο ενός διακόπτη, όπως μπορεί αυτό να συμβαίνει σε sockets.

10 Προγραμματισμός οδηγούμενου από το γεγονός προγραμματισμού (object driven programming) και του οπτικός προγραμματισμός (visual programming).

Μία ρηξικέλευθη προγραμματιστική τεχνική αποτελεί ο προγραμματισμός με βάση τα γεγονότα. Σύμφωνα με αυτή, ο οδηγός για τη σύνταξη προγράμματος αποτελούν κάποια συγκεκριμένα γεγονότα (events), τα οποία μας καθοδηγούν το πώς θα συντάξουμε τη δομή του προγράμματος. Παρότι αυτή η τεχνική δεν έμοιαζε με τις προϋπάρχουσες, κέρδισε φανατικούς υποστηρικτές δεδομένου ότι μπορούσε να περιγράψει με ευκρίνεια τη ροή της πληροφορίας μέσα στο πρόγραμμα.

Ο προγραμματισμός με βάση τα γεγονότα είναι άρρηκτα συνυφασμένος με τον οπτικό προγραμματισμό (visual programming). Τα περισσότερα σύγχρονα προγραμματιστικά περιβάλλοντα έχουνε μετατραπεί σε οπτικοποιημένα. Η παράσταση του προγραμματισμού και ο κώδικας, ακόμα και όταν και όταν πρέπει να γραφτεί στα

πρότυπα του δομημένου προγραμματισμού, αυτό γίνεται κάτω από ένα συγκεκριμένο κουτάκι ή σχήμα. Σε αυτό το σχήμα γράφουμε τον κώδικα τον οποίο θα εκτελέσουμε. Φυσικά ένα από τα πιο διαδεδομένα γραφικά περιβάλλοντα οπτικού προγραμματισμού αποτελεί το Visual Studio, ενώ ακολουθούν το Delfi (Pascal) και το Eclipse (Android Applications και Java)

11 Αντικειμενοστραφής Προγραμματισμός

Αντικειμενοστραφής προγραμματισμός με μία σχεδιαστική τεχνική ανάπτυξη αλγορίθμων η οποία έχει αποκτήσει σημαντική αποδοχή. Η λογική του αντικειμενοστραφούς προγραμματισμού βασίζεται στη σχεδίαση με βάση τα αντικείμενα. Κάθε νοητή οντότητα εμπεριέχεται σε ένα διακριτό σχήμα και όλες οι δομές αλλά και ο κώδικας που περικλείεται σε αυτή υπάρχει μέσα στο αντικείμενο. Με την ίδια μεθοδολογία αναπτύσσονται και οι σχέσεις μεταξύ των δεδομένων και της πληροφορίας που υπάρχουν στο σχήμα μας.

Οι πιο διαδεδομένες γλώσσες αντικειμενοστραφούς προγραμματισμού είναι C++ και JAVA. Η τελευταία μάλιστα, μετά από την εξαιρετικά εκτεταμένη εφαρμογή της στο ιντερνέτ έχει γίνει πιθανότατα η νούμερο ένα γλώσσα στην ανάπτυξη λογισμικού αυτή τη στιγμή.

Πιο αναλυτικά, με τη μέθοδο του αντικειμενοστραφούς προγραμματισμού δεν υπάρχει διάκριση μεταξύ δεδομένων και διαδικασιών. Και τα δύο μαζί αποτελούν μία αυτόνομη οντότητα, η οποία συνδέεται με τη σειρά της με άλλες αυτόνομες οντότητες που και αυτές είναι σειρά τους δεδομένα και διαδικασίες.

Μία σημαντική καινοτομία του αντικειμενοστραφούς προγραμματισμού αποτέλεσε η δημιουργία κλάσεων αντικειμένων. Κάθε κλάση ορίζει συγκεκριμένα χαρακτηριστικά για τα δεδομένα που περιλαμβάνει ταυτόχρονα. Τα αντικείμενα καταλαμβάνουν χώρο στη μνήμη του υπολογιστή ενώ οι κλάσεις προδιαγράφουν πως οι διαδικασίες θα επιδράσουν πάνω σε αυτά.

12 Σύγκριση μεταξύ δομημένου και αντικειμενοστραφούς προγραμματισμού

Στην ανωτέρω ανάλυση διαπιστώνουμε πως οι αιτίες που ώθησαν στην ανάπτυξη του δομημένου προγραμματισμού οδήγησαν και στην ανάπτυξη του αντικειμενοστραφούς. Και με τις δύο προγραμματιστικές τεχνικές έχουμε επιτύχει ευκολία στη συντήρηση, δομημένο τρόπο προγραμματισμού και εύκολη συντήρηση κώδικα μεγάλων εφαρμογών. Παρ' όλες τις λογικές ομοιότητες η ευκολία του αντικειμενοστραφούς προγραμματισμού αλλά και της δυναμικής που αυτός είχε στο χειρισμό μεγάλων όγκων δεδομένων μας, οδήγησε το να έχουμε στην σχεδόν ολοκληρωτική επικράτηση του αντικειμενοστραφούς προγραμματισμού.

13 Ανάλυση Αντικειμενοστραφούς Προγραμματισμού

Η βασική σύλληψη στον αντικειμενοστραφή είναι η κλάση (class), η οποία είναι μία αυτοτελής αφαιρετική αναπαράσταση κατηγοριών αντικειμένων. Θα μπορούσαμε να πούμε ότι πρακτικά ένας τύπος δεδομένων, αλλά με το γενικότερο της ορισμό μπορούμε να περιγράψουμε αφαιρετικά μία άλλη μορφή όπως για παράδειγμα έναν αλγόριθμο ή μία πηγή δεδομένων.

Μία κλάση μπορεί να είναι είτε δημόσια, δηλαδή public, είτε ιδιωτική δηλαδή private. Οι δημόσιες κλάσεις μπορούν να τις καλέσουν από οποιοδήποτε σημείο της εφαρμογής, ενώ ιδιωτικές αναφέρονται μόνο στο τμήμα της εφαρμογής στο οποίο ορίζονται. Οι διαδικασίες των εφαρμογών εντός των κλάσεων λέγονται μέθοδοι (methods), ενώ τα χαρακτηριστικά λέγονται τους attributes.

Μία κλάση πρέπει ιδανικά να είναι εννοιολογικά αυτόνομη, να περιέχει δηλαδή μόνο πεδία τα οποία περιγράφουν μία κατηγορία αντικειμένων και διεργασίες που τρέχουν, όταν αυτές καλούνται από το εξωτερικό πρόγραμμα, χωρίς να εξαρτώνται από άλλα δεδομένα ή κώδικα εκτός της κλάσης. Είναι δυνατόν να επαναχρησιμοποιηθεί ως τμήμα διαφορετικών προγραμμάτων.

Αντικείμενο μιας κλάσης είναι ένα στιγμιότυπο αυτής, το οποίο μπορεί να καταγραφεί σε οποιαδήποτε χρονική στιγμή, και ταυτόχρονα περιέχει χαρακτηριστικά τα οποία έχουν περιγραφεί μέσω της κλάσης.

14 Χαρακτηριστικές ιδιότητες αντικειμενοστραφούς προγραμματισμού

Μία πολύ σημαντική τεχνική αντικειμενοστραφούς προγραμματισμού αποτελεί η **ενθυλάκωση δεδομένων (data encapsulation)**. Μέσω αυτής, οι κλάσεις καταφέρνουν να κρατήσουν κρυφά τα ιδιωτικά δεδομένα από το υπόλοιπο πρόγραμμα, και ταυτόχρονα μέσω των μεθόδων οι οποίες γίνονται public να μπορούν να διαδίδουν τα χαρακτηριστικά τους σε όλο το πρόγραμμα.

Αφαιρετικότητα λέγεται η αφαίρεση δεδομένων η οποία χαρακτηρίζει την προγραμματιστική τεχνική κατά την οποία η κλάση αποτελεί το μοντέλο- πρότυπο των δεδομένων. Κάποιες κατηγορίες αντικειμένων μέσω αυτής της τεχνικής δεν είναι απαραίτητο να τις σχεδιάσουμε και τελικώς να έχουμε για κάθε νέα οντότητα καινούργια δομή. Χρησιμοποιούμε την προηγούμενη δομή, όπως αυτή έχει περιγραφεί επακριβώς στη κλάση.

Μία ακόμα σημαντική προγραμματιστική τεχνική υποδομή που φαίνεται μέσα από τον αντικειμενοστραφή προγραμματισμό αποτελεί η **κληρονομικότητα των κλάσεων**. Μέσω της κληρονομικότητας μία κλάση μπορεί να κληροδοτήσει τα χαρακτηριστικά της σε μία άλλη κλάση. Η νέα κλάση μπορεί να έχει επιπρόσθετα και τα δικά της χαρακτηριστικά. Αυτό το οποίο δεν συμβαίνει είναι ότι η μεταγενέστερη κλάση που λέγεται υπόκλαση, δηλαδή ο απόγονος της αρχικής κλάσης, δεν κληροδοτεί τα χαρακτηριστικά της αρχική κλάση (στον γονέα) αλλά μπορεί να έχει και αυτόνομα χαρακτηριστικά.

15 Εργαλεία δημιουργίας εφαρμογών στο λειτουργικό σύστημα Microsoft Windows

Η Microsoft ξεκίνησε την ανάπτυξη στο .NET Framework στα τέλη της δεκαετίας του 1990, όμως στην αγορά τα πρώτα προϊόντα δόθηκαν στις αρχές της δεκαετίας του 2000.

Αν και θεωρητικά μπορεί να εγκατασταθεί σε οποιοδήποτε προσωπικό υπολογιστή, το οποίο έχει Windows, δεν μπορούμε να εξαιρέσουμε κάποιες ασυμβατότητες που είναι πιθανό να προκύψουν, κυρίως λόγω της έκδοσης τους.

Η έκδοση 3.0 του .NET Framework συνήθως ταιριάζει με τα windows server 2008 και τα Windows Vista, ενώ η έκδοση του Framework 3.5 ήτανε φτιαγμένο για τα Windows 7. Στη συνέχεια οι περισσότερες εκδόσεις του framework είναι φτιαγμένες για το visual studio. Από το visual studio 2010 και στη συνέχεια υποστήριξε ταυτόχρονα και υλοποιήσεις για κινητές συσκευές. Επίσης είχαμε ταυτόχρονα υλοποιήσεις και για ενσωμάτωση υπολογιστικά συστήματα όπως για παράδειγμα το raspberry pi.

Φυσικά όπως θα περιμέναμε, η έκδοση .NET framework για τις εκδόσεις για κινητά αλλά και για φορητές συσκευές τα οποία έχουν υπολογιστικά συστήματα, είναι πιο ελαφριά με μόνο τα χαρακτηριστικά που είναι απαραίτητα να είναι ενσωματωμένα.

15.1 Τι είναι το .NET Framework?

Το NET Framework είναι ένα λογισμικό το οποίο εγκαθίσταται σε ηλεκτρονικούς υπολογιστές οι οποίοι έχουν το λειτουργικό σύστημα windows. Το λογισμικό αυτό είναι απαραίτητο για να τρέξουν πάρα πολλά προγράμματα οποία βασίζονται στη

συγκεκριμένη πλατφόρμα, ενώ το συγκριτικό πλεονέκτημα με άλλες υλοποιήσεις είναι ότι η πλατφόρμα ταιριάζει με πάρα πολλές γλώσσες προγραμματισμού.

Στη συνέχεια θα περιγράψουμε λίγο πιο τεχνικά πως έχει υλοποιηθεί το συγκεκριμένο framework.

15.2 Αρχιτεκτονική

Η βασική βιβλιοθήκη framework παρέχει όλα τα στοιχεία τα οποία βοηθάνε στον προγραμματισμό, όπως για παράδειγμα πρόσβαση σε δεδομένα, διεπαφή χρήστη και εύκολη διασύνδεση με πολλές βάσεις δεδομένων. Η κρυπτογραφία αλλά και επικοινωνία μέσω δικτύου είναι εξασφαλισμένη. Βλέπουμε λοιπόν, ότι οι προγραμματιστές χρησιμοποιούν το δικό τους κώδικα, που σε συνδυασμό .NET framework κάνει την εφαρμογή τους μοναδική και με κατάλληλη ασφάλεια.

Επίσης, όλα τα προγράμματα τα οποία έχουν δημιουργηθεί με το .NET framework μπορούν να εκτελεστούν σε οποιοδήποτε περιβάλλον υπολογιστή υπό την προϋπόθεση ότι η πλατφόρμα περιλαμβάνει CommonLanguageRuntime (CLR).

Το CLR είναι ουσιαστικά μία εικονική μηχανή έτσι ώστε οι προγραμματιστές να μην έχουν την ανάγκη να υπολογίζουν το υλικό στο οποίο ανεχτεί έχει αναπτύξει το πρόγραμμά τους ενώ ταυτόχρονα ενσωματωμένα άλλα σημαντικά χαρακτηριστικά όπως είναι η ασφάλεια η διαχείριση της μνήμης αλλά και ο χειρισμός των λαθών

Θα διαπιστώσουμε πως υπάρχει και μία μεγάλη βιβλιοθήκη FCL η οποία παρέχει διαλειτουργικότητα για κάθε δυνατή γλώσσα προγραμματισμού. Αν αλλάξουμε το υλικό του υπολογιστή στο οποίο τρέχουμε το ανάλογο πρόγραμμα, τότε αναλαμβάνει η Common Language Runtime (CLR), μια εφαρμογή εικονικής μηχανής που παρέχει διάφορες υπηρεσίες, όπως η ασφάλεια, η διαχείριση μνήμης και ο χειρισμός εξαιρέσεων. FCL και CLR μαζί αποτελούν το .NET Framework μέσω των οποίων κάνουν όλες τις εφαρμογές τους συμβατές.

16 Microsoft Visual Basic

Παρακάτω θα χρησιμοποιηθεί η γλώσσα προγραμματισμού visual basic. Διαπιστώσαμε ότι με σχετικά εύκολο τρόπο γίνεται η χρησιμοποίηση των βιβλιοθηκών του framework, ενώ ο χρήστης ωφελείται από την χρήση αυτής για την ανάπτυξη καθώς αποκτά πρόσβαση σε βιβλιοθήκες του .Net Framework.

17 Ολοκληρωμένο Περιβάλλον Εργασίας (Integrated Development Environment)

Το περιβάλλον εργασίας στο Microsoft Visual Studio 2015 είναι συνέχεια των προηγούμενων εκδόσεων του visual studio, με ακόμα καλύτερη λειτουργικότητα. Επιτρέπει την ευκολότερη χρήση εργαλείων, ενώ και η αποσφαλμάτωση του κώδικα γίνεται με πολύ μεγαλύτερη ευκολία. Επίσης μπορεί να επιτευχθεί η εμφάνιση ανάλογα με τις επιθυμίες του χρήστη, ενώ ταυτόχρονα το πρόγραμμα θυμάται τις προηγούμενες επιλογές. Αυτό σημαίνει ότι η λύση για ανάπτυξη μέσα από το Visual Studio είναι εξαιρετικά εύκολη, με την εφαρμογή να ξεκινά τη συγκεκριμένη γλώσσα στην οποία έχει συνηθίσει ο χρήστης για να αναπτύσσει τις εφαρμογές του.

Ένα άλλο στοιχείο το οποίο αξίζει να αναφερθεί αποτελεί το γεγονός ότι μέσω του Visual Studio πλέον μπορείς να αναπτύξεις εφαρμογές σε linux αλλά και σε android.

18 Λύσεις και Έργα (Solutions and Projects)

Το Microsoft Visual Studio 2015 κατηγοριοποιεί τα έργα ως λύσεις και έργα. Αυτό σημαίνει ότι κάτω από το φάκελο project, υπάρχουν όλα τα έργα τα οποία ανέπτυξε, ενώ κάτω από το φάκελο λύσεις υπάρχουν όλα τα τελευταία αρχεία, τα οποία είναι ουσιαστικά προγράμματα τα οποία έχουν ολοκληρωθεί και αποτελούν ολοκληρωμένες λύσεις (solutions).

19 Εργαλεία Δοκιμών (Testing Tools)

Στο Microsoft Visual Studio παρέχεται επίσης τη λειτουργικότητα του ελέγχου των λαθών έτσι ώστε οι προγραμματιστές να είναι σε θέση να αναπαράγουν τα τυχόν σφάλματα και ταυτόχρονα να ελέγχουν τη σταθερότητα τις αναπτυσσόμενες εφαρμογές.

Με κατάλληλη μορφή, εμφανίζονται μηνύματα στο χρήστη τα οποία μπορούν να καθοδηγήσουν σε εύρεση λύσης του προβλήματος, ενώ με τον αντίστοιχο κωδικό του λάθους, υπάρχει μεγάλο πλήθος πληροφορίας στο ιντερνέτ. Κατά συνέπεια, γίνεται αποσφαλμάτωση με εύκολο τρόπο.

20 Σχεδιασμός και Μοντελοποίηση (Design and Modeling)

Το Visual Studio στο οποίο παρέχεται λειτουργικότητα για σχεδιασμό και ανάπτυξη. Επίσης παρέχεται καλύτερη διαχείριση από το χρήστη για εφαρμογές που είναι οπτικά συνδεδεμένες οπότε ελαχιστοποιείται η πιθανότητα σφάλματος.

21 Microsoft Visual Studio SDK (Software Development Kit)

Το Microsoft Visual Studio SDK (Software Development Kit) είναι μια επέκταση του Microsoft Visual Studio 2015 το οποίο περιλαμβάνει βιβλιοθήκες και εργαλεία τα οποία βοηθούν το χρήστες για να αναπτύξουν εφαρμογές κατάλληλες όχι μόνο για windows, αλλά ταυτόχρονα δοκιμάζεται και η εφαρμογή για android αλλά ακόμα και για windows phone.

Για το τελευταίο υπάρχει και ένας emulator, ο οποίος λειτουργεί ως και η απεικόνιση του windows phone. Με αυτό ο προγραμματιστής, χωρίς να έχει στην κατοχή του τηλέφωνο, μπορεί να ξέρει πώς θα φαίνεται η εφαρμογή του όπως σε ένα windows phone.

Μία καινούργια λειτουργικότητα που έχει το visual studio 2015 είναι η δυνατότητα ανάπτυξης εφαρμογής universal η οποία λειτουργεί και στα windows 10 αλλά και στα κινητά και τα tablet τα οποία βασίζονται σε windows 10

Δυστυχώς στην περίπτωση έχουμε ένα σημαντικό περιορισμό ως προς το υλικό του υπολογιστή στο οποίο γίνεται ανάπτυξη. Με την ακόλουθη παραμετροποίηση να είναι η μοναδική η οποία υποστηρίζει ανάπτυξη universal εφαρμογών. Χρειάζεται λοιπόν, Windows 10, Visual Studio 2015 και Microsoft Visual Studio SDK

Ένα χαρακτηριστικό ακόμα το οποίο είναι σημαντικό και βοηθάει πάρα πολλούς προγραμματιστές, είναι η δήλωση κάθε συσκευή στην οποία γίνεται ανάπτυξη ως συσκευή ανάπτυξης(Developer Device) και να δοκιμάζει ποια είναι εφαρμογή του στο Windows Store.

22 Τύποι Δεδομένων στην Visual Basic

Οι τύποι δεδομένων στη Visual Basic χωρίζονται σε ακέραιους και μεταβλητές γίνεται κινητής υποδιαστολής. Στη συνέχεια αναφέρονται όλοι οι δυνατοί τύποι δεδομένων, όπως αναλυτικά παρουσιάζονται στο site MSDN.

Εφαρμογή Κοινωνικού Περιεχομένου- Ηλεκτρονική Συνταγογράφηση Φαρμάκων

Visual Basic type	Common language runtime type structure	Nominal storage allocation	Value range
Boolean	Boolean	Depends on implementing platform	True or False
Byte	Byte	1 byte	0 through 255 (unsigned)
Char (single character)	Char	2 bytes	0 through 65535 (unsigned)
Date	DateTime	8 bytes	0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999
Decimal	Decimal	16 bytes	0 through +/-79,228,162,514,264,337,593,543,950,335 (+/- 7.9...E+28) [†] with no decimal point; 0 through +/- 7.9228162514264337593543950335 with 28 places to the right of the decimal:
			smallest nonzero number is +/- 0.0000000000000000000000000000001 (+/-1E-28) [†]
			4.94065645841246544E-324 [†] for negative values;
Double (double-precision floating-point)	Double	8 bytes	-1.79769313486231570E+308 through -4.94065645841246544E-324 [†] for positive values
			4.94065645841246544E-324 through 1.79769313486231570E+308 [†] for negative values
			-2,147,483,648 through 2,147,483,647 (signed)
Integer	Int32	4 bytes	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807 (9.2...E+18 [†]) (signed)
Long (long integer)	Int64	8 bytes	-2,147,483,648 through 2,147,483,647 (signed)
Object	Object (class)	4 bytes on 32-bit platform	Any type can be stored in a variable of type Object
		8 bytes on 64-bit platform	
		1 byte	
SByte	SByte	1 byte	-128 through 127 (signed)
Short (short integer)	Int16	2 bytes	-32,768 through 32,767 (signed)
Single (single-precision floating-point)	Single	4 bytes	-3.4028235E+38 through -1.401298E-45 [†] for negative values;
			1.401298E-45 through 3.4028235E+38 [†] for positive values
			0 to approximately 2 billion Unicode characters
String (variable-length)	String (class)	Depends on implementing platform	0 through 4,294,967,295 (unsigned)
UInteger	UInt32	4 bytes	0 through 18,446,744,073,709,551,615 (1.8...E+19 [†]) (unsigned)
ULong	UInt64	8 bytes	0 through 65,535 (unsigned)
User-Defined (structure)	(inherits from ValueType)	Depends on implementing platform	Each member of the structure has a range determined by its data type and independent of the ranges of the other members
UShort	UInt16	2 bytes	0 through 65,535 (unsigned)

Πίνακας 1

22.1 Δομές Ελέγχου

Στη visual basic χρησιμοποιώντας δομές ελέγχου για τον έλεγχο της ροής του προγράμματος ανάλογα με την περίπτωση. Οι πιο συνηθισμένες δομές ελέγχου είναι η if και η switch, οι οποίες πολλές φορές μπορεί να έχουμε την ίδια χρήση στο πρόγραμμα. Παρόλα αυτά έχουν την διαφορά ότι στην πρώτη (στην if), αν ικανοποιείται η συνθήκη τότε το πρόγραμμα συνεχίζει όπως το έχει θέσει ο χρήστης, ενώ αν όχι χρησιμοποιείται else ώστε να συμβεί κάτι άλλο. Στη δεύτερη (switch) είναι από την αρχή επιλεγμένες οι περιπτώσεις του τι θα συμβεί κάτω από τον έλεγχο της ίδιας πρότασης.

22.2 Δομές Επανάληψης

Στον προγραμματισμό η δομή επανάληψης χρησιμοποιείται στο πρόγραμμά μας όταν θέλουμε να επαναλάβουμε ένα κομμάτι κώδικα. Για την το λόγο αυτό , έχουμε δύο εντολές την for while. Αν και μεταξύ τους έχουν κάποια κοινά χαρακτηριστικά και έναν κοινό στόχο, ο τρόπος προσέγγισης αλλά και η διεξαγωγή της επανάληψης διαφέρει από τη μια στη άλλη.

22.3 Η δομή for

Η for ως η πιο απλή δομή, την χρησιμοποιούμε όταν γνωρίζουμε εκ των προτέρων τον αριθμό των επαναλήψεων. Αυτό σημαίνει πως ο κώδικας ο οποίος περιέχεται στη δομή for θα εκτελεστεί για καθορισμένο αριθμό βημάτων, εν αντιθέσει με την

22.4 Η δομή while

Η οποία χρησιμοποιείται ώστε να πραγματοποιηθεί η επανάληψη για όσες φορές ικανοποιούνται συγκεκριμένες συνθήκες. Θεωρητικά , ανάλογα με την είσοδο που λαμβάνει, μπορεί να μην τρέξει ούτε μία φορά, ενώ οι επαναλήψεις διαφέρουν ως προς το πλήθος ανάλογα των δεδομένων εισόδου.

23 Πλατφόρμα Ανάπτυξης

Ως πλατφόρμα ανάπτυξης επιλέγει η Visual Studio 2015 που έχει σημαντικές βελτιώσεις σε σχέση με τις αντίστοιχες Visual Studio 2010 και Visual Studio 2013 ιδιαίτερα σε ότι αφορά Universal Application WEB Application ASP.NET.

Κρίνεται όμως σκόπιμο εδώ να αναφερθούμε συγκριτικά σε δυο διαφορετικές γενιές Visual Studio συγκεκριμένα Visual Studio 6 SP 6 και Visual Studio 2015 .

24 Visual Studio 2015

Το πρόγραμμα Microsoft Visual Studio 2015 αποτελεί το βασικό εργαλείο που χρησιμοποιούν οι προγραμματιστές για να αναπτύξουν εφαρμογές οι οποίες τρέχουν σε windows και windows phone.

Η νέα αυτή πλατφόρμα σχεδιασμένη κυρίως για την ανάπτυξη εφαρμογών σε όλες τις πλατφόρμες όπως Mobile Phones, tablets, Ipads (UNIVERSAL APPLICATION) κυρίως σε Cloud servers και βασικότερα στην πλατφόρμα AZURE της Microsoft.

Υποστηρίζει όλες της Γλώσσες προγραμματισμού και όλες τις βάσεις Δεδομένων, ακόμη και σε απομακρυσμένους υπολογιστές, συνδεδεμένη με τον sql server με IP+port.

Συνδέεται με Data record sets με connection Strings ή ODBC.

Ένα σημαντικό μειονέκτημα είναι ότι οι μεταβολές στους πίνακες δεν μεταφέρονται στην εφαρμογή, παρά μόνο με νέο Data record set. Βασικό πρόβλημα με δεδομένο ότι τα queries Insert, Select, Update σχεδιάζονται εσωτερικά στους πίνακες του Data Record Sets

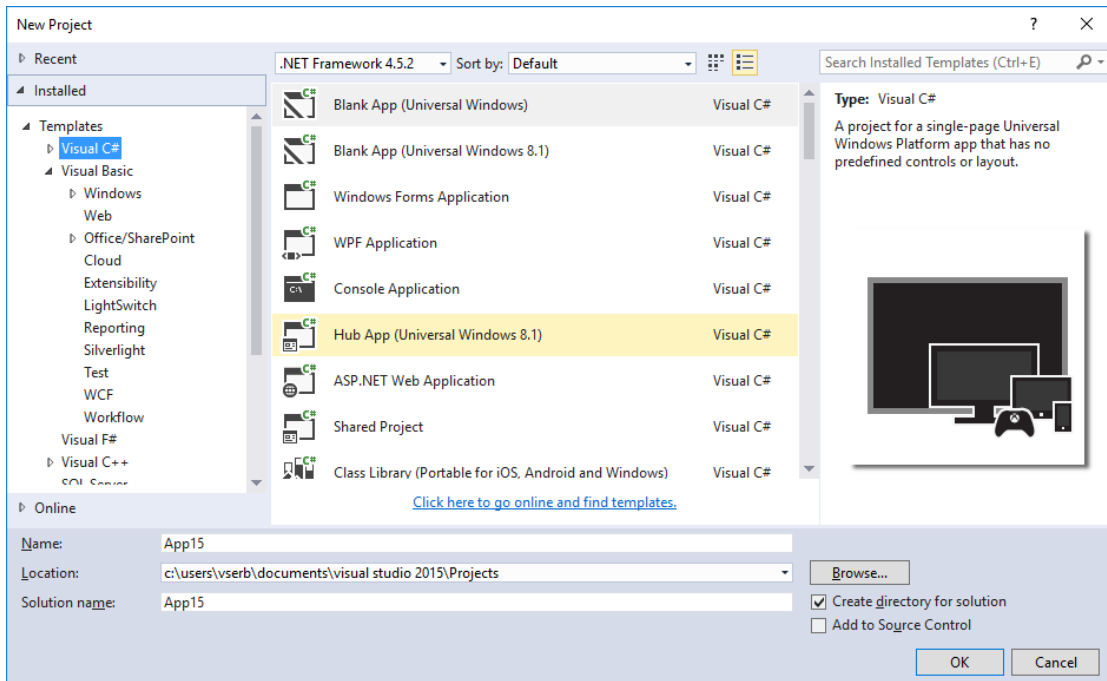
Οπότε θα πρέπει να ξανασχεδιαστούν αν θέλουμε κατά ελάχιστον να τροποποιήσουμε τα δεδομένα τα οποία εισάγονται στην εφαρμογή.

Η μεταφορά των στοιχείων των πινάκων γίνεται με queries που εκτελείται στο ON LOAD Event κάθε φόρμας

Μικρά bugs εντοπίζονται σε διάφορα control events όπως FORM ACTIVATION BUTTON ON CLICK στα οποία διαπιστώσαμε ότι δεν κάνουν πάντα ότι υπόσχεται το όνομά τους

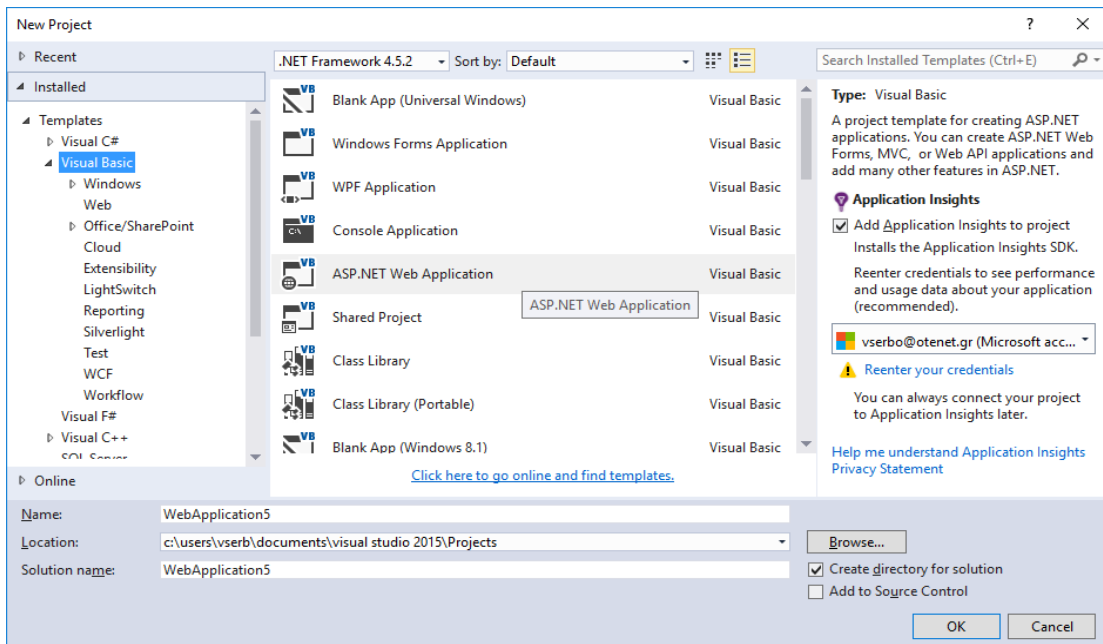
Πέρα όμως από αυτά είναι πολύ εύκολη η μεταφορά σε άλλους υπολογιστές με πολύ απλό setup package που θεωρητικά και ένας απλός χρήστης θα μπορούσε να το κάνει

Οι εφαρμογές που μπορούν να αναπτυχθούν σε κάθε γλώσσα προγραμματισμού φαίνονται στις παρακάτω εικόνες

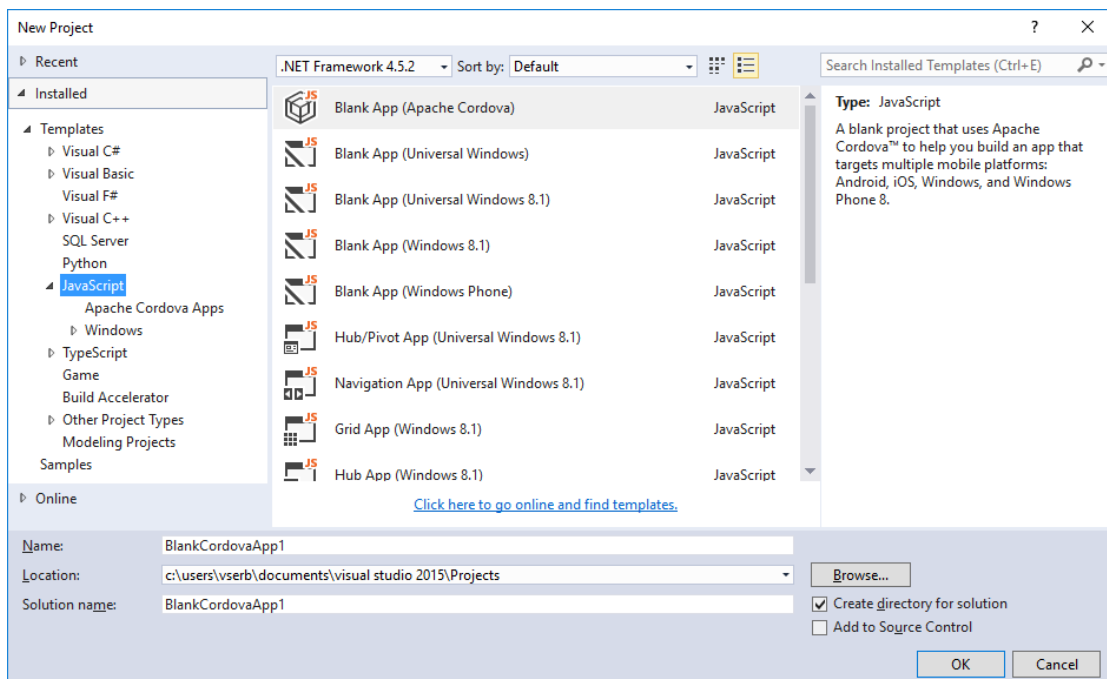


Εικόνα 1

Εφαρμογή Κοινωνικού Περιεχομένου- Ηλεκτρονική Συνταγογράφηση Φαρμάκων



Εικόνα 2



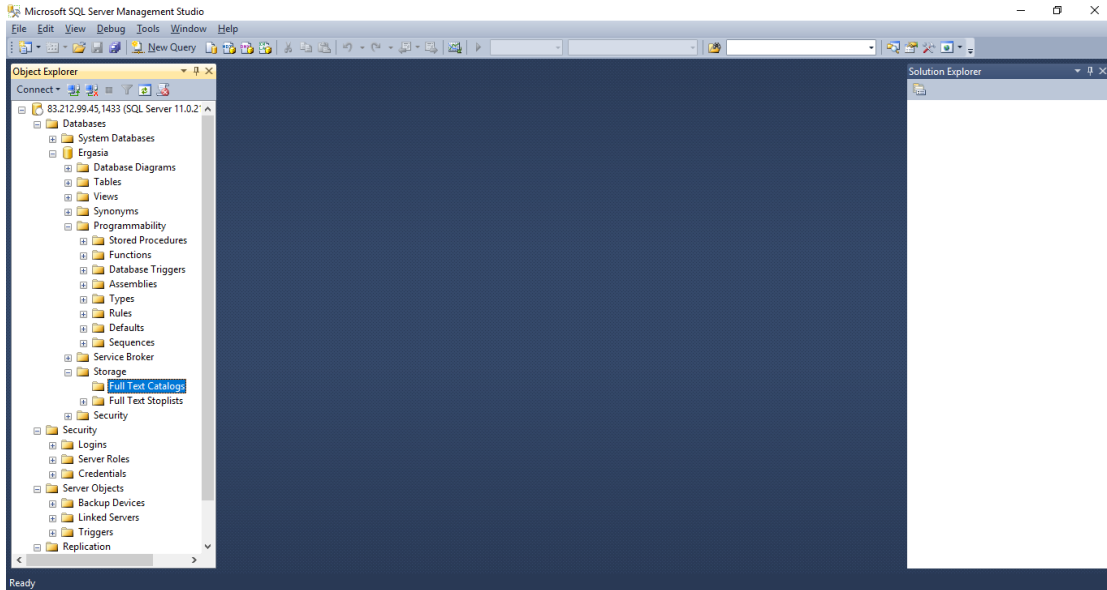
Εικόνα 3

Εξετάζοντας τις διαφορές με το νέο Visual Studio 2017 (trial Version) δεν διακρίνει κάτι σημαντικό πέρα από τις βελτιώσεις που αφορούν την πλατφόρμα AZURE.

25 Βάση Δεδομένων SQL2012

Για την βάση δεδομένων επιλέγει ο **SQL2012 R2**. Ένα πολύ σταθερό περιβάλλον που παρέχει δυνατότητες απομακρυσμένης διαχείρισης.

Δεν έχει διαφορές στο σχεδιασμό πινάκων η views από άλλες Version του SQL SERVER. Παρουσιάζει όμως περισσότερες ευκολίες στο σχεδιασμό Store Procedures καθώς και διάφορα επιπλέον εργαλεία διαχείρισης δεδομένων.



Εικόνα 4

26 Πίνακες

26.1 Πίνακας Πελατών(Dbo.Customers)

Πεδίο	Τύπος	Περιγραφή
Cust_Id	int	Βασικό Κλειδί Cust_id
Cust_Lname	nvarchar(50)	Επώνυμο
Cust_Fname	nvarchar(50)	Όνομα
Cust_Address	nvarchar(50)	Διεύθυνση
Cust_Region	nvarchar(50)	Περιοχή
Cust_ZIP	nchar(10)	T.K
Cust_Phone	nvarchar(20)	Τηλέφωνο
Cust_mail	nvarchar(50)	Email
Cust_AMKA	nvarchar(20)	A.M.K.A
Cust_AFM	nvarchar(20)	A.Φ.Μ
Cust_TYPE	nchar(10)	Ταμείο Ασφάλισης

Πίνακας 2

26.2 Πίνακας Γιατρών (Dbo.Iatroi)

Πεδίο	Τύπος	Περιγραφή
DoctorId	int	Κωδικός
LastNmae	nvarchar(255)	Επώνυμο
FirstName	nvarchar(255)	Όνομα
FatherName	nvarchar(255)	Πατρώνυμο
Specialty	nvarchar(255)	Ειδικότητα
Address	nvarchar(255)	Διεύθυνση
Region	nvarchar(255)	Περιοχή
ZIP	nvarchar(255)	T.K
Phone	nvarchar(255)	Τηλέφωνο

Πίνακας 3

26.3 Πίνακας Φαρμάκων

Πεδίο	Τύπος	Περιγραφή
SqId	int	Unchecked
EOF_CODE	nvarchar(255)	Κωδικός EOF
DESCRIPTION	nvarchar(255)	Περιγραφή
INCREMET	nvarchar(255)	Δραστική Ουσία
LICENSE	nvarchar(255)	Αδεία

Πίνακας 4

26.4 Πίνακας Φαρμακείων

Πεδίο	Τύπος	Περιγραφή
Sore_Id	int	Unchecked
Store_Name	nvarchar(50)	Επωνυμία
Sore_Adress	nvarchar(50)	Διεύθυνση
Store_Region	nvarchar(50)	Περιοχή
Sore_zip	nchar(10)	T.K

Πίνακας 5

26.5 Πίνακας Αρίθμησης Συνταγών

Πεδίο	Τύπος	Περιγραφή
RESPNUM	int	A/A Εγγραφής
RECNUM	int	Πλήθος Εγγραφών

Πίνακας 6

26.6 Βασικός Πίνακας Συνταγών

Πεδίο	Τύπος	Περιγραφή
SqlId	int	Unchecked
Res_Id	int	A/A Συνταγής
Res_Date	date	Ημερομηνία Καταχώρησης
Res_Doc	nchar(10)	Κωδικός Ιατρού
Res_Cust	int	Κωδικός Πελάτη
Res_Rems	nvarchar(50)	Παρατηρήσεις
Store01	nchar(10)	Φαρμακείο1
Store02	nchar(10)	Φαρμακείο2
Store03	nchar(10)	Φαρμακείο3
Res_Check	int	Unchecked
Res_descr	nchar(20)	Ένδειξη Εκτέλεσης

Πίνακας 7

26.7 Αναλυτικός Πίνακας Συνταγών

Πεδίο	Τύπος	Περιγραφή
SqlId	int	Unchecked
Res_Num	int	A/A Συνταγής
Drag_Eof	nvarchar(255)	Κωδικός Φαρμάκου
Drag_Desc	nvarchar(255)	Περιγραφή
Drag_Quant	int	Ποσότητα

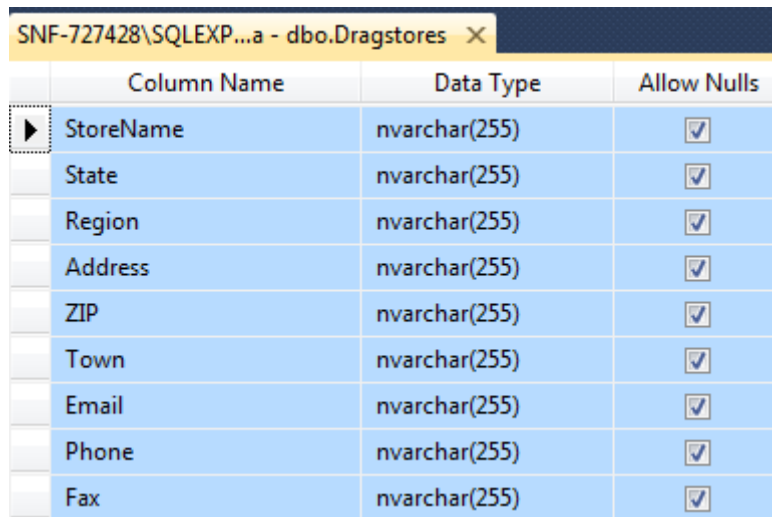
Πίνακας 8

27 Γενικές Παρατηρήσεις Δεδομένων

Τα στοιχεία των πελατών αντλήθηκαν από άλλη εφαρμογή και τροποποιήθηκαν για λόγους διασφάλισης των προσωπικών δεδομένων αλλά και για τις ανάγκες της εφαρμογής.

Τα στοιχεία των γιατρών αντλήθηκαν από το αρχείο του ιατρικού συλλόγου.

Τα στοιχεία των φαρμακείων δημιουργήθηκε για να καλύψει τις ανάγκες της εφαρμογής ειδικά για την δυνατότητα επιλογής φαρμακείων της περιοχής του πελάτη. Το αρχείο Φαρμάκων προέρχεται από ανοικτές πηγές δεδομένων, κατάλληλα τροποποιημένο, εφόσον υπάρχει στο Internet αρχείο φαρμακείων, αλλά δεν κάλυπτε πλήρως τις ανάγκες της εφαρμογής όσον αφορά τις περιοχές.



Column Name	Data Type	Allow Nulls
StoreName	nvarchar(255)	<input checked="" type="checkbox"/>
State	nvarchar(255)	<input checked="" type="checkbox"/>
Region	nvarchar(255)	<input checked="" type="checkbox"/>
Address	nvarchar(255)	<input checked="" type="checkbox"/>
ZIP	nvarchar(255)	<input checked="" type="checkbox"/>
Town	nvarchar(255)	<input checked="" type="checkbox"/>
Email	nvarchar(255)	<input checked="" type="checkbox"/>
Phone	nvarchar(255)	<input checked="" type="checkbox"/>
Fax	nvarchar(255)	<input checked="" type="checkbox"/>

Εικόνα 5

Εφαρμογή Κοινωνικού Περιεχομένου- Ηλεκτρονική Συνταγογράφηση Φαρμάκων

StoreName	State	Region	Address	ZIP	Town	Email	Phone	Fax
1 Κεντρικό Φαρμακείο	Αττικής	Αμαρουσίου	Απ. Παύλου 12	15123	Μαρούσι	kentrikofamakeio@eopyy.gov.gr	2108110686, 2108110687	2108110693
2 Φαρμακείο Αθήνας	Αττικής	Αθηναίων	Αγ. Κωνσταντίνου 16	10241	Αθήνα	famakeioathinon@eopyy.gov.gr	2105213686, 2105213688	2105200021
3 Φαρμακείο Αλεξάνδρας	Αττικής	Αθηναίων	Λ. Αλεξάνδρας 119	11475	Αθήνα	famakeioalex@gmail.com	2106479713	2106401466
4 Φαρμακείο Αλεξανδρούπολης	Εβρου	Αλεξανδρούπολης	Αν. Θράκης 58	68100	Αλεξανδρούπολη	alexandroupofamakeio@eopyy.gov.gr	2551085077	2551085065
5 Φαρμακείο Βόλου	Μαγνησίας	Βόλου	Θρακών 20	38333	Βόλος	volosfamakeio@eopyy.gov.gr	2421020625	2421039834
6 Φαρμακείο Γκύζη	Αττικής	Αθηναίων	Παράσκου 23	11473	Γκύζη	gyzfamakeio@eopyy.gov.gr	2106424940	2106495281
7 Φαρμακείο Εύοσμου	Θεσσαλονίκης	Θεσσαλονίκης	Καραολή Δημητρίου 12 & Κοσμά Αιτωλού 1	56224	Εύοσμος	evosmosfamakeio@eopyy.gov.gr	2310709200	
8 Φαρμακείο Ηρακλείου	Ηρακλείου	Ηρακλείου	3ο χιλμ Εθνικής Οδού Ηρακλείου-Φαιστού	71500	Ηράκλειο	iraklifamakeio@eopyy.gov.gr	2810280671, 2810719224	2810224604
9 Φαρμακείο Ιωαννίνων	Ιωαννίνων	Ιωαννίνων	Παύλου Μελά 23-25	45444	Ιωάννινα	ioanninafamakeio@eopyy.gov.gr	2651077930, 2651083631	2651078239
10 Φαρμακείο Καβάλας	Καβάλας	Καβάλας	Φιλικής Εταιρείας 6	65403	Καβάλα	kavalafamakeio@eopyy.gov.gr	2510450524, 2510450525	2510834528
11 Φαρμακείο Καλλιθέας	Αττικής	Καλλιθέας	Δαβάκη 44	17672	Καλλιθέα	kallitheatfamakeio@eopyy.gov.gr	2109594504	2109594505
12 Φαρμακείο Κατερίνης	Πελαίας	Κατερίνης	Κανελάουλου Παναγιώτη 1	60100	Κατερίνη	ped43.eargy@eopyy.gov.gr	23510-47215, -47210, -74977	2351074977
13 Φαρμακείο Κέντρου Θεσσαλονίκης	Θεσσαλονίκης	Θεσσαλονίκης	Αριστοτέλους 19	54624	Θεσσαλονίκη	thessalonikifamakeio@eopyy.gov.gr	2310254350	2310254347
14 Φαρμακείο Κέρκυρας	Κέρκυρας	Κέρκυρας	3η Παρ. Ι. Θεοτόκη	49100	Κέρκυρα	kerkyrafamakeio@eopyy.gov.gr	2661096905	2661096902
15 Φαρμακείο Κομοτηνής	Ροδόπης	Κομοτηνής	Υψηλάντου 2	69100	Κομοτηνή	komotinafamakeio@eopyy.gov.gr	2531059293	
16 Φαρμακείο Λαμίας	Φθιώτιδας	Λαμίων	Θερμοπυλών 77	35100	Λαμία	lamiafamakeio@eopyy.gov.gr	2231056473	2231056472
17 Φαρμακείο Λάρισας	Λάρισας	Λαρισίων	Ρούσβελτ 4	41222	Λάρισα	larisafamakeio@eopyy.gov.gr	2410534180, 2410534095	2410534763
18 Φαρμακείο Μυτιλήνης	Λέσβου	Μυτιλήνης	Πλ. Μαρτύρων	81100	Μυτιλήνη	lesvosfamakeio@eopyy.gov.gr	2251057440	2251057430
19 Φαρμακείο Νέας Ιωνίας	Αττικής	Νέας Ιωνίας	Αλ. Παπαγαύλη 91	14234	Ν. Ιωνία	nealionfamakeio@eopyy.gov.gr	2102758737	2102775080
20 Φαρμακείο Πατρών	Αχαΐας	Πατρών	Γιαννιτών 3	26223	Πάτρα	patrafamakeio@eopyy.gov.gr	2610466553, 2610466555	2610466554

Εικόνα 6

27.1 Application DataSetEgasia

Εδώ παρουσιάζονται οι πηγές δεδομένων με τον τρόπο που έχουν εισαχθεί στο vs 2015

The screenshot shows the Visual Studio 2015 DataSet Designer for 'ErgasiaDataSet'. The designer displays several data sources and their associated TableAdapters. Each data source is connected to a TableAdapter, which is then linked to a DataSet. The data sources include AthensZip, Customers, iatroi, LASTRESP, ReceiptsMain, DRAGS, and ReceiptsDetails. Each TableAdapter has a 'Fill, GetData ()' method associated with it. The DataSet Designer also shows the 'Fill, GetData ()' method for the entire DataSet.

Εικόνα 7

28 Γενικά σχόλια για τον κώδικα και τα CONTROLS στις φόρμες

Η ονομασία των controls διατηρήθηκε η default.

Π.χ.

textbox1 textbox2

Datagridview1,Datagridview2

Button1,Button2

MenuStrip1

Στις Φόρμες η ονομασία δόθηκε ανάλογα με την λειτουργία κάθε μίας

To windowsState=normal

StartPosition =CenterScreen

Η διαχείριση των πινάκων γίνεται βασικά με εσωτερικά queries π.χ fill, fillby ,insertquerie,updatequerie και με databindigs σε Datagrids,textboxies

Η επιλογή των τιμών των πεδίων γίνεται είτε με μεταφορά των τιμών των CellContent

σε textboxies π.χ

TextBox5.Text = Me.DataGridView6.Rows(e.RowIndex).Cells(3).Value

TextBox6.Text = Me.DataGridView6.Rows(e.RowIndex).Cells(2).Value

Είτε με μεταφορά των τιμών του current Record στις κοινές μεταβλητές π.χ

StoreName = Me.DragStoreBindingSource.Current("Store_Name")

Στο event DataGridView CellClick

Σε ορισμένα controls ενεργοποιείται το ToolTip με πληροφορίες για την λειτουργία

Π.χ **Ανοιγμα φόρμας Συνταγών**

Στην φόρμα 2 καταχώρηση των διορθώσεων ή νέων καταχωρήσεων στο αρχείο πελατών γίνεται με function key (F10).

Στα αρχεία γιατρών φαρμακείων και φαρμάκων δεν έχει σχεδιαστεί διόρθωση η καταχώρηση νέων εγγραφών. Θα πρέπει να προβλεφθεί ενημέρωση της βάσης από εξωτερικά σωστά ενημερωμένα στοιχεία των αντίστοιχων πινάκων.

29 Εφαρμογή για Mobile, Tablet & Desktop σε C# (Windows 10 μέσω του Store)

29.1 Αρχική φόρμα

Στην αρχική φόρμα έχουμε την επιλογή να επιλέξουμε την ειδικότητα μας (Ιατρός ή Φαρμακοποιός) και στο πεδίο από κάτω πληκτρολογούμε τον κωδικό που έχει δοθεί από τους διαχειριστές. Έπειτα πατάμε τον κουμπί “Είσοδος” και η εφαρμογή μας κατευθύνει στην αντίστοιχη φόρμα.



Εικόνα 8

29.2 Φόρμα Ιατρού

20

e-Prescription
Καλωσήρθες Ιατρέ

Στοιχεία Ιατρού: ΑΒΡΑΜΙΔΗΣ ΒΑΣΙΛΕΙΟΣ Ειδικότητα: ΩΤΟΡΙΝΟΛΑΡΥΓΓΟΛΟΓΟΣ Τηλέφωνο: 2107253109

Αναζήτηση Πελάτη: Εισάγετε Α.Μ.Κ.Α. Αναζήτηση

ΚΩΔΙΚΟΣ Ε.Ο.Φ. - ΟΝΟΜΑ - ΔΡΑΣΤΙΚΗ ΟΥΣΙΑ ΦΑΡΜΑΚΟΥ : ΚΑΤΑΧΩΡΗΜΕΝΑ ΦΑΡΜΑΚΑ :

1854104 ΑΡΟΤΕΛ ΡΑΡΑCΕΤΑΜΟΛ
1854105 ΑΡΟΤΕΛ ΡΑΡΑCΕΤΑΜΟΛ
1854102 ΑΡΟΤΕΛ ΡΑΡΑCΕΤΑΜΟΛ
1854115 ΑΡΟΤΕΛ ΡΑΡΑCΕΤΑΜΟΛ
1854113 ΑΡΟΤΕΛ ΡΑΡΑCΕΤΑΜΟΛ
1854116 ΑΡΟΤΕΛ ΡΑΡΑCΕΤΑΜΟΛ
1854112 ΑΡΟΤΕΛ ΡΑΡΑCΕΤΑΜΟΛ
1854114 ΑΡΟΤΕΛ ΡΑΡΑCΕΤΑΜΟΛ

Επιλεγμένο Φάρμακο:

Ποσότητα:

Καθαρισμός Φαρμάκων

Αποσύνδεση

Εικόνα 9

Για να δημιουργήσει μια συνταγή πρώτα βρίσκει τα στοιχεία του πελάτη από το Α.Μ.Κ.Α του και πατάει το κουμπί “Αναζήτηση” και το σύστημα αυτόματα φέρνει τα στοιχεία του.

20

e-Prescription
Καλωσήρθες Ιατρέ

Στοιχεία Ιατρού: ΑΒΡΑΜΙΔΗΣ ΒΑΣΙΛΕΙΟΣ Ειδικότητα: ΩΤΟΡΙΝΟΛΑΡΥΓΓΟΛΟΓΟΣ Τηλέφωνο: 2107253109

Αναζήτηση Πελάτη: 1511010101 Αναζήτηση

Ονοματεπώνυμο: ΞΕΝΟΥ ΘΕΩΝΗ Διεύθυνση: 1511 Διεύθυνση Πόλη: ΒΟΥΛΙΑΓΜΕΝΗ Τ.Χ: 166 71 Τηλέφωνο: 15110909 e-mail: ΘΕΩΝΗ@mail.com

Εικόνα 10

Έπειτα ο Ιατρός επιλέγει το φάρμακο που θέλει να χορηγήσει από την λίστα, το εμφανίζει ακριβώς δίπλα για επαλήθευση, πληκτρολογεί από κάτω την ποσότητα και πατάει το “Καταχώρηση Φαρμάκου”.



Εικόνα 11

Σε περίπτωση επιτυχούς καταχώρησης μας εμφανίζει το φάρμακο στην λίστα με τα καταχωρημένα φάρμακα καθώς εμφανίζει και μήνυμα κάτω δεξιά “Καταχώρηση επιτυχής!”

Σε περίπτωση λάθους (για παράδειγμα να μην έχουμε βάλει ποσότητα φαρμάκου ή να μην έχουμε καταχωρήσει τον πελάτη μέσω Α.Μ.Κ.Α.), μας εμφανίζει σχετικό μήνυμα λάθους.



Εικόνα 12

Τέλος, Με το κουμπί κάτω δεξιά “Καθαρισμός φαρμάκων” καθαρίζει την λίστα των καταχωρημένων φαρμάκων. Και εμφανίζει σχετικό μήνυμα “Διαγραφή Επιτυχής”



Εικόνα 13

Με το κουμπί “Αποσύνδεση” επιστρέφουμε στην αρχική φόρμα.

29.3 Φόρμα Φαρμακοποιού

Κατά την είσοδο του φαρμακοποιού με τον κωδικό του το σύστημα εμφανίζει τα στοιχεία του φαρμακείου του. Εκείνος εισάγει το Α.Μ.Κ.Α. του πελάτη και πατάει το κουμπί αναζήτησης και εκτός από τα στοιχεία του πελάτη το σύστημα εμφανίζει και την συνταγή που είναι προς εκτέλεση με την λίστα των φαρμάκων.



Εικόνα 14

Πατώντας το κουμπί "Εκτέλεση συνταγής" η συνταγή σβήνεται από την βάση δεδομένων μας και καθαρίζει το Α.Μ.Κ.Α. του πελάτη από την συνταγή, καθώς και εμφανίζεται και σχετικό μήνυμα κάτω από το κουμπί.

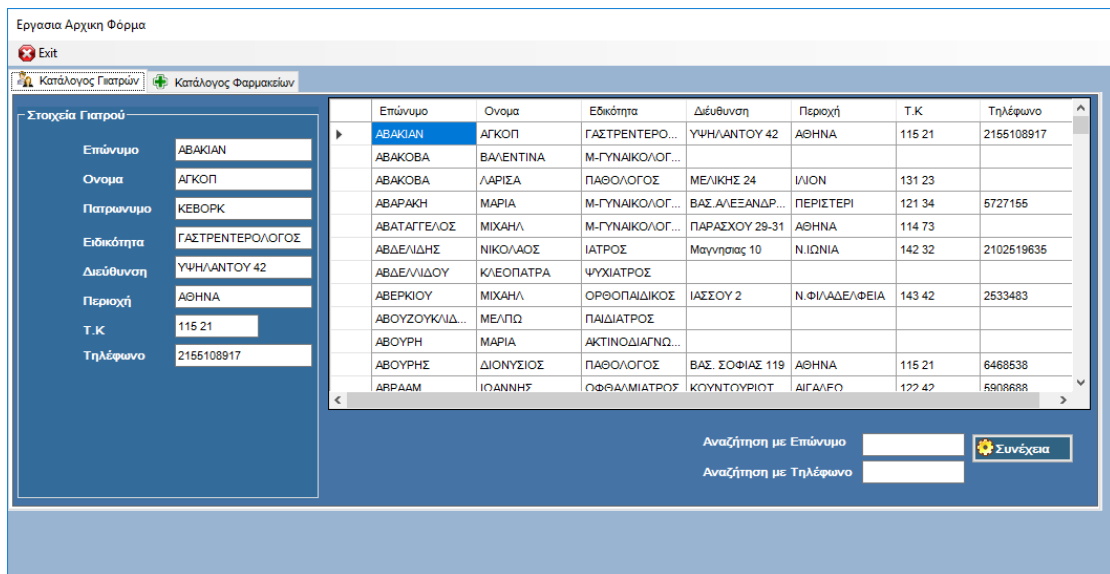


Εικόνα 15

30 Εφαρμογή για Desktop σε Visual Basic (Windows 7 και προγενέστερες εκδόσεις)

30.1 ΛΕΙΤΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ

Με το άνοιγμα της εφαρμογής εμφανίζεται η αρχική φόρμα στο TAB με τον κατάλογο γιατρών, όπου μας εμφανίζει όλους τους γιατρούς που είναι καταχωρημένοι στην βάση. Στο παράθυρο αυτό ο γιατρός επιλέγει τα στοιχεία του καθώς μπορεί να αναζητήσει τα στοιχεία του με βάση το επώνυμο ή το τηλέφωνο και πατάει το κουμπί “Συνέχεια”.



Εικόνα 16

Στο παράθυρο αυτό ο γιατρός επιλέγει τα στοιχεία του καθώς μπορεί να αναζητήσει τα στοιχεία του με βάση το επώνυμο ή το τηλέφωνο και πατάει το κουμπί “Συνέχεια”.

Είναι δυνατή η αναζήτηση με μέρος του επώνυμο η του τηλεφώνου

Εφαρμογή Κοινωνικού Περιεχομένου- Ηλεκτρονική Συνταγογράφηση Φαρμάκων

Εργασία Αρχική Φόρμα

Exit

Κατάλογος Γιατρών Κατάλογος Φαρμακείων

Στοιχεία Γιατρού

Επώνυμο: ΒΑΣΙΛΕΙΔΟΥ
 Όνομα: ΣΟΦΙΑ
 Πατρωνυμο: ΦΑΙΔΩΝΑ
 Ειδικότητα: ΨΥΧΙΑΤΡΟΣ
 Διεύθυνση: Λ. ΜΑΡΑΘΩΝΟΣ 10-12
 Περιοχή: ΔΡΟΣΙΑ
 Τ.Κ: 145 72
 Τηλέφωνο: 2108136668

Επώνυμο	Όνομα	Ειδικότητα	Διεύθυνση	Περιοχή	Τ.Κ	Τηλέφωνο
ΒΑΣΙΛΕΙΔΟΥ	ΔΗΜΗΤΡΑ	ΙΑΤΡΟΣ				
ΒΑΣΙΛΕΙΔΟΥ	ΕΥΗ	ΧΕΙΡΟΥΡΓΟΣ Π...	ΑΧΕΛΩΟΥ 38	ΒΡΙΛΗΣΣΙΑ	152 35	6013660
ΒΑΣΙΛΕΙΔΟΥ	ΚΑΛΛΙΝΙΚΗ	Μ-ΓΥΝΑΙΚΟΛΟΓ...	ΚΗΦΙΣΙΑΣ 105	ΑΘΗΝΑ	115 26	6984024
ΒΑΣΙΛΕΙΔΟΥ	ΜΑΡΙΑ	ΓΕΝΙΚΟΣ ΙΑΤΡΟΣ				
ΒΑΣΙΛΕΙΔΟΥ	ΣΟΦΙΑ	ΨΥΧΙΑΤΡΟΣ	Λ. ΜΑΡΑΘΩΝΟ...	ΔΡΟΣΙΑ	145 72	2108136668
ΒΑΣΙΛΕΙΑΣ	ΔΗΜΗΤΡΙΟΣ	ΙΑΤΡΟΣ				
ΒΑΣΙΛΕΙΟΥ	ΑΘΑΝΑΣΙΟΣ	ΠΑΘΟΛΟΓΟΣ	ΓΕΝΝΗΜΑΤΑ 19	ΑΘΗΝΑ	115 24	6915500
ΒΑΣΙΛΕΙΟΥ	ΑΛΕΞΑΝΔΡΑ	ΩΤΟΡΙΝΟΛΑΡΥ...				
ΒΑΣΙΛΕΙΟΥ	ΑΛΕΞΙΟΣ	ΩΤΟΡΙΝΟΛΑΡΥ...	Λ. ΠΑΠΑΓΟΥ 21	ΖΩΓΡΑΦΟΥ	157 73	7781327
ΒΑΣΙΛΕΙΟΥ	ΑΝΤΩΝΙΟΣ	ΠΝΕΥΜΟΝΟΛΟ...				
ΒΑΣΙΛΕΙΟΥ	ΑΠΟΣΤΟΛΟΣ	ΓΕΝΙΚΟΣ ΙΑΤΡΟΣ				
ΒΑΣΙΛΕΙΟΥ	ΑΡΙΣΤΕΙΔΗΣ	ΝΕΥΡΟΛΟΓΟΣ				

Αναζήτηση με Επώνυμο: ΒΑΣΙΛΕ

Αναζήτηση με Τηλέφωνο:

Συνέχεια

Εικόνα 17

Εργασία Αρχική Φόρμα

Exit

Κατάλογος Γιατρών Κατάλογος Φαρμακείων

Στοιχεία Γιατρού

Επώνυμο: ΣΠΑΛΑΣ
 Όνομα: ΕΥΣΤΑΘΙΟΣ
 Πατρωνυμο: ΙΩΑΝΝΗ
 Ειδικότητα: ΟΦΘΑΛΜΙΑΤΡΟΣ
 Διεύθυνση: ΔΗΜΑΡΧΕΙΟΥ 18
 Περιοχή: ΑΙΓΑΛΕΩ
 Τ.Κ: 122 41
 Τηλέφωνο: 5902754

Επώνυμο	Όνομα	Ειδικότητα	Διεύθυνση	Περιοχή	Τ.Κ	Τηλέφωνο
ΣΠΑΛΑΣ	ΕΥΣΤΑΘΙΟΣ	ΟΦΘΑΛΜΙΑΤΡΟΣ	ΔΗΜΑΡΧΕΙΟΥ 18	ΑΙΓΑΛΕΩ	122 41	5902754
ΓΙΑΝΝΟΥΛΗ	ΓΕΩΡΓΙΑ	ΔΕΡΜΑΤΟΛΟΓ...	ΔΗΜΑΡΧΕΙΟΥ 8	ΑΙΓΑΛΕΩ	122 42	5902069

Αναζήτηση με Επώνυμο:

Αναζήτηση με Τηλέφωνο: 5902

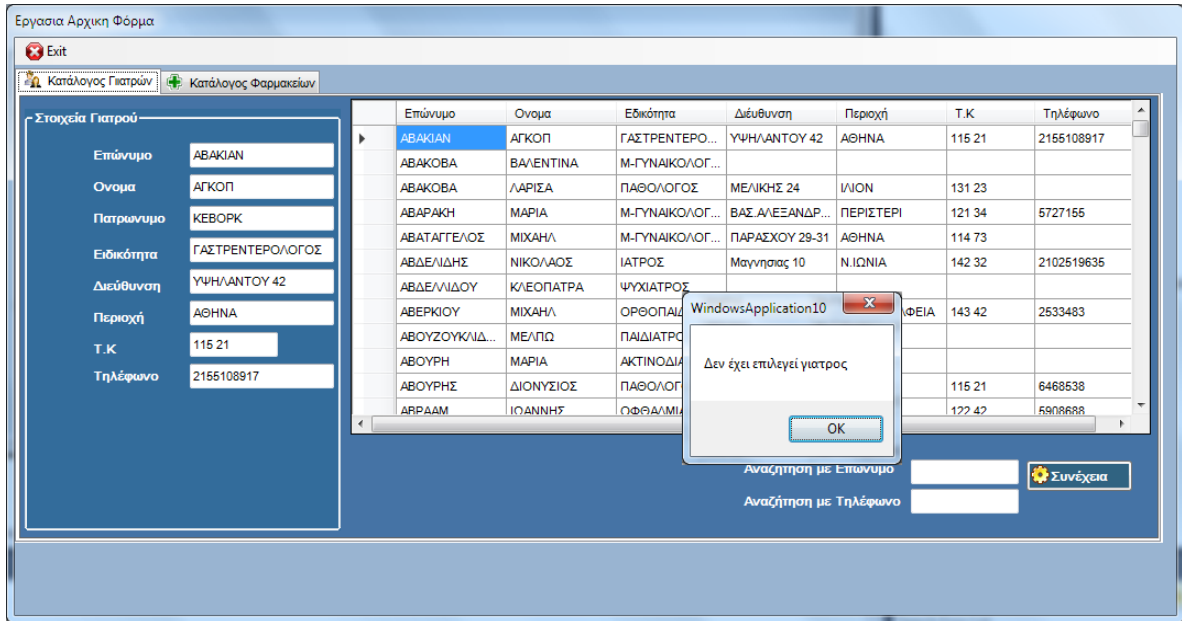
Συνέχεια

Εικόνα 18

Με το button “Συνέχεια” μεταφέρεται στην φόρμα συνταγών

Αν δεν έχει επιλεγεί γιατρός εμφανίζεται σχετικό μήνυμα

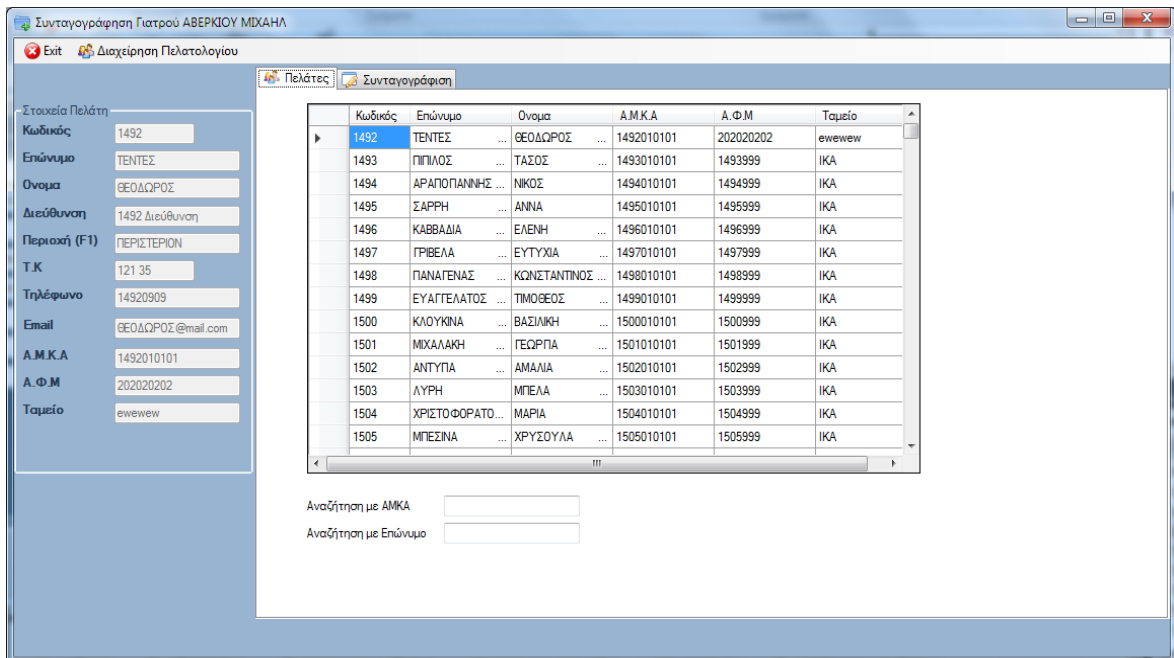
Εφαρμογή Κοινωνικού Περιεχομένου- Ηλεκτρονική Συνταγογράφηση Φαρμάκων



Εικόνα 19

Με το OK η εφαρμογή επιστρέφει στην επιλογή γιατρού

3.0.2 Φόρμα συνταγογράφησης



Εικόνα 20

Στην επικεφαλίδα της φόρμας εμφανίζεται το Επώνυμο και το Όνομα του γιατρού.

Με το άνοιγμα της φόρμας εμφανίζονται όλοι οι πελάτες που είναι καταχωρημένοι στην βάση, και αναζητά τα στοιχεία του πελάτη που θέλει να καταχωρήσει την συνταγή , αφού επιλέξει τον εκάστοτε πελάτη πάει στο TAB “Συνταγογράφηση”

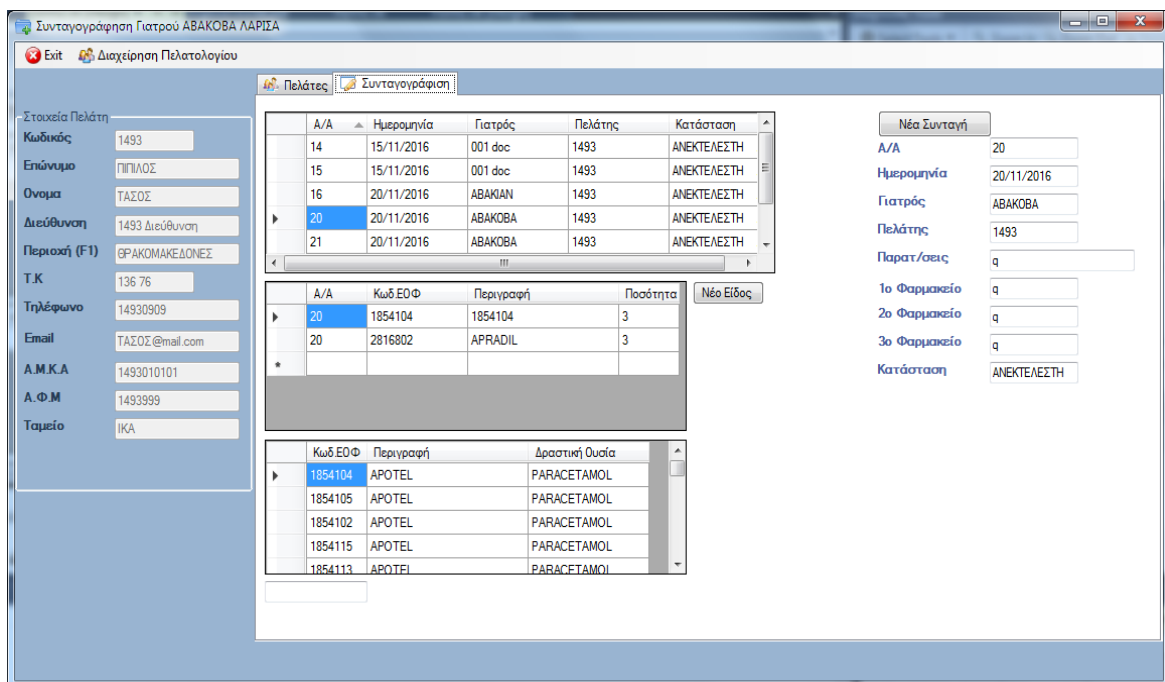
Είναι δυνατή η αναζήτηση πελάτη είτε με ΑΜΚΑ είτε με το Επώνυμο Πελάτη στα αντίστοιχα πεδία.

Στην συνέχεια η εφαρμογή μεταφέρεται στην δεύτερη σελίδα των συνταγών, εκεί εμφανίζονται οι συνταγές του πελάτη ανεξαρτήτως ποιός γιατρός την έχει συνταγογραφήσει.

Στο πάνω πεδίο εμφανίζονται τα βασικά στοιχεία της συνταγής καθώς και αν η συνταγή έχει εκτελεστεί.

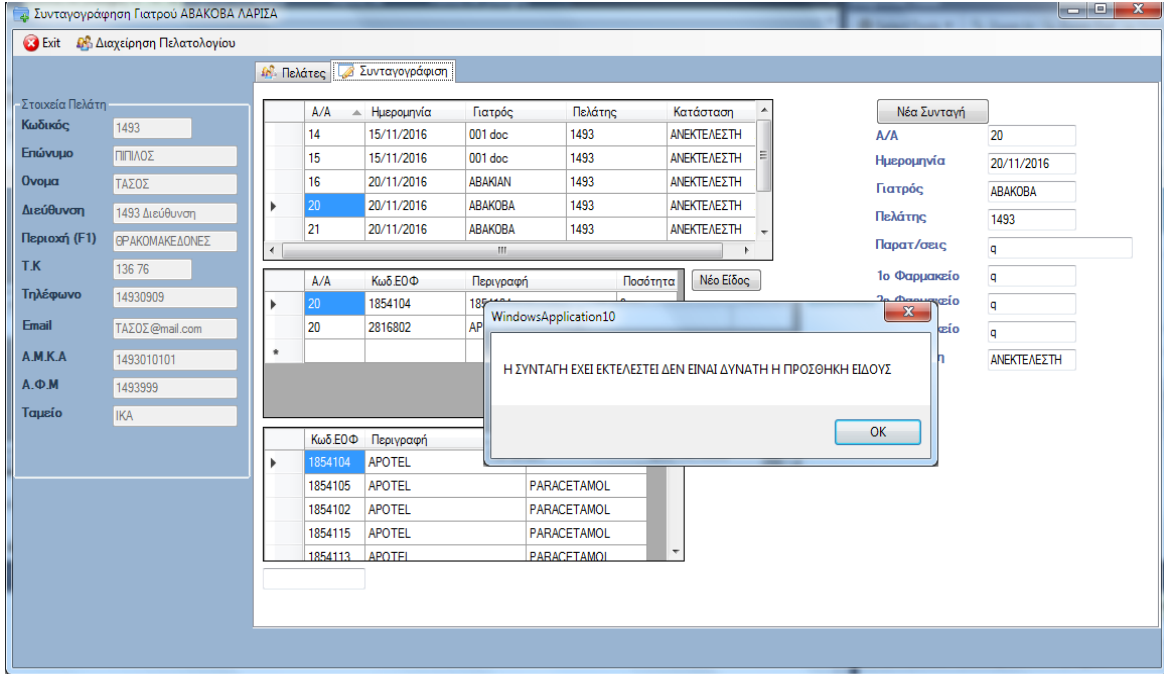
Στο δεύτερο είναι τα είδη κάθε συνταγής, δηλαδή ένα ιστορικό φαρμάκων που έχει συνταγογραφηθεί στον συγκεκριμένο πελάτη.

Στο τρίτο είναι ο κατάλογος φαρμάκων με την μεθοδολογία που έχει περιγραφεί στις προδιαγραφές.



Εικόνα 21

Ο γιατρός έχει την δυνατότητα να καταχωρήσει νέα συνταγή ή να προσθέσει φάρμακα σε συνταγές που δεν έχουν εκτελεστεί. Στην περίπτωση που προσπαθήσει να προσθέσει σε εκτελεσμένη συνταγή νέο φάρμακο η εφαρμογή τον ενημερώνει με μήνυμα.



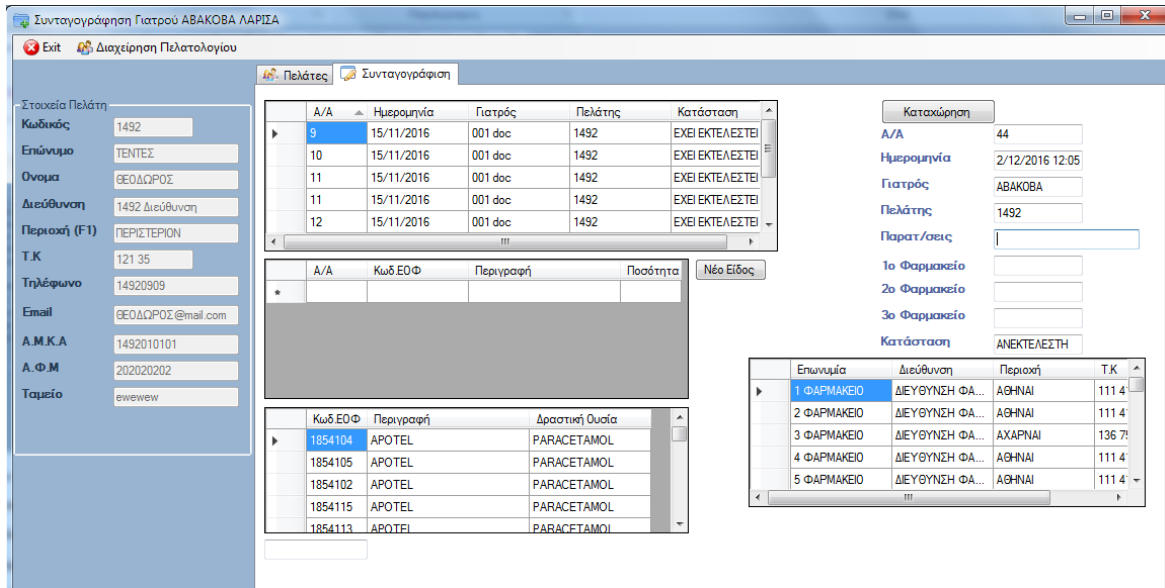
Εικόνα 22

Στην περίπτωση νέας συνταγής η εφαρμογή ανοίγει τον πίνακα αρίθμησης συνταγών και αυξάνει κατά 1 τον α/α.

Στην νέα συνταγή εμφανίζεται το πεδίο επιλογής από τον κατάλογο των φαρμακείων της περιοχής του πελάτη με βάση το Τ.Κ

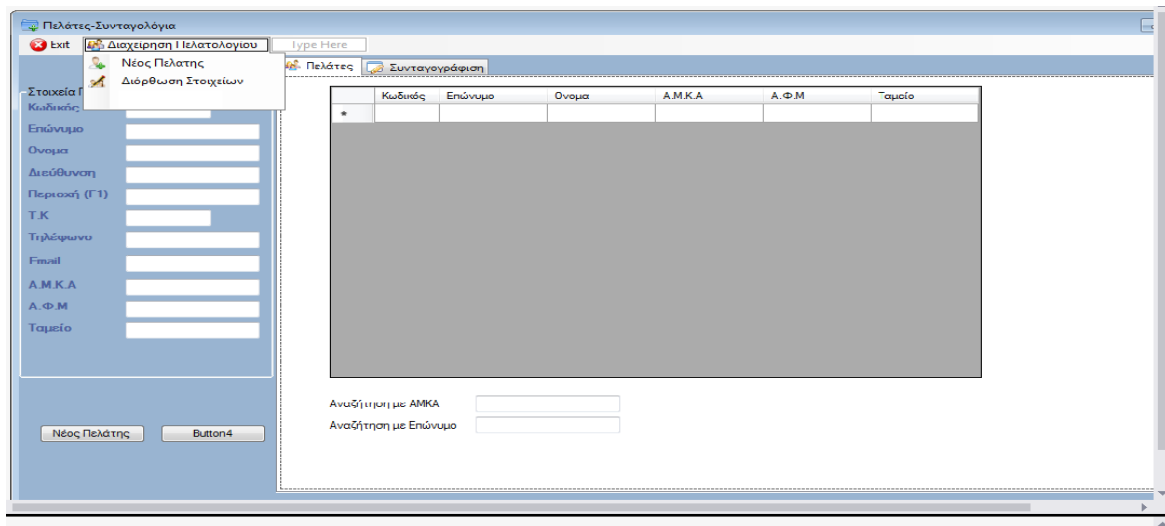
Εφαρμογή Κοινωνικού Περιεχομένου- Ηλεκτρονική Συνταγογράφηση Φαρμάκων

Σε κάθε πεδίο φαρμακείων 1,2,3 επιλέγει το εκάστοτε φαρμακείο που επιθυμεί ο πελάτης και ολοκληρώνει την συνταγογράφηση πατώντας το κουμπί “Καταχώρηση”



Εικόνα 23

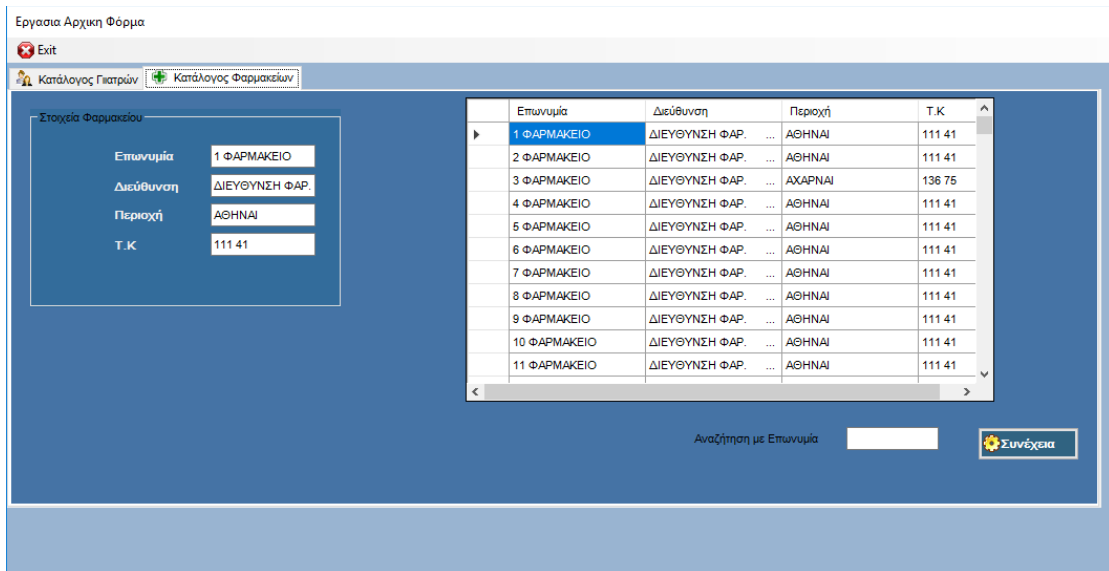
Στο Menu υπάρχει η επιλογή “Διαχείριση Πελατών” όπου εμφανίζονται 2 επιλογές για καταχώρηση νέου πελάτη ή διόρθωση υπαρχων πελάτη. Στο πεδίο που βρίσκεται από κάτω συμπληρώνει ή διορθώνει τα στοιχεία πατάει στο αντίστοιχο κουμπί που βρίσκεται από κάτω του πεδίου.



Εικόνα 24

30.3 Εκτέλεση Συνταγών

Αρχικά γίνεται στη πρώτη φόρμα η επιλογή φαρμακείου, όπως και στους γιατρούς και εδώ μας εμφανίζει πίνακα με όλα τα φαρμακεία που υπάρχουν στην βάση δεδομένων, καθώς μπορούμε να κάνουμε αναζήτηση του φαρμακείου με βάση την επωνυμία από το αντίστοιχο πεδίο. Επιλέγουμε το φαρμακείο και πατάμε το κουμπί “Συνέχεια”



Εικόνα 25

Με το άνοιγμα της φόρμας εμφανίζονται όλοι οι πελάτες που είναι καταχωρημένοι στην βάση, και αναζητά τα στοιχεία του πελάτη που θέλει να εκτελέσει την συνταγή, αφού επιλέξει τον εκάστοτε πελάτη πάει στο TAB “Συνταγογράφηση”

Είναι δυνατή η αναζήτηση πελάτη είτε με ΑΜΚΑ είτε με το Επώνυμο Πελάτη στα αντίστοιχα πεδία.

Εφαρμογή Κοινωνικού Περιεχομένου- Ηλεκτρονική Συνταγογράφηση Φαρμάκων

Εκτέλεση Συνταγών

Exit

Κωδικός: 1492

Επώνυμο: ΤΕΝΤΕΣ

Όνομα: ΘΕΟΔΩΡΟΣ

Διεύθυνση: 1492 Διεύθυνση

Περιοχή: ΑΧΑΡΝΑΙ

T.K: 136 75

Τηλέφωνο: 14920909

Email: ΘΕΟΔΩΡΟΣ@mail.com

A.M.K.A: 1492010101

A.Φ.Μ: 202020202

Ταμείο: ewewew

Κωδικός	Επώνυμο	Όνομα	A.M.K.A	Α.Φ.Μ	Ταμείο
1492	ΤΕΝΤΕΣ	ΘΕΟΔΩΡΟΣ	1492010101	202020202	ewewew
1493	ΠΙΠΛΟΣ	ΤΑΣΟΣ	1493010101	1493999	ΙΚΑ
1494	ΑΡΑΠΟΠΑΝΝΗΣ	ΝΙΚΟΣ	1494010101	1494999	ΙΚΑ
1495	ΣΑΡΡΗ	ΑΝΝΑ	1495010101	1495999	ΙΚΑ
1496	ΚΑΒΒΑΔΙΑ	ΕΛΕΝΗ	1496010101	1496999	ΙΚΑ
1497	ΓΡΙΒΕΛΙΑ	ΕΥΓΥΧΙΑ	1497010101	1497999	ΙΚΑ
1498	ΠΑΝΑΓΕΝΑΣ	ΚΩΝΣΤΑΝΤΙΝΟΣ	1498010101	1498999	ΙΚΑ
1499	ΕΥΑΓΓΕΛΑΤΟΣ	ΤΙΜΟΘΕΟΣ	1499010101	1499999	ΙΚΑ
1500	ΚΛΟΥΚΙΝΑ	ΒΑΣΙΛΙΚΗ	1500010101	1500999	ΙΚΑ
1501	ΜΧΑΛΑΚΗ	ΓΕΩΡΓΙΑ	1501010101	1501999	ΙΚΑ
1502	ΑΝΤΥΠΙΑ	ΑΜΑΛΙΑ	1502010101	1502999	ΙΚΑ

Αναζήτηση με ΑΜΚΑ:

Αναζήτηση με Επώνυμο:

Εικόνα 26

Στο στο TAB “Συνταγογράφηση” βλέπουμε όλες τις συνταγές του πελάτη που έχει επιλεγθεί και από εκεί μπορεί να εκτελέσει μόνο μια συνταγή η οποία είναι ανεκτέλεστη πατώντας το κουμπί “Εκτέλεση”.

Εκτέλεση Συνταγών

Exit

Κωδικός: 1492

Επώνυμο: ΤΕΝΤΕΣ

Όνομα: ΘΕΟΔΩΡΟΣ

Διεύθυνση: 1492 Διεύθυνση

Περιοχή: ΑΧΑΡΝΑΙ

T.K: 136 75

Τηλέφωνο: 14920909

Email: ΘΕΟΔΩΡΟΣ@mail.com

A.M.K.A: 1492010101

A.Φ.Μ: 202020202

Ταμείο: ewewew

A/A	Ημερομηνία	Γιατρός	Πελάτης	Κατάσταση
1	9/12/2016	ΑΒΑΚΟΒΑ	1492	ΕΧΕΙ ΕΚΤΕΛΕΣΤΕΙ...
3	10/12/2016	ΑΒΑΤΑΓΓΕΛΟ	1492	ΑΝΕΚΤΕΛΕΣΤΗ ...

Εκτέλεση

A/A: 1

Ημερομηνία: 9/12/2016

Γιατρός: ΑΒΑΚΟΒΑ

Πελάτης: 1492

Παρατ/σεις:

1ο Φαρμακείο: 1 ΦΑΡΜΑΚΕΙ

2ο Φαρμακείο: 2 ΦΑΡΜΑΚΕΙ

3ο Φαρμακείο: 3 ΦΑΡΜΑΚΕΙ

Κατάσταση: ΕΧΕΙ ΕΚΤΕΛΕΣΤΕΙ

A/A	Κωδ. ΕΟΦ	Περιγραφή	Ποσότητα
1	1854104	ΑΡΟΤΕΛ	2
1	1854105	ΑΡΟΤΕΛ	2

Κωδ.ΕΟΦ	Περιγραφή	Δραστική Ουσία
1854104	ΑΡΟΤΕΛ	PARACETAMOL
1854105	ΑΡΟΤΕΛ	PARACETAMOL
1854102	ΑΡΟΤΕΛ	PARACETAMOL
1854115	ΑΡΟΤΕΛ	PARACETAMOL
1854113	ΑΡΟΤΕΙ	PARACETAMOI

Εικόνα 27

31 Διαδικτυακή Εφαρμογή για οποιαδήποτε συσκευή σε HTML – ASP.NET

Οι μεταφορά της εφαρμογής έγινε στο web στον iis-8 αφού παραμετροποιήθηκε και μορφοποιήθηκε για asp pages σύμφωνα με τις οδηγίες της Microsoft

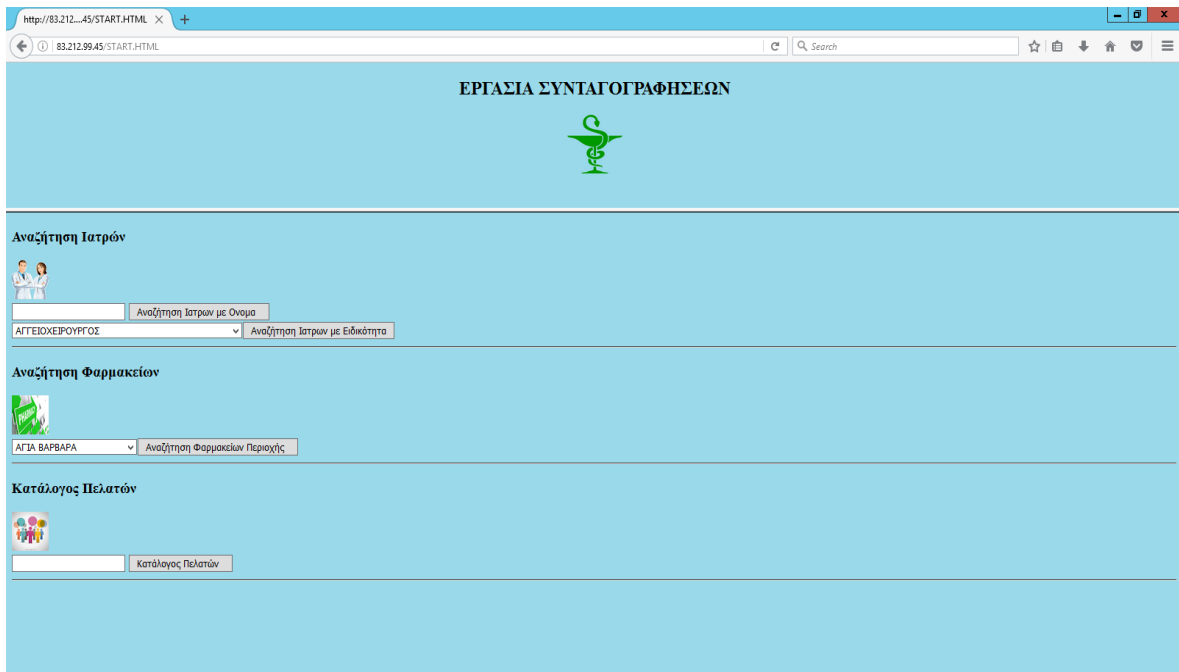
<https://www.iis.net/learn/application-frameworks/running-classic-asp-applications-on-iis-7-and-iis-8>

Η μορφοποίηση για html pages έγινε ως unicode και για asp pages ως utf-8

83.212.99.45/start.html

31.1 ΑΡΧΙΚΗ WEB PAGE START.HTML

Στην αρχική φόρμα έχουμε τρία πεδία το πρώτο για την αναζήτηση των ιατρών, αναζήτηση φαρμακείων και κατάλογος των πελατών.

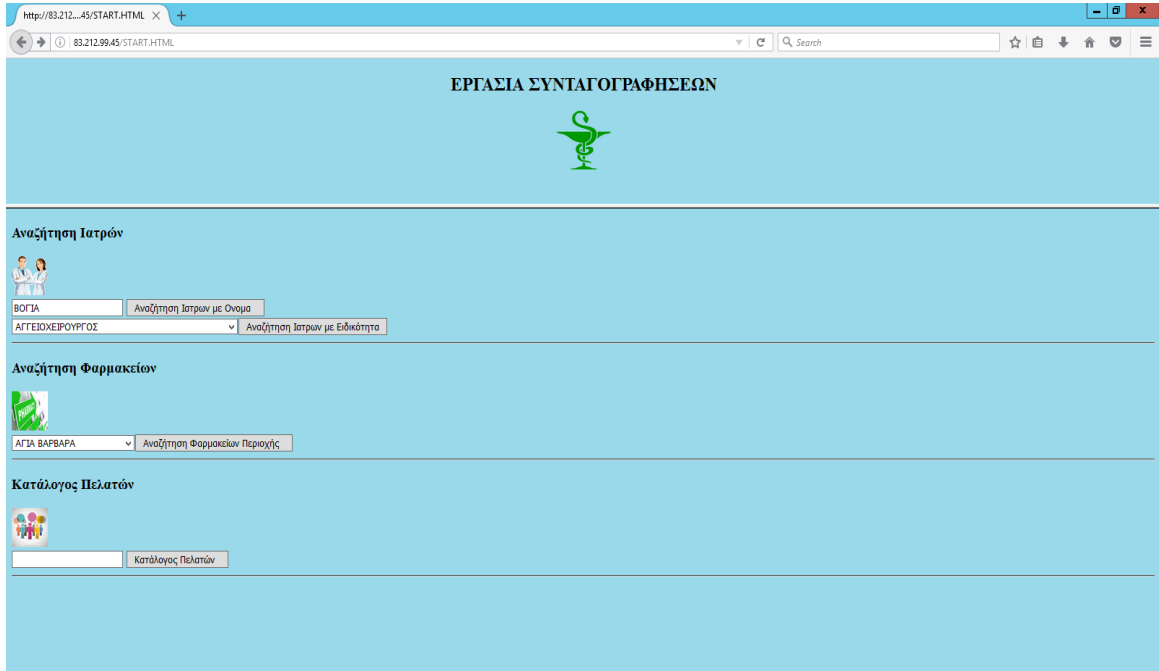


Εικόνα 28

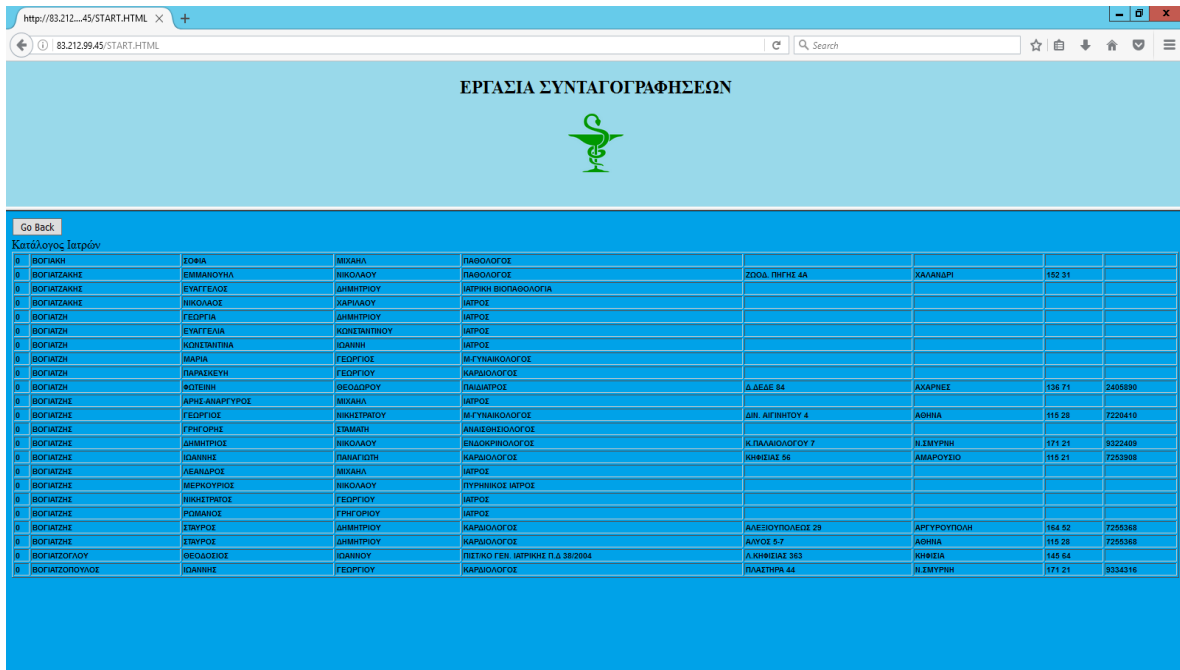
31.2 ΑΝΑΖΗΤΗΣΗ ΓΙΑΤΡΩΝ

Εδώ μπορούμε να κάνουμε αναζήτηση του ιατρού είτε με βάση το όνομα είτε βάση της ειδικότητας

Στην αναζήτηση με βάση το όνομα, πληκτρολογούμε το όνομα του ιατρού ή μέρος του ονόματός του και πατάμε στο πλαϊνό κουμπί. Έπειτα εμφανίζεται σε νέο παράθυρο με όλους τους ιατρούς αυτού του ονόματος.



Εικόνα 29



Εικόνα 30

Στην αναζήτηση με βάση την ειδικότητα, πατάμε πάνω στο αντίστοιχο πεδίο και μας εμφανίζεται η λίστα με όλες τις ειδικότητες των ιατρών, επιλέγουμε και πατάμε στο πλαϊνό κουμπί. Έπειτα εμφανίζεται σε νέο παράθυρο με όλους τους ιατρούς αυτής της ειδικότητας.

Κατάλογος Ιατρών	ΕΠΙΧΕΙΡΗΣΙΑΚΟ	ΕΠΙΧΕΙΡΗΣΙΑΚΟ	ΕΠΙΧΕΙΡΗΣΙΑΚΟ	ΕΠΙΧΕΙΡΗΣΙΑΚΟ	ΕΠΙΧΕΙΡΗΣΙΑΚΟ	ΕΠΙΧΕΙΡΗΣΙΑΚΟ
1] ΑΓΓΕΛΑΚΑΣ	ΙΩΑΝΝΗΣ	ΑΧΙΛΛΕΑΣ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΕΥΤΥΧΙΔΟΥ 18	ΑΘΗΝΑ	116 34 701513
2] ΑΓΓΕΛΑΚΑΣ	ΚΑΤΕΡΙΝΗ	ΑΘΑΝΑΣΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
3] ΑΘΑΝΑΣΙΑΔΗΣ	ΔΗΜΗΤΡΙΟΣ	ΓΕΩΡΓΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
4] ΕΛΕΥΘΕΡΙΟΥ	ΓΕΩΡΓΙΟΣ	ΙΩΑΝΝΗ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΚΗΦΙΣΙΑΣ 18	ΑΘΗΝΑ	115 26 7474154
5] ΚΑΠΕΡΩΝΗΣ	ΝΗΛΙΑ	ΑΘΑΝΑΣΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
6] ΜΠΕΛΛΟΣ	ΙΩΑΝΝΗΣ	ΚΩΝΣΤΑΝΤΙΝΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΥΜΗΤΤΟΥ 112	ΑΘΗΝΑ	118 34 7524005
7] ΜΠΕΣΙΑΣ	ΝΙΚΟΛΑΟΣ	ΚΩΝΣΤΑΝΤΙΝΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
8] ΠΕΡΕΤΗΣ	ΚΩΝΣΤΑΝΤΙΝΟΣ	ΓΕΩΡΓΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
9] ΣΕΚΑΣ	ΝΙΝΩΝ	ΝΙΚΟΛΑΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
10] ΣΙΑΦΑΚΑΣ	ΑΘΑΝΑΣΙΟΣ	ΣΕΛΩΚΟΝΤΑ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΚΟΥΣΙΑΝΟΦΕΚΥ 12	ΑΘΗΝΑ	115 25 6993242
11] ΣΙΓΑΛΑ	ΦΡΑΓΚΙΣΚΑ	ΚΩΝΣΤΑΝΤΙΝΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
12] ΣΚΟΥΡΤΗΣ	ΓΕΩΡΓΙΟΣ	ΕΥΘΥΜΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΕΠΡΟΓΙΑΝΝΗ 13	ΖΩΓΡΑΦΟΥ	157 71 210 7715117
13] ΒΕΡΥΚΟΚΟΣ	ΧΡΗΣΤΟΣ	ΜΙΧΑΗΛ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
14] ΒΟΥΡΛΙΩΤΑΚΗΣ	ΓΕΩΡΓΙΟΣ	ΔΗΜΗΤΡΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΒΑΣ. ΣΟΦΙΑΣ 122	ΑΘΗΝΑ	115 26 217708002
15] ΜΑΡΚΑΤΗΣ	ΦΩΤΙΟΣ	ΑΛΕΞΑΝΔΡΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
16] ΜΑΡΤΙΝΑΚΗΣ	ΒΑΣΙΛΕΙΟΣ	ΓΕΩΡΓΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΒΑΣ. ΣΟΦΙΑΣ 116	ΑΘΗΝΑ	11527 7708559
17] ΜΑΡΜΑΡΟΥ	ΑΛΕΞΑΝΔΡΟΣ	ΚΩΝΣΤΑΝΤΙΝΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΚΑΡΦΕΔΟΥ 41	ΑΘΗΝΑ	2130628757
18] ΜΑΥΡΑΔΟΥ	ΑΛΕΞΑΝΔΡΟΣ	ΚΩΝΣΤΑΝΤΙΝΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	Α. ΠΕΠΤΕΛΗΣ 22	ΧΑΛΚΙΔΙΚΗ	192 34
19] ΜΕΛΑΣ	ΜΙΧΑΗΛ	ΔΗΜΗΤΡΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
20] ΜΟΥΚΑΚΑΚΗΣ	ΚΩΝΣΤΑΝΤΙΝΟΣ	ΓΕΩΡΓΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
21] ΜΟΥΣΤΟΓΙΑΝΝΗΣ	ΠΕΤΡΟΣ	ΙΩΑΝΝΗ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΑΓ. ΝΙΚΟΛΑΟΥ 42	ΛΑΡΙΣΑ	131 23 2623848
22] ΜΠΑΖΟΣ	ΓΕΡΑΣΙΜΟΣ	ΒΑΣΙΛΕΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΔΗΜΑΡΧΕΙΟΥ 8	ΑΙΓΑΛΕΩ	122 42 2114083717
23] ΜΠΑΚΟΓΙΑΝΝΗΣ	ΚΩΝΣΤΑΝΤΙΝΟΣ	ΣΤΕΦΑΝΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	Α. ΚΗΦΙΣΙΑΣ 39	ΑΘΗΝΑ	115 23 2114019800
24] ΜΠΑΚΟΓΙΑΝΝΗΣ	ΧΡΗΣΤΟΣ	ΝΙΚΟΛΑΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
25] ΡΟΥΚΙΑΣ	ΡΩΜΑΝΟΣ	ΔΩΝΤΙΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΚΟΝΙΟΥ ΠΑΛΑΙΟΚΟΥ 4	Ν. ΕΣΣΥΡΗΝ	171 21 8125872
26] ΣΑΛΑΒΕΡΟΣ	ΑΠΟΣΤΟΛΟΣ	ΔΑΜΠΡΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΠΛ. ΒΑΣΟΣ ΚΑΤΡΑΚΗ 8	ΓΥΦΘΑΔΑ	168 75
27] ΠΟΡΤΙΝΟΣ	ΑΘΑΝΑΣΙΟΣ	ΝΙΚΟΛΑΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	Α. ΑΥΡΟΥ 150	ΠΑΙΔΙΑ	190 02 9546096
28] ΠΟΥΛΑΝΤΣΑΣ	ΠΕΤΡΟΣ	ΚΩΝΣΤΑΝΤΙΝΟΥ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
29] ΠΡΟΠΟΝΑΔΑΣ	ΝΙΚΟΛΑΟΣ	ΠΑΡΗ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ			
30] ΡΗΓΟΠΟΥΛΟΣ	ΧΡΗΣΤΟΣ	ΙΩΑΝΝΟΥ-ΕΥΑΓΓΕΛΑ	ΑΓΓΕΙΟΧΕΙΡΟΥΡΓΟΣ	ΥΜΗΤΤΟΥ 4-6	ΧΟΛΑΡΓΟΣ	155 61 8528725

Εικόνα 31

31.3 ΑΝΑΖΗΤΗΣΗ ΦΑΡΜΑΚΕΙΩΝ ΜΕ ΠΕΡΙΟΧΗ

Εδώ κάνουμε αναζήτηση φαρμακείου βάση την περιοχή, πατάμε πάνω στο αντίστοιχο πεδίο και μας εμφανίζεται η λίστα με όλες τις περιοχές, επιλέγουμε και πατάμε στο πλαϊνό κουμπί. Έπειτα εμφανίζεται σε νέο παράθυρο όλα τα φαρμακεία της επιλεγμένης περιοχής.

Κατάλογος Φαρμακείων Περιοχής	ΕΠΙΧΕΙΡΗΣΙΑΚΟ	ΕΠΙΧΕΙΡΗΣΙΑΚΟ	ΕΠΙΧΕΙΡΗΣΙΑΚΟ
1] 1 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 1	ΑΘΗΝΑΙ	111 41
2] 2 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 2	ΑΘΗΝΑΙ	111 41
4] 4 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 4	ΑΘΗΝΑΙ	111 41
5] 5 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 5	ΑΘΗΝΑΙ	111 41
6] 6 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 6	ΑΘΗΝΑΙ	111 41
7] 7 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 7	ΑΘΗΝΑΙ	111 41
8] 8 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 8	ΑΘΗΝΑΙ	111 41
9] 9 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 9	ΑΘΗΝΑΙ	111 41
10] 10 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 10	ΑΘΗΝΑΙ	111 41
11] 11 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 11	ΑΘΗΝΑΙ	111 41
12] 12 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 12	ΑΘΗΝΑΙ	111 41
13] 13 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 13	ΑΘΗΝΑΙ	111 41
14] 14 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 14	ΑΘΗΝΑΙ	111 41
15] 15 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 15	ΑΘΗΝΑΙ	111 41
16] 16 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 16	ΑΘΗΝΑΙ	111 41
17] 17 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 17	ΑΘΗΝΑΙ	111 41
18] 18 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 18	ΑΘΗΝΑΙ	111 41
19] 19 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 19	ΑΘΗΝΑΙ	111 41
20] 20 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 20	ΑΘΗΝΑΙ	111 41
21] 21 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 21	ΑΘΗΝΑΙ	111 41
22] 22 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 22	ΑΘΗΝΑΙ	111 41
23] 23 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 23	ΑΘΗΝΑΙ	111 41
24] 24 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 24	ΑΘΗΝΑΙ	111 41
25] 25 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 25	ΑΘΗΝΑΙ	111 41
26] 26 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 26	ΑΘΗΝΑΙ	111 41
27] 27 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 27	ΑΘΗΝΑΙ	111 41
28] 28 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 28	ΑΘΗΝΑΙ	111 41
29] 29 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 29	ΑΘΗΝΑΙ	111 41
30] 30 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 30	ΑΘΗΝΑΙ	111 41
31] 31 ΦΑΡΜΑΚΕΙΟ	ΔΙΕΥΘΥΝΣΗ ΦΑΡ. 31	ΑΘΗΝΑΙ	111 41

Εικόνα 32

31.4 ΑΝΑΖΗΤΗΣΗ ΠΕΛΑΤΩΝ ΚΑΙ ΣΥΝΤΑΓΩΝ

Πατώντας το κουμπί “Κατάλογος Πελατών” εμφανίζεται νέο παράθυρο με όλους τους πελάτες

ΚΑΤΑΛΟΓΟΣ ΠΕΛΑΤΩΝ	Κωδικός Πελάτη	ΑΝΑΖΗΤΗΣΗ ΣΥΝΤΑΓΩΝ										
2835	ΑΒΑΝΑΣΟΠΟΥΛΟΣ	ΓΙΑΝΝΗΣ	2935 Διοίκηση	ΝΕΑ ΣΜΥΡΝΗ	171 21	29350909	ΓΙΑΝΝΗΣ@gmail.com	2935010101	2935099	ΙΚΑ	2016-01-01	3844
3118	ΑΒΑΝΑΣΟΠΟΥΛΟΣ	ΕΛΠΙΔΑ	3118 Διοίκηση	ΓΛΥΦΑΣ (ΓΛΥΦΑΔΑ)	165 52	31180999	ΕΛΠΙΔΑ@gmail.com	3118010101	3118999	ΙΚΑ	2016-01-01	3827
3127	ΑΒΑΝΑΣΙΔΗΣ	ΛΑΖΑΡΟΣ	3127 Διοίκηση	ΓΛΥΦΑΣ (ΓΛΥΦΑΔΑ)	165 52	31270999	ΛΑΖΑΡΟΣ@gmail.com	3127010101	3127999	ΙΚΑ	2016-01-01	3836
6	ΑΒΑΝΑΣΙΔΟΥ	ΚΑΤΕΡΙΝΑ	6 Διοίκηση	ΑΓΙΟΣ ΔΗΜΗΤΡΙΟΣ	173 43	60999	ΚΑΤΕΡΙΝΑ@gmail.com	6010101	6999	ΙΚΑ	2016-01-01	1347
7	ΑΒΑΝΑΣΙΔΟΥ	ΖΑΝΕΤΑ	7 Διοίκηση	ΑΓΙΟΣ ΔΗΜΗΤΡΙΟΣ	173 43	70999	ΖΑΝΕΤΑ@gmail.com	7010101	7999	ΙΚΑ	2016-01-01	1348
1030	ΑΒΑΝΑΣΙΔΗΣ	ΣΤΑΘΗΣ	1030 Διοίκηση	ΑΛΙΜΟΣ	174 55	1030999	ΣΤΑΘΗΣ@gmail.com	1030010101	1030999	ΙΚΑ	2016-01-01	3988
1116	ΑΒΑΝΑΣΙΔΗΣ	ΚΥΡΙΑΚΟΣ	1116 Διοίκηση	ΑΛΙΜΟΣ	174 55	1116999	ΚΥΡΙΑΚΟΣ@gmail.com	1116010101	1116999	ΙΚΑ	2016-01-01	3174
2208	ΑΘΗΝΑΙΟΥ	ΓΙΑΣΙΝΗ	2208 Διοίκηση	ΗΛΙΟΥΠΟΛΙΣ	163 42	22080999	ΓΙΑΣΙΝΗ@gmail.com	2208010101	2208999	ΙΚΑ	2016-01-01	2144
2210	ΑΒΑΝΑΣΟΠΟΥΛΟΣ	ΧΑΡΑΛΑΜΠΟΣ	2210 Διοίκηση	ΗΛΙΟΥΠΟΛΙΣ	163 42	22109999	ΧΑΡΑΛΑΜΠΟΣ@gmail.com	2210010101	2210999	ΙΚΑ	2016-01-01	2246
611	ΑΒΑΝΑΣΙΔΗΣ	ΝΙΚΗΤΑΣ	611 Διοίκηση	ΑΘΗΝΑΙ	111 41	6110999	ΝΙΚΗΤΑΣ@gmail.com	611010101	611999	ΙΚΑ	2016-01-01	1979
661	ΑΒΑΝΑΣΟΠΟΥΛΟΥ	ΜΑΡΙΑ	661 Διοίκηση	ΑΘΗΝΑΙ	111 41	6610999	ΜΑΡΙΑ@gmail.com	661010101	661999	ΙΚΑ	2016-01-01	2035
709	ΑΒΑΝΑΣΟΠΟΥΛΟΥ	ΑΔΑΜΑΝΤΙΑ	709 Διοίκηση	ΑΘΗΝΑΙ	111 41	7090999	ΑΔΑΜΑΝΤΙΑ@gmail.com	709010101	709999	ΙΚΑ	2016-01-01	2768
1343	ΑΒΑΝΑΣΟΠΟΥΛΟΣ	ΒΑΓΓΕΛΗΣ	1343 Διοίκηση	ΑΧΑΡΝΑΙ	136 75	13430999	ΒΑΓΓΕΛΗΣ@gmail.com	1343010101	1343999	ΙΚΑ	2016-01-01	3401
1346	ΑΒΑΝΑΣΙΔΗΣ	ΠΟΥΛΚΑΡΤΟΣ	1346 Διοίκηση	ΑΧΑΡΝΑΙ	136 75	13460999	ΠΟΥΛΚΑΡΤΟΣ@gmail.com	1346010101	1346999	ΙΚΑ	2016-01-01	3404
2729	ΑΒΑΝΑΣΟΠΟΥΛΟΥ	ΜΠΛΑΝΚΑ	2729 Διοίκηση	ΝΕΑ ΣΜΥΡΝΗ	171 21	27290999	ΜΠΛΑΝΚΑ@gmail.com	2729010101	2729999	ΙΚΑ	2016-01-01	3429
3344	ΑΒΑΝΑΣΙΟΥ	ΧΡΥΣΟΥΛΑ	3344 Διοίκηση	ΓΛΥΦΑΣ (ΓΛΥΦΑΔΑ)	165 52	33440999	ΧΡΥΣΟΥΛΑ@gmail.com	3344010101	3344999	ΙΚΑ	2016-01-01	4652
1709	ΑΒΑΝΑΣΟΠΟΥΛΟΥ	ΜΑΡΙΑ	1709 Διοίκηση	ΒΥΡΩΝ	162 31	17090999	ΜΑΡΙΑ@gmail.com	1709010101	1709999	ΙΚΑ	2016-01-01	279
1982	ΑΒΑΝΑΣΟΠΟΥΛΟΥ	ΝΑΤΑΣΙΑ	1982 Διοίκηση	ΓΑΛΛΙΣΙΟΝ	111 46	19820999	ΝΑΤΑΣΙΑ@gmail.com	1982010101	1982999	ΙΚΑ	2016-01-01	589
245	ΑΒΑΝΑΣΟΠΟΥΛΟΥ	ΒΑΣΙΛΙΚΗ	245 Διοίκηση	ΑΘΗΝΑΙ	111 41	2450999	ΒΑΣΙΛΙΚΗ@gmail.com	245010101	245999	ΙΚΑ	2016-01-01	1584

Εικόνα 33

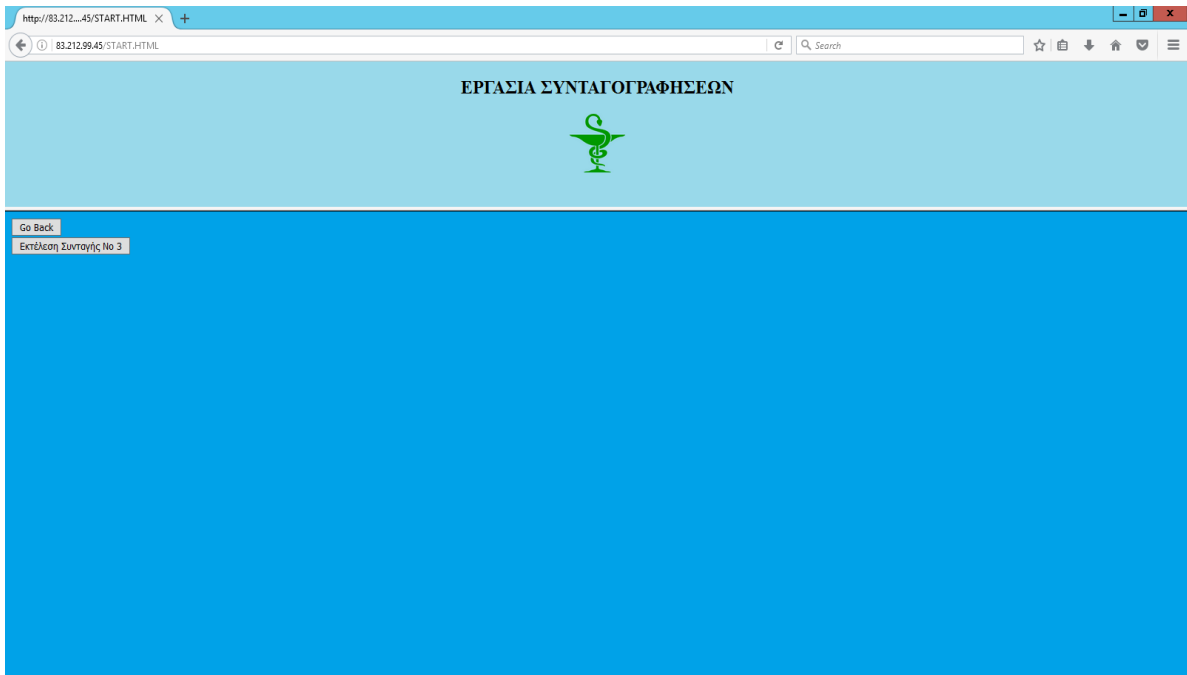
Εδώ μπορούμε να επιλέξουμε τον πελάτη με βάση το κωδικό του. Πληκτρολογούμε το κωδικό του πελάτη και πατάμε στο πλαϊνό κουμπί. Έπειτα εμφανίζεται σε νέο παράθυρο όλα τα στοιχεία του πελάτη και το ιστορικό των συνταγών του.

ΠΕΛΑΤΗΣ 1492	Κωδικός Πελάτη	ΑΝΑΖΗΤΗΣΗ ΣΥΝΤΑΓΩΝ										
1492	ΠΕΤΕΣ	ΘΕΟΔΩΡΟΣ	1492 Διοίκηση	ΑΧΑΡΝΑΙ	136 75	14920999	ΘΕΟΔΩΡΟΣ@gmail.com	1492010101	20202092	ewewew	2016-01-01	34
ΣΥΝΤΑΓΕΣ												
1	ΙΒΑΚΟΒΑ	2016-12-09	1492	NEW REM	1 ΦΑΡΜΑΚΕΙ	2 ΦΑΡΜΑΚΕΙ	3 ΦΑΡΜΑΚΕΙ	ΕΚΕΙ ΕΚΤΕΛΕΙΤΕΙ	1854104	ΑΡΤΕΤΕΛ	2	
1	ΙΒΑΚΟΒΑ	2016-12-09	1492	NEW REM	1 ΦΑΡΜΑΚΕΙ	2 ΦΑΡΜΑΚΕΙ	3 ΦΑΡΜΑΚΕΙ	ΕΚΕΙ ΕΚΤΕΛΕΙΤΕΙ	1854105	ΑΡΤΕΤΕΛ	2	
3	ΙΒΑΡΑΓΓΕΛΟ	2016-12-10	1492	NEW REM	1 ΦΑΡΜΑΚΕΙ	2 ΦΑΡΜΑΚΕΙ	3 ΦΑΡΜΑΚΕΙ	ΕΚΕΙ ΕΚΤΕΛΕΙΤΕΙ	2379902	ΑΡΤΕΤΕΛ EXTRA	1	

Εκτύπωση Συνταγής Με Α/Α

Εικόνα 34

Για την εκτέλεση ανεκτέλεστης συνταγής πληκτρολογούμε το A/A της συνταγής και πατάμε στο πλαϊνό κουμπί. Έπειτα εμφανίζεται σε νέο παράθυρο δυο κουμπιά τον ένα για επιστροφή πίσω στον πελάτη και το άλλο για τη επιβεβαίωση της εκτέλεσης της συνταγής πατώντας το, μας εμφανίζεται σχετικό μήνυμα ότι ολοκληρώθηκε.



Εικόνα 35

32 ΠΑΡΑΡΤΗΜΑ Ι – ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ (C#)

32.1 Αρχική φόρμα – MainPage.xaml.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;
using Windows.UI.Popups;
using SQLitePCL;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=402352&clcid=0x409

namespace e_priscription_v02._5
{
    /// <summary>
    /// An empty page that can be used on its own or navigated to within a Frame.
    /// </summary>
    public sealed partial class MainPage : Page
    {
        String radiosel;
        public MainPage()
        {
            this.InitializeComponent();
            this.NavigationCacheMode = NavigationCacheMode.Required;
        }

        public SQLitePCL.SQLiteConnection dbcoonection = new SQLiteConnection("NeaErgasia.db");

        private void tbLogin_GotFocus(object sender, RoutedEventArgs e)
        {
            tbLogin.Text = "";
        }

        private void tbLogin_TextChanged(object sender, TextChangedEventArgs e)
        {
            long a;
            if (!long.TryParse(tbLogin.Text, out a))
            {
                // If not int clear textbox text or Undo() last operation
                tbLogin.Text = "";
            }
        }
    }
}
```

```

    }
}

private void bLogin_Click(object sender, RoutedEventArgs e)
{
    switch (radiosel)
    {
        case "Doctor":
            String sSQL = @"SELECT [DoctorId] from iatroi where [DoctorId] = " + tbLogin.Text;
            ISQLiteStatement dbstate = dbcoonection.Prepare(sSQL);
            while (dbstate.Step() == SQLiteResult.ROW)
            {
                var itemId = tbLogin.Text;
                this.Frame.Navigate(typeof(DoctorPage), itemId);
            }
            break;

        case "Drugstore":
            sSQL = @"SELECT [Store_Id] from dragstores where [Store_Id] = " + tbLogin.Text;
            dbstate = dbcoonection.Prepare(sSQL);
            while (dbstate.Step() == SQLiteResult.ROW)
            {
                var itemId = tbLogin.Text;
                this.Frame.Navigate(typeof(DrugstorePage), itemId);
            }
            break;
    }
}

private void radioDoctor_Checked(object sender, RoutedEventArgs e)
{
    radiosel = "Doctor";
}

private void radioDrugstore_Checked(object sender, RoutedEventArgs e)
{
    radiosel = "Drugstore";
}
}
}

```

32.2 Φόρμα Ιατρού – DoctorPage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;
using Windows.UI.Popups;
using SQLitePCL;

```

// The Blank Page item template is documented at <http://go.microsoft.com/fwlink/?LinkId=234238>

```

namespace e_priscription_v02._5
{
    /// <summary>
    /// An empty page that can be used on its own or navigated to within a Frame.
    /// </summary>
    public sealed partial class DoctorPage : Page
    {
        public DoctorPage()
        {
            this.InitializeComponent();
            this.NavigationCacheMode = NavigationCacheMode.Required;
        }

        public SQLitePCL.SQLiteConnection dbcoonection = new SQLiteConnection("NeaErgasia.db");
        protected override void OnNavigatedTo(NavigationEventArgs e)
        {
            base.OnNavigatedTo(e);
            string id = e.Parameter.ToString();
            tbId.Text = id;
        }

        private void textBox_TextChanged(object sender, TextChangedEventArgs e)
        {
        }

        private void bLogout_Click(object sender, RoutedEventArgs e)
        {
            this.Frame.Navigate(typeof(MainPage));
            tbSuccess.Text = "";
            tbAMKA.Text = "";
        }

        private void Grid_Loaded(object sender, RoutedEventArgs e)
        {
            String sSQL = @"SELECT [DoctorId],[LastNmae],[FirstName],[Specialty],[Phone] from iatroi where [DoctorId] = " +
            tbId.Text;
            ISQLiteStatement dbstate = dbcoonection.Prepare(sSQL);
            while (dbstate.Step() == SQLiteResult.ROW)
            {
                tbname.Text = dbstate["LastNmae"] as string + " " + dbstate["FirstName"] as string;
                tbSpeciality.Text = dbstate["Specialty"] as string;
                tbPhone.Text = dbstate["Phone"] as string;
            }

            var Items = new List<string>();
            sSQL = @"SELECT [EOF_CODE],[DESCRIPTION],[INCREMET] from DRAGS";
            dbstate = dbcoonection.Prepare(sSQL);
            while (dbstate.Step() == SQLiteResult.ROW)
            {
                Items.Add(dbstate["EOF_CODE"] + " " + dbstate["DESCRIPTION"] + " " + dbstate["INCREMET"]);
            }
            listDrugsDescription.ItemsSource = Items;
        }
    }
}

```

```

}

private void tbSearch_GotFocus(object sender, RoutedEventArgs e)
{
    tbAMKA.Text = "";
}

private void bSearch_Click(object sender, RoutedEventArgs e)
{
    var Items = new List<string>();
    String sSQL = @"SELECT
[Cust_Lname],[Cust_Fname],[Cust_AMKA],[Cust_Address],[Cust_Region],[Cust_ZIP],[Cust_Phone],[Cust_mail] from
customers where [Cust_AMKA] =" + tbAMKA.Text;
    ISQLiteStatement dbstate = dbcoonection.Prepare(sSQL);
    while (dbstate.Step() == SQLiteResult.ROW)

    {
        Items.Add("Όνοματεπώνυμο: "+dbstate["Cust_Lname"]+" "+dbstate["Cust_Fname"]+" Διεύθυνση:
"+dbstate["Cust_Address"]+" Πόλη: "+dbstate["Cust_Region"]+" Τ.Κ.: "+dbstate["Cust_ZIP"]+" Τηλέφωνο:
"+dbstate["Cust_Phone"]+" e-mail: "+dbstate["Cust_mail"]);
    }
    listcustomer.ItemsSource = Items;

    var Items1 = new List<string>();
    sSQL = @"SELECT [amka],[Drag],[Quant] from preskr where [amka] =" + tbAMKA.Text + "";
    dbstate = dbcoonection.Prepare(sSQL);
    while (dbstate.Step() == SQLiteResult.ROW)
    {
        Items1.Add(dbstate["Drag"] + " " + dbstate["Quant"]);
    }
    listDrugsPrscription.ItemsSource = Items1;
}

private void listcustomer_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
}

private void textBlock_Copy5_SelectionChanged(object sender, RoutedEventArgs e)
{
}

public void listDrugsDescription_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    tbSelDrug.Text = listDrugsDescription.SelectedItem.ToString();
}

private void textBlock_Copy4_SelectionChanged(object sender, RoutedEventArgs e)
{
}

private async void bAdd_Click(object sender, RoutedEventArgs e)
{
    if (tbAMKA.Text== "")
    {

```

```

        MessageBox msgbox = new MessageBox("Παρακαλώ συμπληρώστε το Α.Μ.Κ.Α. του πελάτη");
        await msgbox.ShowAsync();
    }
    else if (tbQuantity.Text == "") {
        MessageBox msgbox = new MessageBox("Παρακαλώ συμπληρώστε το ποσότητα του φαρμάκου");
        await msgbox.ShowAsync();
    }
    else
    {
        string sSQL = @"insert into preskr ([amka],[drag],[quant])
            values ('" + tbAMKA.Text + "','" + tbSelDrug.Text + "','" + tbQuantity.Text + "')";
        dbcoonection.Prepare(sSQL).Step();
        {
            var Items = new List<string>();
            String sSQL1 = @"SELECT [amka],[Drag],[Quant] from preskr where [amka] like '" + tbAMKA.Text + "%'";
            ISQLiteStatement dbstate = dbcoonection.Prepare(sSQL1);
            while (dbstate.Step() == SQLiteResult.ROW)
            {
                Items.Add(dbstate["Drag"] + " " + dbstate["Quant"]);
            }
            listDrugsPriscription.ItemsSource = Items;
        }
        tbSuccess.Text = "Καταχώρηση επιτυχής!";
    }
}

}

private void listDrugsPriscription_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
}

private void bDelete_Click(object sender, RoutedEventArgs e)
{
    String sSQL = @"delete from preskr where [amka] =" + tbAMKA.Text;
    ISQLiteStatement dbstate = dbcoonection.Prepare(sSQL);
    dbstate.Step();
    tbSuccess.Text = "Διαγραφή επιτυχής!";
}

private void tbAMKA_TextChanged(object sender, TextChangedEventArgs e)
{
}
}
}
}

```

32.3 Φόρμα Φαρμακοποιού – DrugstorePage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;

```


Εφαρμογή Κοινωνικού Περιεχομένου- Ηλεκτρονική Συνταγογράφηση Φαρμάκων

```
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;
using Windows.UI.Popups;
using SQLitePCL;

// The Blank Page item template is documented at http://go.microsoft.com/fwlink/?LinkId=234238

namespace e_priscription_v02._5
{
    /// <summary>
    /// An empty page that can be used on its own or navigated to within a Frame.
    /// </summary>
    public sealed partial class DrugstorePage : Page
    {
        public DrugstorePage()
        {
            this.InitializeComponent();
            this.NavigationCacheMode = NavigationCacheMode.Required;
        }
        public SQLitePCL.SQLiteConnection dbcoonection = new SQLiteConnection("NeaErgasia.db");

        protected override void OnNavigatedTo(NavigationEventArgs e)
        {
            base.OnNavigatedTo(e);
            string id = e.Parameter.ToString();
            tbId.Text = id;
        }

        private void bLogout_Click(object sender, RoutedEventArgs e)
        {
            this.Frame.Navigate(typeof(MainPage));
            tbSucess.Text = "";
            tbAMKA.Text = "";
        }

        private void Grid_Loaded(object sender, RoutedEventArgs e)
        {
            var Items = new List<string>();
            string sSQL = @"SELECT [Store_Id],[Store_Name],[Address],[State],[Region],[ZIP],[Phone],[Email] from
dragstores where [Store_Id] =" + tbId.Text;
            ISQLiteStatement dbstate = dbcoonection.Prepare(sSQL);
            while (dbstate.Step() == SQLiteResult.ROW)
            {
                Items.Add("Όνομασία: " + dbstate["Store_Name"] + " Διεύθυνση: " + dbstate["Address"] + " Πόλη: " +
dbstate["Region"] + " Νομός: " + dbstate["State"] + " Τ.Κ.: " + dbstate["ZIP"] + " Τηλέφωνο: " + dbstate["Phone"] + " e-
mail: " + dbstate["Email"]);
            }
            listdrugstore.ItemsSource = Items;
        }

        private void bSearch_Click(object sender, RoutedEventArgs e)
        {
            var Items = new List<string>();
            string sSQL = @"SELECT
[Cust_Lname],[Cust_Fname],[Cust_AMKA],[Cust_Address],[Cust_Region],[Cust_ZIP],[Cust_Phone],[Cust_mail] from
customers where [Cust_AMKA] like "" + tbAMKA.Text + "%"";
            ISQLiteStatement dbstate = dbcoonection.Prepare(sSQL);
            while (dbstate.Step() == SQLiteResult.ROW)
```

```

        {
            Items.Add("Όνοματεπώνυμο: " + dbstate["Cust_Lname"] + " " + dbstate["Cust_Fname"] + " Διεύθυνση: " +
dbstate["Cust_Address"] + " Πόλη: " + dbstate["Cust_Region"] + " Τ.Κ.: " + dbstate["Cust_ZIP"] + " Τηλέφωνο: " +
dbstate["Cust_Phone"] + " e-mail: " + dbstate["Cust_mail"]);
        }
        listcustomer.ItemsSource = Items;

        var Items1 = new List<string>();
        sSQL = @"SELECT [amka],[Drag],[Quant] from preskr where [amka] ='" + tbAMKA.Text + "'";
        dbstate = dbcoonection.Prepare(sSQL);
        while (dbstate.Step() == SQLiteResult.ROW)
        {
            Items1.Add(dbstate["Drag"] + " " + dbstate["Quant"]);
        }
        listDrugs.ItemsSource = Items1;

    }

    private void tbSearch_GotFocus(object sender, RoutedEventArgs e)
    {
        tbAMKA.Text = "";
    }

    private void listdrugstore_SelectionChanged(object sender, SelectionChangedEventArgs e)
    {
    }

    private void listcustomer_SelectionChanged(object sender, SelectionChangedEventArgs e)
    {
    }

    private void bSuccess_Click(object sender, RoutedEventArgs e)
    {
        String sSQL = @"delete from preskr where [amka] like '" + tbAMKA.Text + "%'";
        ISQLiteStatement dbstate = dbcoonection.Prepare(sSQL);
        dbstate.Step();
        tbSucess.Text = "Η συνταγή εκτελέστηκε με επιτυχία!";
    }

    private void tbSucess_SelectionChanged(object sender, RoutedEventArgs e)
    {
    }

    private void listDrugs_SelectionChanged(object sender, SelectionChangedEventArgs e)
    {
    }
}
}

```

33 ΠΑΡΑΡΤΗΜΑ Ι – ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ (VISUAL BASIC)

Παρακάτω επισυνάπτεται ο πηγαίος κώδικας της κεντρικής εφαρμογής που είναι γραμμένη σε Visual Basic.

33.1 Πίνακας Κοινών Μεταβλητών

Module Module1

```
Public recnum As Integer 'Μετρητής Εγγραφών  
Public DoctorLName, DoctorFname As String 'Επώνυμο και όνομα Γιατρού  
Public CustLastName As String 'Επώνυμο Πελάτη  
Public StoreName, UpdateType As String 'Επωνυμία Φαρμακείου  
Public NextForm As String 'Επόμενη Φορμα  
Public NewRes As Integer 'Α/Α Συνταγής  
Public Responce As String 'Έλεγχος Αποδοχής  
Public StoreNum As String 'Δείκτης Φαρμακείου  
Public DragQuant As Integer 'Ποσότητα Φαρμάκου  
Public LastCust As Integer 'Κωδικός Πελάτη  
Public DragEof, DreagDescr As String
```

End Module

33.2 Custom Tables Queries

33.2.1 Insert Queries

```
INSERT INTO [dbo].[Customers] ([Cust_Id], [Cust_Lname], [Cust_Fname], [Cust_Address], [Cust_Region],  
[Cust_ZIP], [Cust_Phone], [Cust_mail], [Cust_AMKA], [Cust_AFM], [Cust_TYPE], [Cust_Day], [Cust_sql])  
VALUES (@Cust_Id, @Cust_Lname, @Cust_Fname, @Cust_Address, @Cust_Region, @Cust_ZIP,  
@Cust_Phone, @Cust_mail, @Cust_AMKA, @Cust_AFM, @Cust_TYPE, @Cust_Day, @Cust_sql)
```

33.2.2 Update Queries

```
UPDATE dbo.Customers set  
Cust_Lname=@Cust_Lname,Cust_Fname=@Cust_Fname,Cust_Address=@Cust_Address,  
Cust_Region=@Cust_Region,Cust_ZIP=@Cust_ZIP, Cust_Phone=@Cust_Phone, Cust_mail=@Cust_mail,  
Cust_AMKA=@Cust_AMKA, Cust_AFM=@Cust_AFMwhere cust_id=@Cust_Id
```

33.2.3 ReceiptsMain Queries

```
INSERT INTO [dbo].[ReceiptsMain] ([Res_Id], [Res_Date], [Res_Doc], [Res_Cust], [Res_Rems], [Store01],  
[Store02], [Store03], [Res_Check], [Res_descr]) VALUES (@Res_Id, @Res_Date, @Res_Doc, @Res_Cust,  
@Res_Rems, @Store01, @Store02, @Store03, @Res_Check, @Res_descr)
```

33.2.4 ReceiptsDetails Queries

```
INSERT INTO [dbo].[ReceiptsDetails] ([SqlId], [Res_Num], [Drag_Eof], [Drag_Desc], [Drag_Quant]) VALUES  
(@SqlId, @Res_Num, @Drag_Eof, @Drag_Desc, @Drag_Quant)
```

33.3 Συνολικός Κώδικας Αρχικής Φόρμας

Public Class StartForm

Private Sub StartForm_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'TODO: This line of code loads data into the 'ErgasiaDataSet1.AthensZip' table. You can move, or remove it, as needed.

Me.AthensZipTableAdapter.Fill(Me.ErgasiaDataSet1.AthensZip)

'TODO: This line of code loads data into the 'ErgasiaDataSet.DragStore' table. You can move, or remove it, as needed.

Me.DragStoreTableAdapter.Fill(Me.ErgasiaDataSet.DragStore)

'TODO: This line of code loads data into the 'ErgasiaDataSet.iatroi' table. You can move, or remove it, as needed.

Me.IatroiTableAdapter.Fill(Me.ErgasiaDataSet.iatroi)

End Sub

Private Sub TextBox3_TextChanged_1(sender As Object, e As EventArgs) Handles TextBox3.TextChanged

'-----Αναζήτηση Αρχείου Φαρμακείων με Επωνυμία-----

Me.DragStoreBindingSource.Filter = "Trim(Store_Name) like '%" & Trim(TextBox3.Text) & "%"

End Sub

Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles

ExitToolStripMenuItem.Click

End

End Sub

Private Sub TabPage1_Click(sender As Object, e As EventArgs) Handles TabPage1.Click

NextForm = "DoctorForm"

End Sub

Private Sub TabPage2_Click(sender As Object, e As EventArgs) Handles TabPage2.Click

NextForm = "StoresForm"

End Sub

Private Sub TextBox1_TextChanged_1(sender As Object, e As EventArgs) Handles TextBox1.TextChanged

'-----Αναζήτηση Αρχείου Γιατρών με Επώνυμο-----

Me.IatroiBindingSource.Filter = "LastNmae like '" & TextBox1.Text & "%"

End Sub

Private Sub TextBox2_TextChanged_1(sender As Object, e As EventArgs) Handles TextBox2.TextChanged

'-----Αναζήτηση Αρχείου Γιατρών με Τηλέφωνο-----

Me.IatroiBindingSource.Filter = "Phone like '" & TextBox2.Text & "%"

End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

If DoctorLName = "" Then

MsgBox("Δεν έχει επιλεγεί γιατρος ")

Exit Sub

End If

DoctorForm.Text = "Συνταγογράφηση Γιατρού " & DoctorLName & " " & DoctorFname

DoctorForm.Show()

End Sub

Private Sub DataGridView1_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick

' DoctorLName = DataGridView1.Rows(e.RowIndex).Cells(0).Value

' DoctorFname = DataGridView1.Rows(e.RowIndex).Cells(1).Value

```
DoctorLName = Me.IatroiBindingSource.Current("LastNmae")  
DoctorFname = Me.IatroiBindingSource.Current("FirstName")  
End Sub
```

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click  
If StoreName = "" Then  
MsgBox("Δεν έχει επιλεγεί Φαρμακείο")  
Exit Sub  
End If  
StoresForm.Show()  
End Sub
```

```
Private Sub DataGridView1_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles  
DataGridView1.CellContentClick  
End Sub
```

```
Private Sub Label1_Click(sender As Object, e As EventArgs) Handles Label1.Click  
End Sub
```

```
Private Sub Label2_Click(sender As Object, e As EventArgs) Handles Label2.Click  
End Sub
```

```
Private Sub Label3_Click(sender As Object, e As EventArgs) Handles Label3.Click  
End Sub
```

```
Private Sub Label5_Click(sender As Object, e As EventArgs) Handles Label5.Click  
End Sub
```

```
Private Sub Label11_Click(sender As Object, e As EventArgs) Handles Label11.Click  
End Sub
```

```
Private Sub GroupBox2_Enter(sender As Object, e As EventArgs) Handles GroupBox2.Enter  
End Sub
```

```
Private Sub Label14_Click(sender As Object, e As EventArgs) Handles Label14.Click  
End Sub
```

```
Private Sub TextBox14_TextChanged(sender As Object, e As EventArgs) Handles TextBox14.TextChanged  
End Sub
```

```
Private Sub DataGridView2_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles  
DataGridView2.CellContentClick  
End Sub
```

```
Private Sub DataGridView2_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles  
DataGridView2.CellClick  
StoreName = Me.DragStoreBindingSource.Current("Store_Name")  
End Sub  
End Class
```

33.4 Συνολικός Κώδικας Φόρμας Συνταγογραφήσεων

```
Public Class DoctorForm  
Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.DragStore' table. You can move, or remove it, as needed.

```
Me.DragStoreTableAdapter.Fill(Me.ErgasiaDataSet.DragStore)
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.LASTRESP' table. You can move, or remove it, as needed.

```
Me.LASTRESPTableAdapter.Fill(Me.ErgasiaDataSet.LASTRESP)
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.DRAGS' table. You can move, or remove it, as needed.

```
Me.DRAGSTableAdapter.Fill(Me.ErgasiaDataSet.DRAGS)
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.ReseipsDetails' table. You can move, or remove it, as needed.

```
Me.ReseipsDetailsTableAdapter.Fill(Me.ErgasiaDataSet.ReseipsDetails)
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.ReceipsMain' table. You can move, or remove it, as needed.

```
Me.ReceipsMainTableAdapter.Fill(Me.ErgasiaDataSet.ReceipsMain)
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.Customers' table. You can move, or remove it, as needed.

```
Me.CustomersTableAdapter.Fill(Me.ErgasiaDataSet.Customers)
```

```
' TextBox12.Text = 0
```

```
Me.TabControl1.SelectedTab = tabPage1
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.AthensZip' table. You can move, or remove it, as needed.

```
Me.AthensZipTableAdapter.Fill(Me.ErgasiaDataSet.AthensZip)
```

```
CustLastName = ""
```

```
End Sub
```

```
Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
```

```
ExitToolStripMenuItem.Click
```

```
StartForm.Show()
```

```
Me.Close()
```

```
End Sub
```

```
Private Sub TextBox3_TextChanged(sender As Object, e As EventArgs)
```

```
End Sub
```

```
Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs)
```

```
Me.ReceipsMainBindingSource.Filter = "Res_Cust=" & Val(TextBox1.Text)
```

```
Me.ReceipsMainBindingSource.Sort = "res_id asc"
```

```
End Sub
```

```
Private Sub TextBox12_TextChanged(sender As Object, e As EventArgs) Handles TextBox12.TextChanged
```

```
Me.ReseipsDetailsBindingSource.Filter = "Res_Num =" & Val(TextBox12.Text)
```

```
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
```

```
Me.LASTRESPTableAdapter.Fill(Me.ErgasiaDataSet.LASTRESP)
```

```
Me.DataGridView5.Visible = True
```

```
If Button1.Text = "Νέα Συνταγή" Then
```

```
NewRes = Val(TextBox21.Text)
```

```
NewRes = NewRes + 1
```

```
TextBox12.Text = NewRes
```

```
TextBox13.Text = Now
```

```
TextBox14.Text = DoctorLName
```

```
TextBox15.Text = TextBox1.Text
```

```
    TextBox16.Text = ""
    TextBox17.Text = ""
    TextBox18.Text = ""
    TextBox19.Text = ""
    TextBox20.Text = "ΑΝΕΚΤΕΛΕΣΤΗ"
    TextBox16.Focus()
    Button1.Text = "Καταχώρηση"

Else
    Button1.Text = "Νέα Συνταγή"
    Responce = MsgBox("Να καταχωρηθεί η συνταγή;", vbYesNo)
    If Responce = vbNo Then
        Exit Sub
    End If
    Me.ReceipsMainTableAdapter.InsertQuery(NewRes, TextBox13.Text, TextBox14.Text,
    TextBox15.Text, TextBox16.Text, TextBox17.Text, TextBox18.Text, TextBox19.Text, 0, TextBox20.Text)
    Me.ReceipsMainTableAdapter.Fill(Me.ErgasiaDataSet.ReceipsMain)
    Me.ReceipsMainBindingSource.MoveLast()
    Me.LASTRESPTTableAdapter.UpdateQuery(NewRes)
    Me.ReceipsMainBindingSource.MoveLast()
    Me.DataGridView5.Visible = False
    Button1.Enabled = False
End If
End Sub

Private Sub TextBox17_TextChanged(sender As Object, e As EventArgs) Handles TextBox17.TextChanged
End Sub

Private Sub TextBox17_GotFocus(sender As Object, e As EventArgs) Handles TextBox17.GotFocus
    StoreNum = 1
End Sub

Private Sub TextBox18_TextChanged(sender As Object, e As EventArgs) Handles TextBox18.TextChanged
End Sub

Private Sub TextBox18_GotFocus(sender As Object, e As EventArgs) Handles TextBox18.GotFocus
    StoreNum = 2
End Sub

Private Sub TextBox19_TextChanged(sender As Object, e As EventArgs) Handles TextBox19.TextChanged
End Sub

Private Sub TextBox19_GotFocus(sender As Object, e As EventArgs) Handles TextBox19.GotFocus
    StoreNum = 3
End Sub

Private Sub TextBox6_TextChanged(sender As Object, e As EventArgs)
    Me.DragStoreBindingSource.Filter = "sore_zip ='" & TextBox6.Text & "'"
End Sub

Private Sub DataGridView5_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles
DataGridView5.CellContentClick
End Sub
```

```
Private Sub DataGridView5_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles
DataGridView5.CellClick
    Select Case StoreNum
        Case Is = 1
            TextBox17.Text = DataGridView5.Rows(e.RowIndex).Cells(0).Value
        Case Is = 2
            TextBox18.Text = DataGridView5.Rows(e.RowIndex).Cells(0).Value
        Case Is = 3
            TextBox19.Text = DataGridView5.Rows(e.RowIndex).Cells(0).Value
    End Select
End Sub
```

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    If TextBox20.Text = "ΕΧΕΙ ΕΚΤΕΛΕΣΤΕΙ " Then
        MsgBox("Η ΣΥΝΤΑΓΗ ΕΧΕΙ ΕΚΤΕΛΕΣΤΕΙ ΔΕΝ ΕΙΝΑΙ ΔΥΝΑΤΗ Η ΠΡΟΣΘΗΚΗ ΕΙΔΟΥΣ")
        Exit Sub
    End If
    If DragEof = "" Then
        MsgBox("Δεν έχει πειλεγεί είδος")
        Exit Sub
    End If

    DragQuant = InputBox("Ποσότητα Είδους " & DragEof & " " & DreagDescr)
    Responce = MsgBox("Να καταχωρηθεί το είδος;", vbYesNo)
    If Responce = vbYes Then
        Me.ReseipsDetailsTableAdapter.InsertQuery(1, Val(TextBox12.Text), DragEof, DreagDescr,
        DragQuant)
        Me.ReseipsDetailsTableAdapter.Fill(Me.ErgasiaDataSet.ReseipsDetails)
        DragEof = ""
    End If
    Button1.Enabled = True
End Sub
```

```
Private Sub DataGridView4_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles
DataGridView4.CellContentClick
End Sub
```

```
Private Sub DataGridView4_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles
DataGridView4.CellClick
    TextBox22.Text = Me.DataGridView4.Rows(e.RowIndex).Cells(0).Value
    TextBox23.Text = Me.DataGridView4.Rows(e.RowIndex).Cells(1).Value
    DragEof = Me.DRAGSBindingSource.Current("EOF_CODE")
    DreagDescr = Me.DRAGSBindingSource.Current("DESCRIPTION")
End Sub
```

```
Private Sub TextBox24_TextChanged(sender As Object, e As EventArgs) Handles TextBox24.TextChanged
    Me.DRAGSBindingSource.Filter = "DESCRIPTION like '" & TextBox24.Text & "%"
End Sub
```

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    GroupBox1.Enabled = True
    Me.CustomersBindingSource.Sort = "cust_id asc"
    Me.CustomersBindingSource.MoveLast()
    LastCust = Val(TextBox1.Text)
    LastCust = LastCust + 1
End Sub
```



```
MsgBox(LastCust)
TextBox1.Text = LastCust
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ""
TextBox5.Text = ""
TextBox6.Text = ""
TextBox7.Text = ""
TextBox8.Text = ""
TextBox9.Text = ""
TextBox10.Text = ""
TextBox11.Text = ""
End Sub

Private Sub TextBox2_TextChanged(sender As Object, e As EventArgs)
End Sub

Private Sub GroupBox1_Enter(sender As Object, e As EventArgs) Handles GroupBox1.Enter
End Sub

Private Sub Label8_Click(sender As Object, e As EventArgs)
End Sub

Private Sub TextBox8_TextChanged(sender As Object, e As EventArgs)
End Sub

Private Sub Label9_Click(sender As Object, e As EventArgs)
End Sub

Private Sub TextBox9_TextChanged(sender As Object, e As EventArgs)
End Sub

Private Sub Label10_Click(sender As Object, e As EventArgs)
End Sub

Private Sub Label1_Click(sender As Object, e As EventArgs) Handles Label1.Click
End Sub

Private Sub TextBox1_TextChanged_1(sender As Object, e As EventArgs) Handles TextBox1.TextChanged
    Me.ReceipsMainBindingSource.Filter = "Res_Cust=" & Val(TextBox1.Text)
    Me.ReceipsMainBindingSource.Sort = "res_id asc"
End Sub

Private Sub Label2_Click(sender As Object, e As EventArgs) Handles Label2.Click
End Sub

Private Sub TextBox2_TextChanged_1(sender As Object, e As EventArgs) Handles TextBox2.TextChanged
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
    Response = MsgBox("Να καταχωρηθεί ο Πελάτης ;", vbYesNo)
    If Response = vbNo Then
        Me.Panel1.Visible = False
        Exit Sub
    End If
End Sub
```

```

    Me.CustomersTableAdapter.InsertQuery>LastCust, TextBox2.Text, TextBox3.Text, TextBox4.Text,
    TextBox5.Text, TextBox6.Text, TextBox7.Text, TextBox8.Text, TextBox9.Text, TextBox10.Text,
    TextBox11.Text, Now, 0)
    Me.CustomersTableAdapter.Fill(Me.ErgasiaDataSet.Customers)
    Me.CustomersBindingSource.MoveLast()
End Sub

```

```

Private Sub ΝέοςΠελατηςToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ΝέοςΠελατηςToolStripMenuItem.Click
    GroupBox1.Enabled = True
    Me.CustomersBindingSource.Sort = "cust_id asc"
    Me.CustomersBindingSource.MoveLast()
    LastCust = Val(TextBox1.Text)
    LastCust = LastCust + 1
    MsgBox>LastCust)
    TextBox1.Text = LastCust
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    TextBox7.Text = ""
    TextBox8.Text = ""
    TextBox9.Text = ""
    TextBox10.Text = ""
    TextBox11.Text = ""
    UpdateType = "New"
End Sub

```

```

Private Sub DoctorForm_KeyDown(sender As Object, e As KeyEventArgs) Handles Me.KeyDown
'KEY DOWN F10=121-----
If e.KeyValue = 121 Then
    'UpdateType =CORRECTION OF UPDATES OR NEW FOR NEW CUSTOMER
    Select Case UpdateType
        Case Is = "Corection"
            Responce = MsgBox("Να καταχωρηθούν οι Διορθώσεις;", vbYesNo)
            If Responce = vbNo Then
                Panel1.Visible = False
                GroupBox1.Enabled = False
                Exit Sub
            End If
            Me.CustomersTableAdapter.UpdateQuery(TextBox2.Text, TextBox3.Text, TextBox4.Text,
            TextBox5.Text, TextBox6.Text, TextBox7.Text, TextBox8.Text, TextBox9.Text, TextBox10.Text,
            TextBox11.Text, Val(TextBox1.Text))
            GroupBox1.Enabled = False
            Case Is = "New"
                Responce = MsgBox("Να καταχωρηθεί ο Πελατης;", vbYesNo)
                If Responce = vbNo Then Exit Sub
                Me.CustomersTableAdapter.InsertQuery>LastCust, TextBox2.Text, TextBox3.Text,
                TextBox4.Text, TextBox5.Text, TextBox6.Text, TextBox7.Text, TextBox8.Text, TextBox9.Text,
                TextBox10.Text, TextBox11.Text, Now, 0)
                Me.CustomersTableAdapter.Fill(Me.ErgasiaDataSet.Customers)
                Me.CustomersBindingSource.MoveLast()
                GroupBox1.Enabled = False
            End Select

```

```
End If  
End Sub
```

```
Private Sub ΔιόρθωσηΣτοιχείωνToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles  
ΔιόρθωσηΣτοιχείωνToolStripMenuItem.Click  
    GroupBox1.Enabled = True  
    UpdateType = "Corection"  
End Sub
```

```
Private Sub ΔιαχείρισηΠελατολογίουToolStripMenuItem_Click(sender As Object, e As EventArgs)  
Handles ΔιαχείρισηΠελατολογίουToolStripMenuItem.Click  
End Sub
```

```
Private Sub TextBox25_TextChanged(sender As Object, e As EventArgs) Handles TextBox25.TextChanged  
    Me.CustomersBindingSource.Filter = "Cust_AMKA like '" & TextBox25.Text & "%"  
End Sub
```

```
Private Sub Label21_Click(sender As Object, e As EventArgs) Handles Label21.Click  
End Sub
```

```
Private Sub Label22_Click(sender As Object, e As EventArgs) Handles Label22.Click  
End Sub
```

```
Private Sub TextBox26_TextChanged(sender As Object, e As EventArgs) Handles TextBox26.TextChanged  
    Me.CustomersBindingSource.Filter = "Cust_Lname like '" & TextBox26.Text & "%"  
End Sub
```

```
Private Sub DataGridView6_CellContentClick(sender As Object, e As DataGridViewCellEventArgs)  
End Sub
```

```
Private Sub TextBox5_TextChanged(sender As Object, e As EventArgs) Handles TextBox5.TextChanged  
End Sub
```

```
Private Sub TextBox5_KeyDown(sender As Object, e As KeyEventArgs) Handles TextBox5.KeyDown  
    If e.KeyValue = 112 Then  
        Me.Panel1.Visible = True  
    End If  
End Sub
```

```
Private Sub DataGridView1_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles  
DataGridView1.CellContentClick  
End Sub
```

```
Private Sub Label23_Click(sender As Object, e As EventArgs)  
End Sub
```

```
Private Sub TabPage1_Click(sender As Object, e As EventArgs) Handles TabPage1.Click  
End Sub
```

```
Private Sub TextBox22_TextChanged(sender As Object, e As EventArgs) Handles TextBox22.TextChanged  
End Sub
```

```
Private Sub DataGridView6_CellContentClick_1(sender As Object, e As DataGridViewCellEventArgs)  
End Sub
```

```
Private Sub Label23_Click_1(sender As Object, e As EventArgs)  
End Sub
```

```
Private Sub DataGridView6_CellContentClick_2(sender As Object, e As DataGridViewCellEventArgs)
```

```
End Sub
Private Sub Panel1_Paint(sender As Object, e As PaintEventArgs)
End Sub
Private Sub DataGridView6_CellContentClick_3(sender As Object, e As DataGridViewCellEventArgs)
Handles DataGridView6.CellContentClick
End Sub
Private Sub DataGridView6_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles
DataGridView6.CellClick
    TextBox5.Text = Me.DataGridView6.Rows(e.RowIndex).Cells(3).Value
    TextBox6.Text = Me.DataGridView6.Rows(e.RowIndex).Cells(2).Value
    Me.Panel1.Visible = False
End Sub

Private Sub TextBox27_TextChanged(sender As Object, e As EventArgs) Handles TextBox27.TextChanged
    Me.AthensZipBindingSource.Filter = "town LIKE '" & TextBox27.Text & "%"
End Sub
Private Sub Panel1_Paint_1(sender As Object, e As PaintEventArgs) Handles Panel1.Paint
End Sub
Private Sub TabPage2_Click(sender As Object, e As EventArgs) Handles TabPage2.Click
End Sub
Private Sub DataGridView1_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles
DataGridView1.CellClick
    CustLastName = Me.CustomersBindingSource.Current("Cust_Lname")
End Sub
Private Sub TabPage2_DragOver(sender As Object, e As DragEventArgs) Handles TabPage2.DragOver
End Sub
Private Sub TextBox23_TextChanged(sender As Object, e As EventArgs) Handles TextBox23.TextChanged
End Sub
End Class
```

33.5 Συνολικός Κώδικας εκτέλεσης συνταγών

Public Class StoresForm

```
Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ExitToolStripMenuItem.Click
    StartForm.Show()
    Me.Close()
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If TextBox20.Text = "ΕΧΕΙ ΕΚΤΕΛΕΣΤΕΙ " Then
        MsgBox("Η ΣΥΝΤΑΓΗ ΕΧΕΙ ΕΚΤΕΛΕΣΤΕΙ")
        Exit Sub
    End If
    Me.ReceipsMainTableAdapter.UpdateQuery("ΕΧΕΙ ΕΚΤΕΛΕΣΤΕΙ", Val(TextBox12.Text))
    Me.ReceipsMainTableAdapter.Fill(Me.ErgasiaDataSet.ReceipsMain)
End Sub
```

```
Private Sub StoresForm_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'TODO: This line of code loads data into the 'ErgasiaDataSet.DragStore' table. You can move, or remove
it, as needed.
    Me.DragStoreTableAdapter.Fill(Me.ErgasiaDataSet.DragStore)
    'TODO: This line of code loads data into the 'ErgasiaDataSet.AthensZip' table. You can move, or remove
it, as needed.
    Me.AthensZipTableAdapter.Fill(Me.ErgasiaDataSet.AthensZip)
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.LASTRESP' table. You can move, or remove it, as needed.

```
Me.LASTRESPTableAdapter.Fill(Me.ErgasiaDataSet.LASTRESP)
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.DRAGS' table. You can move, or remove it, as needed.

```
Me.DRAGSTableAdapter.Fill(Me.ErgasiaDataSet.DRAGS)
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.ReseipsDetails' table. You can move, or remove it, as needed.

```
Me.ReseipsDetailsTableAdapter.Fill(Me.ErgasiaDataSet.ReseipsDetails)
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.ReceipsMain' table. You can move, or remove it, as needed.

```
Me.ReceipsMainTableAdapter.Fill(Me.ErgasiaDataSet.ReceipsMain)
```

'TODO: This line of code loads data into the 'ErgasiaDataSet.Customers' table. You can move, or remove it, as needed.

```
Me.CustomersTableAdapter.Fill(Me.ErgasiaDataSet.Customers)
```

```
End Sub
```

```
Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles TextBox1.TextChanged
```

```
Me.ReceipsMainBindingSource.Filter = "Res_Cust=" & Val(TextBox1.Text)
```

```
Me.ReceipsMainBindingSource.Sort = "res_id asc"
```

```
End Sub
```

```
Private Sub TextBox12_TextChanged(sender As Object, e As EventArgs) Handles TextBox12.TextChanged
```

```
Me.ReseipsDetailsBindingSource.Filter = "Res_Num =" & Val(TextBox12.Text)
```

```
End Sub
```

```
Private Sub TextBox20_TextChanged(sender As Object, e As EventArgs) Handles TextBox20.TextChanged
```

```
If TextBox20.Text = "ΕΧΕΙ ΕΚΤΕΛΕΣΤΕΙ " Then
```

```
TextBox20.BackColor = Color.Red
```

```
Else
```

```
TextBox20.BackColor = Color.White
```

```
End If
```

```
End Sub
```

```
Private Sub TextBox25_TextChanged(sender As Object, e As EventArgs) Handles TextBox25.TextChanged
```

```
Me.CustomersBindingSource.Filter = "Cust_AMKA like " & TextBox25.Text & "%"
```

```
End Sub
```

```
Private Sub TextBox26_TextChanged(sender As Object, e As EventArgs) Handles TextBox26.TextChanged
```

```
Me.CustomersBindingSource.Filter = "Cust_Lname like " & TextBox26.Text & "%"
```

```
End Sub
```

```
End Class
```

33.6 Επιμέρους Κώδικας Επιλογής Ιατρών

```
Private Sub DataGridView1_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles
```

```
DataGridView1.CellClick
```

```
DoctorLName = DataGridView1.Rows(e.RowIndex).Cells(0).Value
```

```
DoctorFname = DataGridView1.Rows(e.RowIndex).Cells(1).Value
```

```
DoctorLName = Me.IatroiBindingSource.Current("LastNmae")
```

```
DoctorFname = Me.IatroiBindingSource.Current("FirstName")
```

```
End Sub
```

'-----Είναι δυνατή η αναζήτηση με μέρος του επώνυμο η του τηλεφώνου-----

```
Private Sub TextBox1_TextChanged_1(sender As Object, e As EventArgs) Handles TextBox1.TextChanged
    '-----Αναζήτηση Αρχείου Γιατρών με Επώνυμο-----
    Me.IatroiBindingSource.Filter = "LastNmae like '" & TextBox1.Text & "%'"
End Sub
```

```
Private Sub TextBox2_TextChanged_1(sender As Object, e As EventArgs) Handles TextBox2.TextChanged
    '-----Αναζήτηση Αρχείου Γιατρών με Τηλέφωνο-----
    Me.IatroiBindingSource.Filter = "Phone like '" & TextBox2.Text & "%'"
End Sub
```

33.6.1.1 Νέος Πελάτης και Διόρθωση Στοιχείων

```
Private Sub ΝέοςΠελατηςToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ΝέοςΠελατηςToolStripMenuItem.Click
    GroupBox1.Enabled = True
    Me.CustomersBindingSource.Sort = "cust_id asc"
    Me.CustomersBindingSource.MoveLast()
    LastCust = Val(TextBox1.Text)
    LastCust = LastCust + 1
    MsgBox(LastCust)
    TextBox1.Text = LastCust
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox4.Text = ""
    TextBox5.Text = ""
    TextBox6.Text = ""
    TextBox7.Text = ""
    TextBox8.Text = ""
    TextBox9.Text = ""
    TextBox10.Text = ""
    TextBox11.Text = ""
    UpdateType = "New"
End Sub
```

'-----Στην Διόρθωση απλώς επιτρέπει την διόρθωση των στοιχείων και ενημερώνει τον
διακόπτη UpdateType = "Corection"-----

```
Private Sub ΔιόρθωσηΣτοιχειωνToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
ΔιόρθωσηΣτοιχειωνToolStripMenuItem.Click
    GroupBox1.Enabled = True
    UpdateType = "Corection"
End Sub
```

'----- Με το F10 γίνονται η καταχωρήσεις μετά από επιβεβαίωση-----

```
Private Sub DoctorForm_KeyDown(sender As Object, e As KeyEventArgs) Handles Me.KeyDown
    'KEY DOWN F10=121-----
    If e.KeyValue = 121 Then
        'UpdateType =CORRECTION OF UPDATES OR NEW FOR NEW CUSTOMER
        Select Case UpdateType
            Case "Corection"
                Responce = MsgBox("Να καταχωρηθούν οι Διορθώσεις ;", vbYesNo)
                If Responce = vbNo Then
                    Panel1.Visible = False
                    GroupBox1.Enabled = False
```

```

        Exit Sub
    End If
    Me.CustomersTableAdapter.UpdateQuery(TextBox2.Text, TextBox3.Text, TextBox4.Text,
    TextBox5.Text, TextBox6.Text, TextBox7.Text, TextBox8.Text, TextBox9.Text, TextBox10.Text,
    TextBox11.Text, Val(TextBox1.Text))
    GroupBox1.Enabled = False
    Case = "New"
    Responce = MsgBox("Να καταχωρηθεί ο Πελατης;", vbYesNo)
    If Responce = vbNo Then Exit Sub
    Me.CustomersTableAdapter.InsertQuery(LastCust, TextBox2.Text, TextBox3.Text,
    TextBox4.Text, TextBox5.Text, TextBox6.Text, TextBox7.Text, TextBox8.Text, TextBox9.Text,
    TextBox10.Text, TextBox11.Text, Now, 0)
    Me.CustomersTableAdapter.Fill(Me.ErgasiaDataSet.Customers)
    Me.CustomersBindingSource.MoveLast()
    GroupBox1.Enabled = False
End Select
End If
End Sub

```

33.7 Καταχώρηση νέας συνταγής

Button1.Text = "Καταχώρηση"

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Me.LASTRESPTableAdapter.Fill(Me.ErgasiaDataSet.LASTRESP)
    Me.DataGridView5.Visible = True
    If Button1.Text = "Νέα Συνταγή" Then
        NewRes = Val(TextBox21.Text)
        NewRes = NewRes + 1
        TextBox12.Text = NewRes
        TextBox13.Text = Now
        TextBox14.Text = DoctorLName
        TextBox15.Text = TextBox1.Text
        TextBox16.Text = ""
        TextBox17.Text = ""
        TextBox18.Text = ""
        TextBox19.Text = ""
        TextBox20.Text = "ΑΝΕΚΤΕΛΕΣΤΗ"
        TextBox16.Focus()
        Button1.Text = "Καταχώρηση"

    Else
        Button1.Text = "Νέα Συνταγή"
        Responce = MsgBox("Να καταχωρηθεί η συνταγή;", vbYesNo)
        If Responce = vbNo Then
            Exit Sub
        End If
        Me.ReceipsMainTableAdapter.InsertQuery(NewRes, TextBox13.Text, TextBox14.Text,
        TextBox15.Text, TextBox16.Text, TextBox17.Text, TextBox18.Text, TextBox19.Text, 0, TextBox20.Text)
        Me.ReceipsMainTableAdapter.Fill(Me.ErgasiaDataSet.ReceipsMain)
        Me.ReceipsMainBindingSource.MoveLast()
        Me.LASTRESPTableAdapter.UpdateQuery(NewRes)
    End If
End Sub

```

```

Me.ReceipsMainBindingSource.MoveLast()
Me.DataGridView5.Visible = False
Button1.Enabled = False
End If
End Sub

```

34 ΠΑΡΑΡΤΗΜΑ Ι – ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ (HTML-ASP.NET)

Παρακάτω επισυνάπτεται ο πηγαίος κώδικας της δικτυακής εφαρμογής που είναι γραμμένη σε HTML – ASP.NET.

34.1 GENERAL SCRIPT

```

<html>
<frameset rows="25%,80%" >
<frame name="nav" src="TOPPAGE.HTML">
<frame name="top" id="top" ROWS=50%,*"
<frame name="last" id="last" src="menupage2.ASP">
</frameset>
</frameset>
</form>
</html>

```

34.2 ΑΝΑΖΗΤΗΣΗ ΓΙΑΤΡΩΝ ΜΕ ΟΝΟΜΑ

```

<!DOCTYPE html>
<html>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<body style="background-color:RGB(153,217,234);">
<h3>m □ g</h3>

<form action="iatroi-name.asp" method="GET" />
<input type="text" name="CODE" list="CODE">
<input type="submit" value="m □ g 査6x " name="submit" id="submit">
</form>
<%Dim conn
Set conn = Server.CreateObject("ADODB.Connection")
conn.Open "Provider=SQLOLEDB; Data Source = localhost; Initial Catalog = ergasia; User Id = vissas;
Password=1T33w5Yrez3"
set rs=Server.CreateObject("ADODB.recordset")
SET RS1=Server.CreateObject("ADODB.recordset")
rs.Open "SELECT distinct Specialty from iatroi ORDER BY Specialty ", conn
RS1.OPEN "SELECT distinct town from AthensZip ORDER BY town ", conn%>
<form action="iatroi-REGION.asp" method="GET" />
<select name="CODE" onChange="combo(this, 'theinput')">
<% do until rs.eof%>
<%for each x in rs.fields%>
<option><%response.write(x.value)%></option>
<%next
rs.movenext%>
<%loop
rs.close
%>

```



```

</select)">
  <input type="submit" value="m g 藥" name="submit" id="submit">
</form>
<hr>
<h3>m g 藥span</h3>

<form action="DRAGSTORES.asp" method="GET" />

<select name="CODE" onChange="combo(this, 'theinput')">
  <% do until rs1.eof%>
  <%for each x in rs1.fields%>
  <option><%response.write(x.value)%></option>
  <%next
  rs1.movenext%>
  <%loop
  rs1.close
  %>
</select)">
<input type="submit" value="m g 藥" name="submit" id="submit">
</form>
<hr>
<h3>藥 X藥span</h3>
  
  <form action="allcustomers.asp" method="GET" />
  <input type="text" name="CODE" list="CODE">
  <input type="submit" value="藥 X藥" name="submit" id="submit">
</form>
<hr>
</body>
</html>

```

34.3 ΑΝΑΖΗΤΗΣΗ ΓΙΑΤΡΩΝ ΜΕ ΕΙΔΙΚΟΤΗΤΑ

```

SET RS1=Server.CreateObject("ADODB.recordset")
rs.Open "SELECT distinct Specialty from iatroi ORDER BY Specialty ", conn
<form action="iatroi-REGION.asp" method="GET" />
<select name="CODE" onChange="combo(this, 'theinput')">
  <% do until rs.eof%>
  <%for each x in rs.fields%>
  <option><%response.write(x.value)%></option>
  <%next
  rs.movenext%>
  <%loop
  rs.close
  %>
</select)">
  <input type="submit" value="m g 藥" name="submit" id="submit">
</form>

```

34.4 ΑΝΑΖΗΤΗΣΗ ΦΑΡΜΑΚΕΙΩΝ ΜΕ ΠΕΡΙΟΧΗ

```

RS1.OPEN "SELECT distinct town from AthensZip ORDER BY town ", conn%>

```

```

<h3>m  培span</h3>

<form action="DRAGSTORES.asp" method="GET" />
<select name="CODE" onChange="combo(this, 'theinput')">
  <% do until rs1.eof%>
  <%for each x in rs1.fields%>
  <option><%response.write(x.value)%></option>
  <%next
  rs1.movenext%>
  <%loop
  rs1.close
  %>
</select">
<input type="submit" value="m  培 欄 ·  name="submit" id="submit">
</form>

```

34.5 ΑΝΑΖΗΤΗΣΗ ΠΕΛΑΤΩΝ ΚΑΙ ΣΥΝΤΑΓΩΝ

res-received.ASP SCRIPTS

```

<html>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<BODY style="background-color:RGB(0,162,232);">
<button onclick="goBack()">Go Back</button>
<BR>
<script>
function goBack() {
  window.history.back();
}
</script>
<%
  Dim conn
  Set conn = Server.CreateObject("ADODB.Connection")
  conn.Open "Provider=SQLOLEDB; Data Source = localhost; Initial Catalog = ergasia; User Id = vissas;
  Password=1T33w5Yrez3"

  If conn.errors.count = 0 Then
    Response.Write "SϚϚϚ " & Request.QueryString("code")
  End If
  set rs=Server.CreateObject("ADODB.recordset")
  rs.Open "Select * from syntages where Res_Cust = " & Request.QueryString("code"), conn
  SET RS1=Server.CreateObject("ADODB.recordset")
  rs1.Open "Select * from CUSTOMERS where CUST_ID = " & Request.QueryString("code"), conn%>

<table border="1" width="100%">
<% do until rs1.eof%>
<%for each x in rs1.fields%>
<td><b><font face="Arial" size=1><%response.write(x.value)%></FONT></b></td>
<%next
rs1.movenext%>
</tr>
<%loop
rs1.close
%>

```

```
</table>
```

```
<%Response.Write "æCœ" %>
```

```
<table border="1" width="100%">
```

```
<% do until rs.eof%>
```

```
<%for each x in rs.fields%>
```

```
<td><b><font face="Arial" size=1><%response.write(x.value)%></FONT></b></td>
```

```
<%next
```

```
rs.movenext%>
```

```
</tr>
```

```
<%loop
```

```
rs.close
```

```
%>
```

```
</table>
```

```
<form action="UPDATE.asp" method="GET" />
```

```
<input type="text" name="CODE" list="CODE">
```

```
<input type="submit" value="ÛÏÏÏ o`" name="submit" id="submit">
```

```
</form>
```

```
</BODY>
```

```
</html>
```

34.6 ΕΚΤΕΛΕΣΗ ΣΥΝΤΑΓΗΣ

```
<html>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<BODY style="background-color:RGB(0,162,232);">
```

```
<BODY style="background-color:RGB(0,162,232);">
```

```
<button onclick="goBack()">Go Back</button>
```

```
<BR>
```

```
<script>
```

```
function goBack() {
```

```
    window.history.back();
```

```
}
```

```
</script>
```

```
<button onclick="myFunction()">ÛÏÏÏ <%= Request.QueryString("code")%> </button>
```

```
<p id="demo"></p>
```

```
<%
```

```
    Dim conn
```

```
        Set conn = Server.CreateObject("ADODB.Connection")
```

```
        conn.Open "Provider=SQLOLEDB; Data Source = localhost; Initial Catalog = ergasia; User Id = vissas; Password=1T33w5Yrez3"
```

```
    set rs=Server.CreateObject("ADODB.recordset")
```

```
    SET RS1=Server.CreateObject("ADODB.recordset")
```

```
    %>
```

```
<script>
```

```
    function myFunction() {
```

```
    var x;
    if (confirm("Press a button!") == true) {
        x = "update";
        <%rs1.Open "UPDATE ReceiptsMain SET Res_descr=' &rs1.Res_descr &rs1.Res_descr ' where Res_Id = " &
Request.QueryString("code"), conn%>
    } else {
        x = "Cancel";
    }
    document.getElementById("demo").innerHTML = x;
}

</script>
</BODY>
</html>
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Silbershatz Galvin – Gagne, “Λειτουργικά Συστήματα” , Δεύτερη Ελληνική Έκδοση, Εκδόσεις Ιων.
- [2] Παναγιώτης Δ. Κεντερλής, “Ανάπτυξη Διαδικτυακών Εφαρμογών”, *Πρώτη Έκδοση, Αθήνα 2009* .
- [3] Robert Sedgewich, “Αλγόριθμοι Σε C”, *Τρίτη Αμερικάνικη Έκδοση, Εκδόσεις Κλειδάριθμος*.
- [4] MSc Χρήστος Δ. Κυτάγιας, MSc Κώστας Δ. Κυτάγιας, Δρ Γιώργιος Ν. Πρεζεράκος, Δρ Δημήτρης Χ. Κυτάγιας, “Εισαγωγή Στον Αντικειμενοστρεφή Προγραμματισμό”, Έκδοση Σύγχρονη Εκδοτική, Αθήνα 2008.
- [5] Δρ Κυτάγιας Δ., Δρ Ψαρομήλιγκος Ι., “Visual Basic Από Την Θεωρία Στο Εργαστήριο”, Έκδοση Δηρός.
- [6] <https://www.pluralsight.com/blog/software-development/why-are-mobile-app-developers-avoiding-windows-phone-8>
- [7] <http://www.24hrtech.co/main-advantages-and-disadvantages-of-windows-phone-8/>