



## ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ  
ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

### Ολοκληρωμένη Εφαρμογή Διαχείρισης Ηλεκτρονικού Φαρμακείου

#### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Π. Αυγουστής

Γεώργιος Κ. Κοτσανάς

**Επιβλέπων :** Δημήτριος Δ. Κουτσούρης  
Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2011





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ  
ΥΛΙΚΩΝ

## Ολοκληρωμένη Εφαρμογή Διαχείρισης Ηλεκτρονικού Φαρμακείου

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Γεώργιος Π. Αυγουστής

Γεώργιος Κ. Κοτσανάς

**Επιβλέπων :** Δημήτριος Δ. Κουτσούρης  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 1<sup>η</sup> Φεβρουαρίου 2011.

(Υπογραφή)

.....  
Δημήτριος Κουτσούρης  
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....  
Κωνσταντίνα Νικήτα  
Καθηγήτρια Ε.Μ.Π.

(Υπογραφή)

.....  
Παναγιώτης Τσανάκας  
Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2011

*(Υπογραφή)*

.....

Γεώργιος Π. Αυγουστής

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

*(Υπογραφή)*

.....

Γεώργιος Κ. Κοτσανάς

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Γεώργιος Π. Αυγουστής, 2011

Copyright © Γεώργιος Κ. Κοτσανάς, 2011

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τους συγγραφείς και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η χρησιμοποίηση των δυνατοτήτων που μας δίνει η τεχνολογία του διαδικτύου (internet) προς όφελος του τομέα υγείας. Συγκεκριμένα στόχος μας ήταν η δημιουργία μιας ιστοσελίδας που θα δρα ως ολοκληρωμένη εφαρμογή διαχείρισης ηλεκτρονικού φαρμακείου και θα παρέχει την δυνατότητα διατήρησης ιστορικού ασθενών καθώς και χορήγησης θεραπευτικής αγωγής από τους εκάστοτε γιατρούς. Επιπλέον, ενσωματώνει την δυνατότητα παραγγελίας φαρμάκων ελλείψει επαρκούς ποσότητας.

Πραγματοποιήθηκε μελέτη και χρήση της ATC κωδικοποίησης για τα φαρμακευτικά σκευάσματα και της ICD-10 κωδικοποίησης για την περιγραφή των ασθενειών. Επιπρόσθετα το σύνολο των φαρμακευτικών σκευασμάτων αντλήθηκε σύμφωνα με τις προδιαγραφές του Ε.Ο.Φ.. Τέλος η κατηγοριοποίηση των φαρμάκων πραγματοποιήθηκε σύμφωνα με την ονοματοδοσία του drugbank.

Η ιστοσελίδα υλοποιήθηκε σε λειτουργικό περιβάλλον windows 7 με την χρήση της γλώσσας προγραμματισμού C# και των εργαλείων που προσφέρουν οι τεχνολογίες Asp.net, Microsoft Visual Web Developer 2010 και SQL Server Management Studio. Ο web server που χρησιμοποιήθηκε είναι ο ενσωματωμένος Web development server της Microsoft, ο οποίος έχει τη δυνατότητα να εμφανίζει δυναμικές σελίδες και να επικοινωνεί με την βάση δεδομένων. Για την υλοποίηση της βάσης δεδομένων χρησιμοποιήθηκε η πλατφόρμα διαχείρισης Microsoft SQL Server 2008 R2.

**Λέξεις Κλειδιά:** Ηλεκτρονικό Φαρμακείο, ATC, ICD-10, Ε.Ο.Φ., DrugBank, C#, Asp.Net, SQL Server, Microsoft Visual Web Developer.



## **Abstract**

The purpose of this paper is to apply the internet technology and its capabilities to benefit the health sector. Specifically, our goal was to create a website that can be used as an integrated hospital application for drug administration to provide the ability to maintain a patients case history and to record the administration of medications used by each individual doctor. In addition it also incorporates the possibility of ordering drugs if there is an insufficient supply.

A study was completed using ATC coding to specify the drug needed and ICD-10 coding for the description of the diseases. Moreover all medications included are named under the terms of the E.O.F. (National Organization for Medicine). Finally the classification of drugs was made in accordance to the DrugBank names.

The website was implemented in the windows 7 operating system using the c# programming language and the tools which are available with Asp.Net, Microsoft Visual Web Developer 2010 and SQL Server management Studio. The Web server that was used, is the built in Web development server from Microsoft, which has the ability to process dynamic web-pages and connect with the database. The Microsoft SQL Server 2008 R2 platform is used for database administration.

**Keywords:** electronic pharmacy, ATC, ICD-10, E.O.F., DrugBank, C#, Asp.Net, SQL Server, Microsoft Visual Web Developer.

## **Ευχαριστίες**

Θα θέλαμε να ευχαριστήσουμε θερμά τον κ. Κουτσούρη Δημήτριο για τη δυνατότητα που μας προσέφερε να εκπονήσουμε την παρούσα διπλωματική εργασία καθώς και τον κ. Γεωργουδέλη Νικόλαο για την πολύτιμη καθοδήγηση και συμβολή του στην πορεία και ολοκλήρωση της εργασίας.





## Περιεχόμενα

1	Εισαγωγή.....	13
1.1	Αντικείμενο Διπλωματικής.....	13
1.1.1	Φαρμακευτική αγωγή.....	13
1.1.2	Διαχείριση ηλεκτρονικού φαρμακείου.....	14
1.1.3	Αποθηκάριος.....	15
1.2	Οργάνωση κειμένου.....	15
2	Διεθνής Ταξινόμηση των Νόσων (ICD).....	16
2.1	Εισαγωγή.....	16
2.2	Σκοπός και εφαρμογές.....	17
2.3	Γενικές αρχές της ταξινόμησης νόσων.....	18
2.4	Βασική δομή και αρχές ταξινόμησης της ICD.....	19
2.4.1	Τόμοι.....	20
2.4.2	Κεφάλαια.....	22
2.4.3	Ενότητες κατηγοριών.....	23
2.4.4	2.4.4 Κατηγορίες τριών χαρακτήρων.....	24
2.4.5	Υποκατηγορίες τεσσάρων χαρακτήρων.....	24
2.4.6	Συμπληρωματικές υποκατηγορίες για χρήση σε επίπεδο πέμπτου ή περαιτέρω χαρακτήρα.....	25
2.4.7	Οι μη χρησιμοποιούμενοι κωδικοί "U".....	25
3	Το Σύστημα ATC.....	26
3.1	Η Ιστορία του ATC.....	26
3.2	Ο Σκοπός του ATC/DDD.....	27
3.3	Η Δομή του ATC.....	28
3.4	Κριτήρια Εισαγωγής στη Λίστα.....	29
3.5	Αρχές Ταξινόμησης.....	29
3.6	Αλλαγές στην Κωδικοποίηση ATC.....	30
4	Ε.Ο.Φ. (Εθνικός Οργανισμός Φαρμάκων)-DrugBank.....	31
4.1	Ε.Ο.Φ. (Εθνικός Οργανισμός Φαρμάκων).....	31
4.1.1	Γενικά.....	31

4.1.2	Λειτουργία.....	32
4.1.3	Δράση.....	34
4.2	DrugBank .....	38
5	Αρχιτεκτονική .NET, η γλώσσα προγραμματισμού C# και ο SQL Server .....	39
5.1	.NET framework.....	39
5.1.1	Εισαγωγή.....	39
5.1.2	Πλεονεκτήματα χρήσης του .NET framework .....	42
5.1.3	Ιστορία του .NET Framework .....	43
5.1.4	LINQ (Language Integrated Query).....	43
5.1.5	Χρήση Master Pages .....	45
5.1.6	Ajax .....	45
5.2	Η γλώσσα προγραμματισμού C#.....	49
5.3	Γλώσσα Μορφοποίησης Ιστοσελίδων Cascading Style Sheets (CSS).....	50
5.4	Microsoft SQL Server .....	50
5.4.1	Αποθήκευση .....	51
5.4.2	Ενδιάμεση μνήμη .....	51
5.4.3	Ταυτοχρονισμός .....	52
5.4.4	Ανάκτηση δεδομένων.....	52
5.4.5	CLR .....	54
5.4.6	Υπηρεσίες.....	54
6	Βάση δεδομένων της εφαρμογής.....	56
6.1	Σχέσεις της βάσης δεδομένων της εφαρμογής .....	56
6.2	Σχεσιακό Μοντέλο .....	65
6.3	Περιεχόμενα της βάσης δεδομένων.....	66
7	Η Ολοκληρωμένη Εφαρμογή Διαχείρισης Ηλεκτρονικού Φαρμακείου .....	67
7.1	Σενάριο σύνδεσης χρήστη στην ιστοσελίδα.....	67
7.1.1	Σελίδα Login.aspx .....	67
7.2	Σενάριο λογαριασμού γιατρού .....	69
7.2.1	Σελίδα DoctorMasterPage.Master .....	69

7.2.2	Οριζόντιο μενού πλοήγησης.....	70
7.2.3	Σελίδα Patients.aspx .....	71
7.2.4	Σελίδα Visit_History.aspx .....	74
7.2.5	Σελίδα DatabaseContents.aspx .....	82
7.2.6	Σελίδα Category.aspx .....	83
7.2.7	Σελίδα ICD.aspx .....	84
7.2.8	Σελίδα Ingredient.aspx .....	85
7.2.9	Σελίδα RouteName.aspx.....	86
7.2.10	Σελίδα RouteType.aspx .....	87
7.2.11	Σελίδα TreeView.aspx.....	88
7.2.12	Σελίδα Orders.aspx .....	89
7.3	Σενάριο λογαριασμού αποθηκάριου.....	90
7.3.1	Σελίδα Warehouse.Master .....	90
7.3.2	Σελίδα Warehouse.aspx.....	91
8	Επίλογος.....	93
8.1	Σύνοψη και συμπεράσματα.....	93
8.2	Μελλοντικές Επεκτάσεις.....	94
9	Βιβλιογραφία.....	95
10	Παράρτημα.....	97

# 1

## *Εισαγωγή*

### *1.1 Αντικείμενο Διπλωματικής*

Αντικείμενο αυτής της διπλωματικής εργασίας είναι ο σχεδιασμός, η ανάπτυξη και υλοποίηση μίας web εφαρμογής για την χορήγηση φαρμακευτικής αγωγής και τη διαχείριση ηλεκτρονικού φαρμακείου. Η εφαρμογή αυτή περιλαμβάνει τρία σκέλη:

#### *1.1.1 Φαρμακευτική αγωγή*

Η εφαρμογή παρέχει στο γιατρό τη δυνατότητα δημιουργίας και διατήρησης ιστορικού ασθενών (περιλαμβάνοντας πληροφορίες για τον ασθενή και τα φαρμακευτικά σκευάσματα που χορηγήθηκαν). Υπάρχει δυνατότητα αναγραφής φαρμακευτικών σκευασμάτων, καταγράφοντας τον τύπο του φαρμάκου, το σκεύασμα, την κύρια δραστική ουσία, την ημερομηνία χορήγησης, την ασθένεια για την οποία χορηγείται, τον αριθμό σκευασμάτων, καθώς και την δοσολογία τους. Η καρτέλα αυτή επικοινωνεί αυτόματα με την βάση διαχείρισης φαρμακείου, ώστε όσα φάρμακα προστίθενται στην καρτέλα του ασθενούς αυτόματα να αφαιρούνται από το απόθεμα του φαρμακείου.

Σχετικά με την καταχώρηση των φαρμάκων που χορηγούνται στον ασθενή, υπάρχει ένας ενσωματωμένος κατάλογος (σύμφωνα με τις προδιαγραφές του Ε.Ο.Φ [9], [18]) που έχει εμπλουτισθεί για να προτείνει διάφορες πιθανές ονομασίες σκευασμάτων με την καταγραφή

των πρώτων γραμμάτων. Το ίδιο γίνεται και στην καταχώρηση των ασθενειών (με βάση την κωδικοποίηση ICD-10 [2], [3], [6], [8]) του ασθενούς. Για κάθε ασθένεια προτείνονται πιθανά φάρμακα διευκολύνοντας τον θεράποντα ιατρό. Επιπρόσθετα ο γιατρός δύναται να κάνει αναζήτηση φαρμάκων με βάση την κύρια δραστική ουσία (σύμφωνα με την πρότυπη κωδικοποίηση ATC [1], [4]), διατηρώντας τη εμπλουτισμένη δυνατότητα να προτείνονται διάφορες πιθανές ονομασίες κύριων δραστικών ουσιών με την καταγραφή των πρώτων γραμμάτων.

### **1.1.2 Διαχείριση ηλεκτρονικού φαρμακείου**

Υπάρχουν δύο γραμμές αποθήκευσης φαρμάκων:

- Την πρώτη γραμμή αποθήκευσης αποτελεί το φαρμακείο (drugstore), το οποίο περιλαμβάνει τα φάρμακα τα οποία είναι άμεσα διαθέσιμα προς χορήγηση στους ασθενείς.
- Την δεύτερη γραμμή αποθήκευσης αποτελεί το container το οποίο περιλαμβάνει τα φάρμακα που παραγγέλλονται ελλείψει επαρκών ποσοτήτων στην πρώτη γραμμή αποθήκευσης.

Η δομή παραλαβής κάθε γραμμής και η προώθηση της στο επόμενο τμήμα είναι τέτοια που καθιστά εφικτό τον έλεγχο. Έχουν οριστεί δικλίδες ασφαλείας στη διαδικασία ροής των φαρμάκων ανάμεσα στις δύο γραμμές αποθήκευσης ώστε να μην δημιουργούνται ασυμφωνίες.

Ο γιατρός δύναται να εποπτεύσει τα φάρμακα ομαδοποιημένα σε κατηγορίες κατά πάθηση, είδος, τύπο, οδό χορήγησης, ATC και δραστική ουσία. Θα μπορεί να γίνεται αναζήτηση με χρήση των παραπάνω φίλτρων.

Τέλος ο γιατρός έχει την δυνατότητα να παραγγέλνει ποσότητες φαρμάκων όταν τα αποθέματα της πρώτης γραμμής κρίνονται ανεπαρκή.

### **1.1.3 Αποθηκάριος**

Είναι υπεύθυνος για την ικανοποίηση των αιτημάτων-παραγγελιών των γιατρών, και επομένως την ομαλή μεταφορά φαρμάκων από την δεύτερη γραμμή αποθήκευσης στην πρώτη.

## **1.2 Οργάνωση κειμένου**

Το **1ο κεφάλαιο** είναι εισαγωγικό και παρουσιάζει το αντικείμενο που πραγματεύεται η διπλωματική εργασία καθώς και την οργάνωση του τόμου.

Στο **2ο κεφάλαιο** περιγράφεται η Διεθνής Ταξινόμηση των Νόσων (ICD)

Στο **3ο κεφάλαιο** περιγράφεται το σύστημα ATC καθώς και η δομή του.

Στο **4ο κεφάλαιο** παρουσιάζεται Ε.Ο.Φ. (Εθνικός Οργανισμός Φαρμάκων), η λειτουργία και δράση του καθώς και η DrugBank [10].

Στο **5ο κεφάλαιο** γίνεται μια παρουσίαση της Αρχιτεκτονικής .NET, της γλώσσας προγραμματισμού C# και του Microsoft SQL Server

Στο **6ο κεφάλαιο** δίνεται αναλυτικά το σχήμα βάσης δεδομένων όπως σχεδιάστηκε για την συγκεκριμένη ιστοσελίδα. Περιγράφονται όλες οι σχέσεις και αναλύονται λεπτομερώς οι ιδιότητές τους. Επίσης δίνεται το σχεσιακό σχήμα της βάσης δεδομένων μας.

Στο **7ο κεφάλαιο** παρουσιάζεται η λειτουργία του site μέσα από παραδείγματα χρήσεως. Επειδή η ιστοσελίδα μας προσφέρει διαφορετικές δυνατότητες και λειτουργίες ανάλογα με το αν ο χρήστης είναι γιατρός ή αποθηκάριος, περιγράφονται αναλυτικά οι δυνατότητες και των δύο τύπων χρηστών.

Το **8ο κεφάλαιο** αποτελεί μια σύνοψη της παρούσας διπλωματικής εργασίας και παρατίθενται συμπεράσματα και πιθανές μελλοντικές επεκτάσεις του συστήματος που υλοποιήθηκε.

Στο **9ο κεφάλαιο** γίνεται μια παρουσίαση της βιβλιογραφίας που χρησιμοποιήθηκε κατά την εκπόνηση του τόμου και της εφαρμογής.

Στο **10ο κεφάλαιο** είναι το παράρτημα της παρούσας εργασίας όπου παρατίθεται ο κώδικας που υλοποιήθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας.

# 2

## *Διεθνής Ταξινόμηση των Νόσων (ICD)*

### *2.1 Εισαγωγή*

Η κωδικοποίηση κατά ICD ξεκίνησε το 1901 στην έκδοση 1 ενώ σήμερα βρίσκεται σε χρήση η έκδοση 10 η οποία οριστικοποιήθηκε στο 1992.

Το ICD-10 εγκρίθηκε από την τεσσαρακοστή τρίτη Παγκόσμια Συνέλευση Υγείας το Μάιο του 1990 και τέθηκε σε χρήση στα κράτη μέλη που ανήκουν στον Π.Ο.Υ. (Παγκόσμια Οργάνωση Υγείας) από το 1994. Η ταξινόμηση αυτή είναι η τελευταία σε μια σειρά που έχει τις ρίζες της στη δεκαετία του 1850. Η πρώτη έκδοση, που είναι γνωστή ως International List of Causes of Death, υιοθετήθηκε από τη Διεθνή Στατιστική Υπηρεσία το 1893. Κατά τη δημιουργία του το 1948, ο Π.Ο.Υ. ανέλαβε την ευθύνη για το ICD, όταν δημοσιεύτηκε η έκτη Αναθεώρηση, η οποία περιελάμβανε αιτίες νοσηρότητας για πρώτη φορά [8].

Το ICD είναι το διεθνές διαγνωστικό πρότυπο ταξινόμησης για όλα τα γενικά επιδημιολογικά, πολλούς υγειοκεντρικούς σκοπούς και κλινική χρήση. Αυτά περιλαμβάνουν την ανάλυση της γενικής κατάστασης της υγείας ομάδων του πληθυσμού και την παρακολούθηση της συχνότητας εμφάνισης και της εξάπλωσης των ασθενειών και άλλα προβλήματα.



Εκτός όμως της διεθνούς διάστασης των ωφελειών από τη χρήση της ICD-10, υπάρχουν και οφέλη τα οποία σχετίζονται με την επιδημιολογική παρακολούθηση της επίπτωσης και του επιπολασμού των νόσων και άλλων συναφών προβλημάτων υγείας, και την παραγωγή αξιόπιστων στατιστικών στοιχείων τόσο σε εθνικό επίπεδο, όσο και σε περιφερειακό, τοπικό, αλλά και σε επίπεδο μονάδας υγείας. Στην καθημερινή κλινική πράξη, η ICD-10 μπορεί να χρησιμοποιηθεί στον ιατρικό φάκελο, στα πιστοποιητικά θανάτου, για την έκδοση κατευθυντηρίων οδηγιών και για την έρευνα. Τέλος, συμβάλλει αποτελεσματικά στην αποδοτικότερη λειτουργία του τομέα υγείας αφού διευκολύνει θέματα διοίκησης και διαχείρισης υπηρεσιών υγείας, όπως την αποζημίωση των ιατρικών υπηρεσιών και την παρακολούθηση και ανάλυση των δαπανών υγείας, διευκολύνοντας το σχεδιασμό και την κατανομή των πόρων με τέτοιο τρόπο ώστε να βελτιώνονται τόσο η ποιότητα όσο και η αποδοτικότητα των υπηρεσιών υγείας.

## **2.2 Σκοπός και εφαρμογές**

Η ταξινόμηση των νοσημάτων μπορεί να οριστεί ως ένα σύστημα κατηγοριών στις οποίες είναι καταχωρημένες οι παθολογικές οντότητες σύμφωνα με προκαθορισμένα κριτήρια. Ο σκοπός της ICD είναι να επιτρέψει τη συστηματική ανάλυση των καταγραφών, την ερμηνεία και τη σύγκριση των δεδομένων θνησιμότητας και νοσηρότητας που έχουν συγκεντρωθεί από διάφορες χώρες ή περιοχές και σε διαφορετικές χρονικές περιόδους. Η ICD χρησιμοποιείται για να μετατρέψει τις διαγνώσεις νόσων και άλλων προβλημάτων υγείας από λέξεις σε έναν αλφαριθμικό κωδικό που θα επιτρέψει την εύκολη αποθήκευση, ανάκτηση και ανάλυση των δεδομένων.

Στην πράξη, η ICD έχει γίνει το διεθνές πρότυπο ταξινόμησης διαγνώσεων για όλους τους γενικούς επιδημιολογικούς σκοπούς και για πολλούς σκοπούς που αφορούν στη διαχείριση της υγείας. Αυτοί περιλαμβάνουν την ανάλυση της γενικής κατάστασης της υγείας των πληθυσμιακών ομάδων και την παρακολούθηση της επίπτωσης και του επιπολασμού των νόσων και άλλων προβλημάτων υγείας σε σχέση με άλλες παραμέτρους, όπως τα χαρακτηριστικά και οι συνθήκες των προσβληθέντων ατόμων. Η ICD ούτε προορίζεται ούτε είναι κατάλληλη για τη σύνταξη ευρετηρίου συγκεκριμένων κλινικών οντοτήτων. Υπάρχουν επίσης μερικοί περιορισμοί στη χρήση της ICD που αφορούν στις μελέτες οικονομικής φύσης, όπως η κοστολόγηση ή η κατανομή των πόρων.

Η ICD μπορεί να χρησιμοποιηθεί για την ταξινόμηση νοσημάτων και άλλων προβλημάτων υγείας καταχωρημένων σε πολλούς τύπους αρχείων υγείας και φυσικής κίνησης πληθυσμού.

Η αρχική της χρήση ήταν για την ταξινόμηση των αιτίων θνησιμότητας όπως αυτά καταγράφονται στο πιστοποιητικό θανάτου. Αργότερα, το πεδίο της διευρύνθηκε για να περιλάβει διαγνώσεις νοσηρότητας. Είναι σημαντικό να σημειωθεί ότι, ενώ η ICD έχει σχεδιαστεί κυρίως για την ταξινόμηση νοσημάτων και τραυματισμών που ταυτοποιούνται με μια επίσημη διάγνωση, δεν μπορεί να ταξινομηθεί με αυτό τον τρόπο κάθε πρόβλημα ή αιτία που οδηγεί σε επαφή με τις υγειονομικές υπηρεσίες. Γι' αυτό, η ICD καλύπτει μεγάλη ποικιλία σημείων, συμπτωμάτων, παθολογικών ευρημάτων, παραπόνων και κοινωνικών συνθηκών που μπορεί να χρησιμοποιούνται αντί για τη διάγνωση σε ιατρικά ιστορικά ή ιατρικούς φακέλους. Συνεπώς, μπορεί να χρησιμοποιηθεί για την ταξινόμηση δεδομένων που έχουν καταχωρηθεί κάτω από τίτλους όπως "διάγνωση", "αιτία εισαγωγής", "καταστάσεις που αντιμετωπίστηκαν" και "αιτία επίσκεψης", οι οποίοι εμφανίζονται σε ένα ευρύ φάσμα αρχείων υγείας από τα οποία παράγονται στατιστικές και άλλες πληροφορίες που σχετίζονται με την κατάσταση της υγείας.

Παρόλο που η ICD είναι κατάλληλη για πολλές διαφορετικές εφαρμογές, δεν επιτρέπει πάντα την καταχώρηση επαρκών λεπτομερειών για ορισμένες ειδικότητες, και μερικές φορές ίσως να χρειάζονται πληροφορίες σχετικές με τα διαφορετικά χαρακτηριστικά των ταξινομημένων νοσολογικών καταστάσεων. Έχει επίσης προταθεί ότι η ICD πρέπει να περιλαμβάνει ταξινομήσεις επιπρόσθετων πληροφοριών σχετικών με τη κατάσταση υγείας ή τη φροντίδα υγείας.

### ***2.3 Γενικές αρχές της ταξινόμησης νόσων***

Όπως δήλωσε ο William Farr το 1856:

Η ταξινόμηση είναι μια μέθοδος γενίκευσης. Επομένως, πολλές ταξινομήσεις είναι δυνατό να χρησιμοποιηθούν επωφελώς και ο γιατρός, ο παθολογοανατόμος, ή ο ιατροδικαστής, καθένας από την πλευρά του, μπορεί νόμιμα να ταξινομεί τα νοσήματα και τις αιτίες θανάτου με τον τρόπο που θεωρεί καλύτερα προσαρμοσμένο ώστε να διευκολύνει τις έρευνές του, και να εξάγει γενικά συμπεράσματα [7].

Μια στατιστική ταξινόμηση των νόσων πρέπει να περιέχει ένα περιορισμένο αριθμό κατηγοριών που αποκλείονται αμοιβαία, ικανών να συμπεριλάβουν ολόκληρο το φάσμα των νοσολογικών καταστάσεων. Οι κατηγορίες θα πρέπει να επιλέγονται έτσι ώστε να διευκολύνουν τη στατιστική μελέτη των νοσολογικών φαινομένων. Μια συγκεκριμένη νοσολογική οντότητα που έχει ιδιαίτερη σημασία για τη δημόσια υγεία ή που απαντάται

συχνά θα πρέπει να έχει τη δική της κατηγορία. Σε αντίθετη περίπτωση, οι κατηγορίες θα καταχωρούνται σε ομάδες ξεχωριστών αλλά συναφών καταστάσεων. Κάθε νόσημα ή νοσολογική κατάσταση πρέπει να κατέχει μια σαφώς προσδιορισμένη θέση στον κατάλογο των κατηγοριών. Κατά συνέπεια, από την αρχή ως το τέλος της ταξινόμησης θα υπάρχουν υπολειπόμενες κατηγορίες που παραμένουν ανοιχτές για διάφορες καταστάσεις που δεν είναι δυνατόν να καταχωρηθούν στις πιο ειδικές κατηγορίες. Σε όλες αυτές τις υπολειπόμενες κατηγορίες πρέπει να ταξινομούνται όσο το δυνατό λιγότερες νοσολογικές καταστάσεις.

Το στοιχείο της ομαδοποίησης είναι αυτό που διακρίνει τη στατιστική ταξινόμηση από την ονοματολογία, η οποία πρέπει να έχει ένα διαφορετικό τίτλο για κάθε γνωστή νοσολογική κατάσταση. Πάντως, οι έννοιες της ταξινόμησης και της ονοματολογίας σχετίζονται στενά γιατί πολύ συχνά η ονοματολογία είναι καταχωρημένη με συστηματικό τρόπο.

Μια στατιστική ταξινόμηση μπορεί να επιτρέπει διαφορετικά επίπεδα λεπτομέρειας αν διαθέτει μια ιεραρχική δομή με υποδιαιρέσεις. Μια στατιστική ταξινόμηση νοσημάτων θα πρέπει να διατηρεί την ικανότητα τόσο στο να προσδιορίζει συγκεκριμένες νοσολογικές οντότητες όσο και στο να επιτρέπει τη στατιστική παρουσίαση των δεδομένων για ευρύτερες ομάδες, έτσι ώστε να παρέχει τη δυνατότητα εξαγωγής χρήσιμων και κατανοητών πληροφοριών.

Οι ίδιες γενικές αρχές μπορούν να εφαρμοστούν στην ταξινόμηση άλλων προβλημάτων υγείας και αιτιών επαφής με τις υγειονομικές υπηρεσίες, που είναι επίσης ενσωματωμένες στην ICD.

Η ICD έχει εξελιχθεί ως μια πρακτική, και όχι αμιγώς θεωρητική ταξινόμηση, στην οποία έχουν γίνει ορισμένοι συμβιβασμοί μεταξύ των ταξινομήσεων που βασίζονται στην αιτιολογία, την ανατομική εντόπιση, τις συνθήκες έναρξης της νόσου, κτλ. Έχουν γίνει επίσης προσαρμογές προκειμένου να καλυφθεί η ποικιλία των στατιστικών εφαρμογών για τις οποίες προορίζεται η ICD, όπως η θνησιμότητα, η νοσηρότητα, η κοινωνική ασφάλιση και άλλα είδη στατιστικών και μελετών υγείας.

## **2.4 Βασική δομή και αρχές ταξινόμησης της ICD**

Η ICD είναι ταξινόμηση μεταβλητού άξονα. Η δομή διαμορφώθηκε με γνώμονα την αρχική πρόταση του William Farr σε παλαιότερες διεθνείς συζητήσεις σχετικές με τη δομή της ταξινόμησης. Σύμφωνα με το πρότυπο του Farr, για όλους τους πρακτικούς και επιδημιολογικούς σκοπούς, τα στατιστικά δεδομένα των νοσημάτων θα πρέπει να ομαδοποιούνται με τον ακόλουθο τρόπο [6]:

- επιδημικά νοσήματα (λοιμώδη)
- ιδιοσυστατικά ή γενικά νοσήματα τοπικά νοσήματα
- καταχωρημένα ανάλογα με την ανατομική εντόπιση
- διαταραχές ανάπτυξης
- τραυματισμοί

Αυτό το μοντέλο απαντάται στα κεφάλαια της ICD-10. Έχει ανταπεξέλθει στη δοκιμασία του χρόνου και, αν και κατά κάποιο τρόπο αυθαίρετο, θεωρείται ακόμη η πιο εύχρηστη δομή για γενικούς επιδημιολογικούς σκοπούς από όλα τα εναλλακτικά μοντέλα που εξετάστηκαν.

Οι δύο πρώτες και οι δύο τελευταίες από τις ομάδες που αναφέρονται παραπάνω, αποτελούν «ειδικές ομάδες» που ενοποιούν τις νοσολογικές καταστάσεις οι οποίες αν ήταν διασκορπισμένες θα ήταν ακατάλληλα καταχωρημένες για επιδημιολογική μελέτη, όπως για παράδειγμα σε μια ταξινόμηση με βάση κυρίως την ανατομική εντόπιση. Η ομάδα που απομένει, δηλαδή τα «τοπικά νοσήματα ανάλογα με την ανατομική εντόπιση», περιλαμβάνει τα κεφάλαια της ICD που αφορούν καθένα από τα κύρια συστήματα του οργανισμού.

Η διάκριση μεταξύ των κεφαλαίων των «ειδικών ομάδων» και των κεφαλαίων των «οργανικών συστημάτων» έχει πρακτική σημασία για την κατανόηση της δομής της ταξινόμησης, για την κωδικοποίησή της και για την ερμηνεία των στατιστικών που βασίζονται σε αυτή. Θα πρέπει να έχουμε υπόψη ότι, γενικά, οι νοσολογικές καταστάσεις ταξινομούνται κυρίως σε ένα από τα κεφάλαια των «ειδικών ομάδων». Όταν υπάρχει κάποια αμφιβολία σχετικά με την ορθή τοποθέτηση μιας νοσολογικής κατάστασης, τα κεφάλαια των «ειδικών ομάδων» θα πρέπει να έχουν προτεραιότητα.

Η βασική ICD αποτελεί έναν απλό κωδικοποιημένο κατάλογο κατηγοριών τριών χαρακτήρων, καθεμιά από τις οποίες μπορεί να διαιρεθεί περαιτέρω μέχρι και σε δέκα υποκατηγορίες τεσσάρων χαρακτήρων. Στη θέση του καθαρά αριθμητικού συστήματος κωδικοποίησης των προηγούμενων αναθεωρήσεων, η Δέκατη Αναθεώρηση χρησιμοποιεί ένα αλφαριθμητικό κωδικό ο οποίος έχει ένα γράμμα στην πρώτη θέση και έναν αριθμό στη δεύτερη, τρίτη και τέταρτη θέση. Ο τέταρτος χαρακτήρας ακολουθεί μετά από τελεία, σαν να αποτελεί δεκαδικό ψηφίο. Επομένως, μπορούν να υπάρχουν κωδικοί αριθμοί που κυμαίνονται από A00.0 μέχρι Z99.9. Το γράμμα U δε χρησιμοποιείται (βλέπε 2.4.7).

#### **2.4.1 Τόμοι**

Η ICD-10 περιλαμβάνει τρεις τόμους: ο Τόμος 1 περιέχει τις κύριες ταξινομήσεις· ο Τόμος 2 παρέχει οδηγίες στους χρήστες της ICD· και ο Τόμος 3 είναι το Αλφαβητικό Ευρετήριο της ταξινόμησης.

Το μεγαλύτερο μέρος του Τόμου 1 καταλαμβάνει η κύρια ταξινόμηση, που αποτελείται από τον κατάλογο των κατηγοριών τριών χαρακτήρων, τον κατάλογο κωδικών των όρων που περιέχονται και τις υποκατηγορίες τεσσάρων χαρακτήρων. Η «βασική» ταξινόμηση - ο κατάλογος των κατηγοριών τριών χαρακτήρων (Τόμος 1) αποτελεί το υποχρεωτικό επίπεδο για τις αναφορές στη βάση δεδομένων θνησιμότητας του Π.Ο.Υ. και για γενικές διεθνείς συγκρίσεις. Η βασική ταξινόμηση περιέχει επίσης κατάλογο τίτλων των κεφαλαίων και των ενοτήτων. Ο κατάλογος κωδικών, που δίνει πλήρεις λεπτομέρειες σε επίπεδο τεσσάρων χαρακτήρων, διαιρείται σε 21 κεφάλαια.

Ο Τόμος 1 περιέχει επίσης τα ακόλουθα:

- **Μορφολογία νεοπλασμάτων.** Η ταξινόμηση της μορφολογίας των νεοπλασμάτων μπορεί να χρησιμοποιηθεί, αν είναι επιθυμητό, ως ένας πρόσθετος κωδικός για την ταξινόμηση του μορφολογικού (ιστολογικού) τύπου των νεοπλασμάτων τα οποία, με λίγες εξαιρέσεις, είναι ταξινομημένα στο Κεφάλαιο II μόνο με βάση τη συμπεριφορά και την εντόπιση (τοπογραφία).
- **Ειδικοί συνοπτικοί πίνακες.** Επειδή ο πλήρης κατάλογος τεσσάρων χαρακτήρων της ICD, και ακόμη και ο κατάλογος τριών χαρακτήρων, είναι πολύ εκτενείς για να αναγράφονται σε κάθε στατιστικό πίνακα, οι περισσότερες στατιστικές ρουτίνες χρησιμοποιούν ένα συνοπτικό πίνακα που δίνει έμφαση σε ορισμένες μονήρεις καταστάσεις και ομαδοποιεί κάποιες άλλες. Οι τέσσερις ειδικοί πίνακες για την κωδικοποίηση της θνησιμότητας αποτελούν αναπόσπαστο τμήμα της ICD. Οι πίνακες 1 και 2 αφορούν τη γενική θνησιμότητα και οι πίνακες 3 και 4 είναι για τη βρεφική και την παιδική θνησιμότητα (ηλικία 0 - 4 ετών). Υπάρχει ακόμη ένας ειδικός συνοπτικός πίνακας για τη νοσηρότητα. Οι πίνακες αυτοί βρίσκονται στον Τόμο 1.
- **Ορισμοί.** Οι ορισμοί του Τόμου 1 έχουν υιοθετηθεί από την Παγκόσμια Συνέλευση Υγείας και περιλαμβάνονται με σκοπό να διευκολύνουν τη διεθνή συγκρισιμότητα των δεδομένων.
- **Κανόνες ονοματολογίας.** Οι κανόνες που υιοθετήθηκαν από τη Παγκόσμια Συνέλευση Υγείας παραθέτουν τις επίσημες υποχρεώσεις των Κρατών-Μελών του Π.Ο.Υ., όσον αφορά στην ταξινόμηση των νόσων και αιτιών θανάτου, και τη συλλογή και δημοσίευση στατιστικών. Βρίσκονται στον Τόμο 1.

## 2.4.2 Κεφάλαια

Η ταξινόμηση διαιρείται σε 21 κεφάλαια. Ο πρώτος χαρακτήρας του κωδικού της ICD είναι ένα γράμμα, και κάθε γράμμα σχετίζεται με ένα συγκεκριμένο κεφάλαιο, εκτός από το γράμμα D που χρησιμοποιείται τόσο στο Κεφάλαιο II Νεοπλάσματα, όσο και στο Κεφάλαιο III Νόσοι του αίματος και των αιμοποιητικών οργάνων και ορισμένες διαταραχές που εμπλέκουν τον ανοσολογικό μηχανισμό, και το γράμμα H που χρησιμοποιείται τόσο στο Κεφάλαιο VII, Νοσήματα του οφθαλμού και των εξαρτημάτων του όσο και στο Κεφάλαιο VIII Παθήσεις του ωτός και της μαστοειδούς απόφυσης. Τέσσερα κεφάλαια (τα Κεφάλαια I, II, XIX και XX) χρησιμοποιούν περισσότερα από ένα γράμμα στην πρώτη θέση των κωδικών τους.

Κάθε κεφάλαιο περιέχει αριθμό κατηγοριών τριών χαρακτήρων ικανό να καλύψει τα περιεχόμενά του· δε χρησιμοποιούνται όλοι οι διαθέσιμοι κωδικοί, αφήνοντας χώρο για μελλοντική αναθεώρηση και επέκταση.

Τα Κεφάλαια I έως XVII σχετίζονται με νοσήματα και άλλες νοσολογικές καταστάσεις, και το Κεφάλαιο XIX με τραυματισμούς, δηλητηριάσεις και ορισμένες άλλες συνέπειες εξωτερικών αιτιών. Τα υπόλοιπα κεφάλαια συμπληρώνουν το φάσμα των θεμάτων της ύλης που περιλαμβάνεται σήμερα στα διαγνωστικά δεδομένα. Το Κεφάλαιο XVIII καλύπτει τα Συμπτώματα, σημεία και παθολογικά κλινικά και εργαστηριακά ευρήματα, που δεν είναι ταξινομημένα κάπου αλλού. Το Κεφάλαιο XX, Εξωτερικές αιτίες νοσηρότητας και θνησιμότητας, χρησιμοποιήθηκε παραδοσιακά για την ταξινόμηση των αιτιών των τραυματισμών και δηλητηριάσεων, αλλά, από την Ένατη Αναθεώρηση και μετά, καλύπτει και όλες τις καταγεγραμμένες εξωτερικές αιτίες νόσων και άλλων νοσολογικών καταστάσεων. Τέλος, το Κεφάλαιο XXI, Παράγοντες που επηρεάζουν τη κατάσταση υγείας και την επαφή με τις υγειονομικές υπηρεσίες, προορίζεται για την ταξινόμηση των δεδομένων που επεξηγούν την αιτία επαφής με τις υπηρεσίες υγείας ενός ατόμου το οποίο δεν ασθενεί στην παρούσα φάση, ή τις συνθήκες υπό τις οποίες γίνεται η περίθαλψη του ασθενούς στο συγκεκριμένο χρονικό διάστημα ή τις συνθήκες που έχουν κάποια άλλη επίδραση στην περίθαλψη αυτού του ατόμου.

<b>International Statistical Classification of Diseases and Related Health Problems 10th Revision</b>		
<b>Κεφάλαια</b>	<b>Ενότητες</b>	<b>Τίτλος</b>
I	A00-B99	Certain infectious and parasitic diseases

II	C00-D48	Neoplasms
III	D50-D89	Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
IV	E00-E90	Endocrine, nutritional and metabolic diseases
V	F00-F99	Mental and behavioural disorders
VI	G00-G99	Diseases of the nervous system
VII	H00-H59	Diseases of the eye and adnexa
VIII	H60-H95	Diseases of the ear and mastoid process
IX	I00-I99	Diseases of the circulatory system
X	J00-J99	Diseases of the respiratory system
XI	K00-K93	Diseases of the digestive system
XII	L00-L99	Diseases of the skin and subcutaneous tissue
XIII	M00-M99	Diseases of the musculoskeletal system and connective tissue
XIV	N00-N99	Diseases of the genitourinary system
XV	O00-O99	Pregnancy, childbirth and the puerperium
XVI	P00-P96	Certain conditions originating in the perinatal period
XVII	Q00-Q99	Congenital malformations, deformations and chromosomal abnormalities
XVIII	R00-R99	Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
XIX	S00-T98	Injury, poisoning and certain other consequences of external causes
XX	V01-Y98	External causes of morbidity and mortality
XXI	Z00-Z99	Factors influencing health status and contact with health services
XXII	U00-U99	Codes for special purposes

### 2.4.3 *Ενότητες κατηγοριών*

Τα κεφάλαια υποδιαιρούνται σε ομοιογενείς ενότητες («blocks») κατηγοριών τριών χαρακτήρων. Στο Κεφάλαιο I, οι τίτλοι των ενοτήτων αντιπροσωπεύουν δύο άξονες

ταξινόμησης – τον τρόπο μετάδοσης και την ευρύτερη ομάδα των λοιμογόνων οργανισμών. Στο Κεφάλαιο II, ο πρώτος άξονας αφορά στη συμπεριφορά του νεοπλάσματος εντός του φάσματος της συμπεριφοράς, ο άξονας αφορά κυρίως στην εντόπιση, αν και μερικές κατηγορίες τριών χαρακτήρων αφορούν σε σημαντικούς μορφολογικούς τύπους (π.χ. λευχαιμίες, λεμφώματα, μελανώματα, μεσοθηλιώματα, σάρκωμα Kaposi). Το φάσμα των κατηγοριών αναγράφεται μέσα σε παρένθεση μετά από τον τίτλο κάθε ενότητας κατηγοριών.

#### **2.4.4 2.4.4 Κατηγορίες τριών χαρακτήρων**

Μέσα σε κάθε ενότητα, μερικές από τις κατηγορίες τριών χαρακτήρων αφορούν σε μοναδικές καταστάσεις, που έχουν επιλεγεί εξ αιτίας της συχνότητάς τους, της βαρύτητας ή της επιδεκτικότητάς τους σε παρεμβάσεις της δημόσιας υγείας, ενώ άλλες χρησιμοποιούνται για ομάδες νοσημάτων με κάποια κοινά χαρακτηριστικά. Συνήθως υπάρχει πρόβλεψη για την ταξινόμηση «άλλων» καταστάσεων, το οποίο επιτρέπει να συμπεριληφθούν πολλές διαφορετικές, αλλά πιο σπάνιες καταστάσεις, καθώς και μερικές «μη καθορισμένες» καταστάσεις.

#### **2.4.5 Υποκατηγορίες τεσσάρων χαρακτήρων**

Παρόλο που δεν είναι υποχρεωτικό για την αναφορά σε διεθνές επίπεδο, οι περισσότερες κατηγορίες τριών χαρακτήρων υποδιαιρούνται μέσω ενός τέταρτου αριθμητικού δεκαδικού ψηφίου μέχρι και σε δέκα υποκατηγορίες. Όπου μια κατηγορία τριών χαρακτήρων δεν υποδιαιρείται, συνιστάται η χρήση του γράμματος «X» για τη συμπλήρωση της τέταρτης θέσης, έτσι ώστε οι κωδικοί να έχουν ένα σταθερό μήκος, αναγκαίο για την επεξεργασία των δεδομένων.

Οι υποκατηγορίες τεσσάρων χαρακτήρων χρησιμοποιούνται κάθε φορά με τον πιο κατάλληλο τρόπο, καθορίζοντας, για παράδειγμα, διαφορετικές εντοπίσεις ή παραλλαγές, στην περίπτωση που η κατηγορία τριών χαρακτήρων αφορά μια μοναδική νόσο ή ξεχωριστά νοσήματα αν η κατηγορία τριών χαρακτήρων αντιπροσωπεύει ομάδα νοσολογικών καταστάσεων.

Ο τέταρτος χαρακτήρας .8 χρησιμοποιείται κατά κανόνα για «άλλες» νοσολογικές καταστάσεις που ανήκουν στην κατηγορία τριών χαρακτήρων, και ο χαρακτήρας .9



χρησιμοποιείται κυρίως για να αποδώσει την ίδια έννοια που έχει και ο τίτλος της κατηγορίας τριών χαρακτήρων, χωρίς να προσθέτει κάποια επιπλέον πληροφορία.

Όταν οι ίδιες υποδιαιρέσεις τεσσάρων χαρακτήρων εφαρμόζονται σε μια σειρά κατηγοριών τριών χαρακτήρων, αναφέρονται μόνο μια φορά στην αρχή της σειράς. Σε καθεμιά από τις σχετικές κατηγορίες υπάρχει σημείωση που παραπέμπει στην ανεύρεση λεπτομερειών.

#### **2.4.6 Συμπληρωματικές υποκατηγορίες για χρήση σε επίπεδο πέμπτου ή περαιτέρω χαρακτήρα**

Τα επίπεδα των πέντε ή περισσότερων χαρακτήρων αποτελούν συνήθως υποταξινομήσεις κατά μήκος ενός διαφορετικού άξονα από αυτόν με τέσσερις χαρακτήρες. Απαντώνται στο:

Κεφάλαιο XIII- υποδιαιρέσεις με βάση την ανατομική εντόπιση

Κεφάλαιο XIX- υποδιαιρέσεις που χαρακτηρίζουν ανοικτά και κλειστά κατάγματα, καθώς και ενδοκρανιακούς, ενδοθωρακικούς και ενδοκοιλιακούς τραυματισμούς με και χωρίς ανοικτό τραύμα

Κεφάλαιο XX- υποδιαιρέσεις που χαρακτηρίζουν το είδος της δραστηριότητας που λάμβανε χώρα κατά την ώρα του συμβάντος.

#### **2.4.7 Οι μη χρησιμοποιούμενοι κωδικοί "U"**

Οι κωδικοί U00 – U49 προορίζονται για χρήση για την προσωρινή καταχώρηση νέων νοσημάτων αμφίβολης αιτιολογίας. Οι κωδικοί U50 - U99 μπορούν να χρησιμοποιηθούν στην έρευνα, π.χ. κατά τον έλεγχο μιας εναλλακτικής υποταξινόμησης για ένα ειδικό πρόγραμμα.

# 3

## *Το Σύστημα ATC*

### *3.1 Η Ιστορία του ATC*

Το ενδιαφέρον για την κωδικοποίηση στον κλάδο των φαρμάκων ξεκίνησε από τη δεκαετία του '60. Το 1969, σε συμπόσιο με τίτλο «Η κατανάλωση φαρμάκων» που έγινε στο Όσλο, συμφωνήθηκε πως πρέπει να εισαχθεί ένα διεθνώς αποδεκτό σύστημα κωδικοποίησης, το οποίο θα βοηθούσε στον τομέα της έρευνας για τη χρήση των φαρμάκων.

Τροποποιώντας και επεκτείνοντας την κωδικοποίηση European Pharmaceutical Market Research Association (EPHRA), Νορβηγοί ερευνητές ανέπτυξαν το σύστημα κωδικοποίησης ATC (Anatomical Therapeutic Chemical).

Όμως, για να μετρηθεί η χρήση των φαρμάκων είναι πολύ σημαντικό να υπάρχει, εκτός από σύστημα κωδικοποίησης, μονάδα μέτρησης. Λόγω του ότι υπήρχαν πολλές διαφωνίες για τις παραδοσιακές μονάδες, αναπτύχθηκε μια τεχνική μονάδα μέτρησης, η οποία ονομάστηκε DDD (Defined Daily Dose).

Το Σκανδιναβικό Ινστιτούτο Φαρμάκων (NLN), το οποίο ιδρύθηκε το 1975, συνεργάστηκε με Νορβηγούς ερευνητές για να αναπτύξουν το σύστημα ATC/DDD. Το NLN δημοσίευσε τη Σκανδιναβική Στατιστική Φαρμακολογίας το 1976 χρησιμοποιώντας για πρώτη φορά το ATC/DDD. Τότε ήταν που η συγκεκριμένη κωδικοποίηση κέρδισε το παγκόσμιο ενδιαφέρον.

Το 1981 το Ευρωπαϊκό τμήμα της Παγκόσμιας Οργάνωσης Υγείας (WHO) γνωμοδότησε θετικά υπέρ της διεθνούς χρήσης του ATC/DDD. Σε συνδυασμό με αυτή την εξέλιξη, αλλά και για τη διάδοση αυτής της κωδικοποίησης, η WHO προχώρησε στην ίδρυση του WHO Collaborating Centre for Drug Statistics Methodology, το οποίο ιδρύθηκε το 1982 για να έχει την κεντρική ευθύνη και να κατευθύνει τη χρήση του ATC/DDD. Το συγκεκριμένο κέντρο βρίσκεται στο Όσλο, στο Νορβηγικό Ινστιτούτο Δημόσιας Υγείας, και χρηματοδοτείται από τη Νορβηγική κυβέρνηση.

Το 1996 η WHO διέκρινε την ανάγκη να προωθήσει τη χρήση του ATC/DDD ως διεθνές πρότυπο για τις φαρμακολογικές μελέτες. Έτσι, το κέντρο του Όσλο συνδέθηκε απευθείας με το κεντρικό γραφείο της WHO στη Γενεύη, αντί για το Ευρωπαϊκό γραφείο της WHO στην Κοπεγχάγη. Αυτό κρίθηκε σκόπιμο για την ενοποίηση των φαρμακολογικών ερευνών, αλλά και για το στόχο της WHO για παγκόσμια πρόσβαση αλλά και για λελογισμένη χρήση των φαρμάκων, ιδίως στις αναπτυσσόμενες χώρες. Άλλωστε, η πρόσβαση σε έγκυρες και κωδικοποιημένες πληροφορίες είναι σημαντική για τη δημιουργία προτύπων για τη χρήση φαρμάκων, την αναγνώριση των προβλημάτων, εκπαιδευτικές παρεμβάσεις και την παρακολούθηση των συμπερασμάτων που προκύπτουν.

Όταν λήφθηκε η απόφαση της διεθνοποίησης του συστήματος ATC/DDD, η WHO Division of Drug Management and Policies δημιούργησε την WHO International Working Group for Drug Statistics Methodology. Η τελευταία δρα ως εξειδικευμένη σύμβουλος του WHO Collaborating Centre for Drug Statistics Methodology.

### **3.2 Ο Σκοπός του ATC/DDD**

Ο σκοπός του συστήματος ATC/DDD είναι να εξυπηρετεί ως εργαλείο για τη μελέτη της χρήσης των φαρμάκων ώστε να βελτιστοποιηθεί η χρήση τους. Μια συνιστώσα αυτού του στόχου είναι η παρουσίαση στατιστικών στοιχείων για την κατανάλωση φαρμάκων σε διεθνές επίπεδο.

Πρωταρχικό μέλημα των υπευθύνων του Κέντρου και της Ομάδας Εργασίας που αναφέρθηκε νωρίτερα είναι η διατήρηση σταθερών κωδικών ATC και DDD σε βάθος χρόνου, έτσι ώστε να μελετηθούν οι τάσεις της χρήσης φαρμάκων, χωρίς τη δυσκολία που θα προκαλούσαν συνεχείς αλλαγές στο σύστημα. Γι' αυτόν το λόγο οι αλλαγές στην κωδικοποίηση, χωρίς σοβαρή αιτία σχετική με την κατανάλωση φαρμάκων, γίνονται δύσκολα. Έτσι, το σύστημα ATC/DDD δεν είναι κατάλληλο για λήψη αποφάσεων σχετικά με τιμολόγηση και αντικατάσταση φαρμάκων .

Στην ιστοσελίδα της η WHO αναφέρει : «η ταξινόμηση μιας ουσίας στο σύστημα ATC/DDD δεν αποτελεί οδηγία για χρήση της, ούτε συνεπάγεται κρίση για την αποτελεσματικότητα ή τη σχετική αποτελεσματικότητα φαρμάκων ή ομάδων φαρμάκων»[5].

### 3.3 Η Δομή του ATC

Το σύστημα ATC έχει δενδρική δομή πέντε επιπέδων. Το πρώτο επίπεδο αφορά στο ανατομικό σύστημα ή στην πάθηση που δρα το φάρμακο (ένας χαρακτήρας), το δεύτερο επίπεδο αφορά στην θεραπευτική κατηγορία (δύο αριθμητικοί χαρακτήρες), το τρίτο στη φαρμακολογική κατηγορία (ένας χαρακτήρας) και το πέμπτο στη δραστική ουσία (δύο αριθμητικοί χαρακτήρες).

Η ταξινόμηση των φαρμακευτικών προϊόντων στους πίνακες που ακολουθούν αναφέρεται στο πρώτο επίπεδο και έχει ως εξής [18]:

- A ALIMENTARY TRACT AND METABOLISM
- B BLOOD AND BLOOD FORMING ORGANS
- C CARDIOVASCULAR SYSTEM
- D DERMATOLOGICALS
- G GENITO URINARY SYSTEM AND SEX HORMONES
- H SYSTEMIC HORMONAL PREPARATIONS, EXCL. SEX HORMONES AND INSULINS
- J ANTIINFECTIVES FOR SYSTEMIC USE
- L ANTINEOPLASTIC AND IMMUNOMODULATING AGENTS
- M MUSCULO-SKELETAL SYSTEM
- N NERVOUS SYSTEM
- P ANTIPARASITIC PRODUCTS, INSECTICIDES AND REPELLENTS
- R RESPIRATORY SYSTEM
- S SENSORY ORGANS
- V VARIOUS

Ακολουθεί παράδειγμα της θεραπευτικής ουσίας metformin :

A	Alimentary tract and metabolism
---	---------------------------------

	(1st level, anatomical main group)
A10	Drugs used in diabetes (2nd level, therapeutic subgroup)
A10B	Blood glucose lowering drugs, excl. insulins (3rd level, pharmacological subgroup)
A10BA	Biguanides (4th level, chemical subgroup)
A10BA02	metformin (5th level, chemical substance)

### ***3.4 Κριτήρια Εισαγωγής στη Λίστα***

Το WHO Collaborating Centre στο Όσλο δημιουργεί νέες εισαγωγές στην ταξινόμηση ATC μετά από αιτήματα των χρηστών του συστήματος, δηλαδή τις φαρμακευτικές εταιρίες, και τους ερευνητές. Δίνεται προτεραιότητα σε σκευάσματα που περιλαμβάνουν ρητά καθορισμένες ουσίες, οι οποίες έχουν ονομασία INN (International non-proprietary name), τα οποία είναι :

- Νέες χημικές ουσίες που έχουν αιτηθεί άδεια σε αρκετές χώρες
- Ήδη υπάρχουσες ουσίες που χρησιμοποιούνται σε αρκετές χώρες

Τα υπόλοιπα σκευάσματα κρίνονται ανάλογα με την περίπτωση.

Μια καινούρια ουσία συνήθως δε συμπεριλαμβάνεται στην κωδικοποίηση ATC εάν δεν έχει προηγηθεί αίτηση για αδειοδότησή της τουλάχιστον σε μια χώρα.

Παραδοσιακά θεραπευτικά προϊόντα συνήθως δε συμπεριλαμβάνονται στην κωδικοποίηση.

### ***3.5 Αρχές Ταξινόμησης***

Τα σκευάσματα ταξινομούνται ανάλογα με την κύρια θεραπευτική χρήση της κύριας δραστικής ουσίας. Ένα σκεύασμα μπορεί να λάβει περισσότερους από έναν κωδικούς ATC,

εάν διατίθεται σε περισσότερες από μια εκδόσεις με διακριτά διαφορετικές θεραπευτικές χρήσεις.

Ένα σκεύασμα μπορεί να χρησιμοποιηθεί για δύο ή περισσότερες σημαντικές ενδείξεις, και η κύρια θεραπευτική του χρήση μπορεί να διαφέρει μεταξύ των χωρών που διατίθεται. Έτσι, πολλές φορές υπάρχουν πολλές εναλλακτικές επιλογές για την κωδικοποίησή του. Σε αυτά τα σκευάσματα συνήθως δίνεται μόνο ένας κωδικός, με τη βασική να αποφασίζεται ανάλογα με τη διαθέσιμη βιβλιογραφία. Τέτοιου είδους προβλήματα συζητούνται στο WHO International Working Group for Drug Statistics Methodology, όπου αποφασίζεται και η τελική ταξινόμηση. Βέβαια, θα δοθούν αντίστοιχες αναφορές στις σημειώσεις του φαρμάκου, οι οποίες θα τονίζουν τις διαφορετικές χρήσεις του.

### ***3.6 Αλλαγές στην Κωδικοποίηση ATC***

Λόγω του ότι τα διαθέσιμα φάρμακα συνεχώς αλλάζουν και αυξάνονται, οι τακτικές αναθεωρήσεις του συστήματος κωδικοποίησης είναι απαραίτητες. Σύμφωνα με τη W.H.O. :

«Οι αλλαγές στην κωδικοποίηση του συστήματος ATC πρέπει να είναι όσο το δυνατό λιγότερες. Πριν γίνει οποιαδήποτε αλλαγή, πρέπει να ληφθούν υπόψη οι δυσκολίες που θα προκληθούν στους χρήστες του συστήματος, και να συγκριθούν με τα οφέλη τα οποία θα προκύψουν».

Αλλαγές στο σύστημα ATC γίνονται όταν η κύρια χρήση ενός φαρμάκου έχει σαφώς αλλάξει, και όταν είναι αναγκαία η δημιουργία νέων ομάδων για την εισαγωγή νέων ουσιών ή για επίτευξη μεγαλύτερης ακρίβειας στην ομαδοποίηση.

Πρέπει να σημειωθεί πως, όταν αλλάζει ο κωδικός ATC, αναθεωρείται και η μονάδα DDD.

# 4

## ***E.O.Φ. (Εθνικός Οργανισμός Φαρμάκων)- DrugBank***

### ***4.1 E.O.Φ. (Εθνικός Οργανισμός Φαρμάκων)***

#### ***4.1.1 Γενικά***

Ο Εθνικός Οργανισμός Φαρμάκων ιδρύθηκε το 1983 με το Ν. 1316 και είναι Νομικό Πρόσωπο Δημοσίου Δικαίου του Υπουργείου Υγείας και Κοινωνικής Αλληλεγγύης. Αποστολή του ΕΟΦ είναι η προστασία της Δημόσιας Υγείας σε σχέση με την κυκλοφορία στην Ελλάδα[9]:

- φαρμακευτικών προϊόντων ανθρώπινης και κτηνιατρικής χρήσης
- φαρμακούχων ζωοτροφών και προσθετικών ζωοτροφών
- τροφίμων ειδικής διατροφής και συμπληρωμάτων διατροφής
- βιοκτόνων
- ιατρικών βοηθημάτων
- καλλυντικών

Στα πλαίσια της αποστολής αυτής ο ΕΟΦ με απόλυτη διαφάνεια και σε συνεργασία με την Ευρωπαϊκή Ένωση:

- Αξιολογεί και εγκρίνει νέα ασφαλή και αποτελεσματικά προϊόντα
- Παρακολουθεί μετεγκριτικά την ποιότητα, την ασφάλεια και την αποτελεσματικότητα των προϊόντων κατά την κυκλοφορία τους στη χώρα.
- Ελέγχει την παραγωγή, τις κλινικές μελέτες και την κυκλοφορία στην ελληνική αγορά των προϊόντων προκειμένου να τηρούνται οι κανόνες ορθής παραγωγής, εργαστηριακής και κλινικής πρακτικής και να εφαρμόζεται η νομοθεσία όσον αφορά στη διακίνηση, διάθεση, εμπορία και διαφήμισή τους.
- Αναπτύσσει και προωθεί την ιατρική και φαρμακευτική έρευνα.
- Ενημερώνει τους επιστήμονες υγείας, τους αρμόδιους φορείς και το κοινό με αντικειμενικές και χρήσιμες πληροφορίες σχετικά με τα φαρμακευτικά προϊόντα (ανθρώπινης και κτηνιατρικής χρήσης) και τα λοιπά προϊόντα με σκοπό την ορθολογική τους χρήση και την αντικειμενική εκτίμηση της φαρμοκοοικονομικής διάστασής της.

#### **4.1.2 Λειτουργία**

Ο ΕΟΦ διοικείται από Δεκαμελές Διοικητικό Συμβούλιο. Η σημερινή δομή του ΕΟΦ καθορίστηκε με το Π.Δ. 142/89 και την αποτελούν 9 Δ/σεις.

Για την υλοποίηση της αποστολής του ο ΕΟΦ απασχολεί 238 υπαλλήλους, εκ των οποίων 80 είναι φαρμακοποιοί, χημικοί, ιατροί, κτηνίατροι, βιολόγοι και 26 νομικοί, οικονομολόγοι, μαθηματικοί, επιστήμονες πληροφορικής. Παράλληλα συνεργάζεται με εξωτερικούς επιστήμονες (περίπου 400) διαφόρων ειδικοτήτων, συμμετέχει στις επιτροπές και ομάδες εργασίας των αρμοδίων οργάνων της Ευρωπαϊκής Ένωσης και του Συμβουλίου της Ευρώπης με 45 εκπροσώπους του και έχει συγκροτήσει 24 επιστημονικές Επιτροπές και Συμβούλια, που αποτελούνται από ειδικούς επιστήμονες, οι κυριότερες από τις οποίες είναι:

- Επιστημονικό Συμβούλιο Εγκρίσεων. Γνωμοδοτεί για την έγκριση, τροποποίηση, ανανέωση, ανάκληση και αναστολή της άδειας κυκλοφορίας των προϊόντων αρμοδιότητας ΕΟΦ. Λειτουργούν διάφορα τμήματα (συμβατικά φάρμακα, βιολογικά, προϊόντα αίματος, ραδιοφάρμακα, προϊόντα ειδικής διατροφής, κτηνιατρικά φάρμακα και κτηνιατρικά εμβόλια).
- Επιτροπή Φαρμακοεπαγρύπνησης. Αξιολογεί τις ανεπιθύμητες ενέργειες και συνιστά την λήψη μέτρων ή την τροποποίηση της Περίληψης των Χαρακτηριστικών του Προϊόντος (ΠΧΠ) και του Φύλλου Οδηγιών Χρήσης (ΦΟΧ).



- Επιστημονική και Δεοντολογική Επιτροπή Εγκρίσεων Κλινικών Δοκιμών. Εξασφαλίζει την προστασία των δικαιωμάτων, την ασφάλεια και την υγεία των ατόμων που μετέχουν στην κλινική δοκιμή.
- Επιτροπή Φαρμακοποιίας. Συντάσσει την Ελληνική Φαρμακοποιία.
- Επιτροπή Εθνικού Συνταγολογίου. Συντάσσει το Εθνικό Συνταγολόγιο. Αντίστοιχη Επιτροπή λειτουργεί και για το Κτηνιατρικό Συνταγολόγιο.
- Επιτροπή Καλλυντικών. Αποτελείται από ειδικούς επιστήμονες και έχει ως έργο την εισήγηση και επεξεργασία θεμάτων που αφορούν στην κυκλοφορία των καλλυντικών προϊόντων.
- Επιτροπή Ιατρικών Βοηθημάτων. Εισηγείται για τον χαρακτηρισμό των προϊόντων ως ιατρικών βοηθημάτων ή όχι, την κατάταξή τους σε κατηγορίες, τον ορισμό και έλεγχο των Κοινοποιημένων Οργανισμών και για ειδικές ρυθμίσεις σχετικές με την εφαρμογή της νομοθεσίας.
- Δευτεροβάθμιο Επιστημονικό Συμβούλιο. Εξετάζει τις προσφυγές κατά των πρωτοβάθμιων αποφάσεων του ΕΟΦ.

Τέλος ο ΕΟΦ επικουρείται στο έργο του από τις θυγατρικές του εταιρίες κυρίως σε θέματα φαρμακοοικονομικά, ερευνητικά και κάλυψης ελλείψεων της αγοράς, όπως λεπτομερέστερα αναφέρεται παρακάτω.

#### **Ινστιτούτο Φαρμακευτικής Έρευνας και Τεχνολογίας (ΙΦΕΤ) Α.Ε.**

Υποβοηθεί το έργο του ΕΟΦ με την εκπόνηση στατιστικών και οικονομετρικών αναλύσεων, μελετών, ερευνητικών προγραμμάτων και εφαρμογών οργανώσεων και μηχανοργάνωσης. Επίσης το ΙΦΕΤ έχει αναλάβει τη διακίνηση προϊόντων αρμοδιότητας ΕΟΦ για δικό του λογαριασμό ή για λογαριασμό τρίτων για κάλυψη παγίων ή εκτάκτων ελλείψεων της αγοράς.

#### **Ερευνητικό Κέντρο Βιολογικών Υλικών (ΕΚΕΒΥΛ) Α.Ε.**

Κύρια δραστηριότητά του είναι η πιστοποίηση, ο ποιοτικός έλεγχος και η έρευνα γύρω από τα ιατρικά βοηθήματα. Είναι κοινοποιημένος οργανισμός της Ευρωπαϊκής Ένωσης από το 1997 στον τομέα των ιατρικών βοηθημάτων.

#### **Πόροι του ΕΟΦ**

Τους πόρους του Οργανισμού αποτελούν η κρατική επιχορήγηση, τα παράβολα για διάφορες υπηρεσίες που προσφέρει (εγκρίσεις, ανανεώσεις, τροποποιήσεις αδειών κυκλοφορίας, άδειες παραγωγής, βεβαιώσεις, θεωρήσεις τιμολογίων) σχετικά με προϊόντα αρμοδιότητάς του και το ετήσιο πάγιο τέλος που καταβάλλεται από τους υπεύθυνους κυκλοφορίας των προϊόντων (ανταποδοτικό τέλος).

### 4.1.3 Δράση

#### 4.1.3.1 Προεγκριτικοί έλεγχοι

1. Έγκριση άδειας κυκλοφορίας φαρμακευτικών προϊόντων ανθρώπινης και κτηνιατρικής χρήσης

Ισχύουν τρεις διαδικασίες εγκρίσεων:

- Εθνική: αφορά στη χορήγηση άδειας κυκλοφορίας μόνο στην Ελλάδα.
- Αμοιβαία αναγνώριση: όταν ένα φαρμακευτικό ιδιοσκεύασμα έχει εγκριθεί σε ένα κράτος-μέλος της Ευρωπαϊκής Ένωσης, μπορεί να υποβληθεί για έγκριση σε άλλα κράτη-μέλη με την ίδια ακριβώς φαρμακευτική και κλινική τεκμηρίωση. Σε περίπτωση διαφωνίας, παραπέμπεται σε διαιτησία για τελική απόφαση από την Ευρωπαϊκή Επιτροπή.
- Κεντρική διαδικασία: η άδεια εκδίδεται από την Ευρωπαϊκή Επιτροπή μετά από αίτηση στον Ευρωπαϊκό Οργανισμό Φαρμάκων (EMA), αξιολόγηση από τους εμπειρογνώμονες και γνωμάτευση της θεσμοθετημένης Επιτροπής Φαρμακευτικών Ιδιοσκευασμάτων (CPMP: Committee for Proprietary Medicinal Products), όπου συμμετέχουν δύο εκπρόσωποι του κάθε κράτους-μέλους, ή της Επιτροπής Φαρμακευτικών Ιδιοσκευασμάτων για κτηνιατρική χρήση (CVMP: Committee for Veterinary Medicinal Products), όπου επίσης συμμετέχουν δύο εκπρόσωποι του κάθε κράτους-μέλους.

2. Παρακολούθηση κλινικών δοκιμών

Ο ΕΟΦ εγκρίνει τη διεξαγωγή κλινικών δοκιμών και παρακολουθεί και επιθεωρεί την τήρηση της νόμιμης εκτέλεσής τους σύμφωνα με τις Εθνικές ρυθμίσεις, τις κοινοτικές Κατευθυντήριες Γραμμές και τη Διακήρυξη του Ελσίνκι.

3. Διαδικασία έγκρισης και κυκλοφορίας λοιπών προϊόντων αρμοδιότητας ΕΟΦ

- Για τα καλλυντικά ακολουθείται απλουστευμένη διαδικασία. Γνωστοποιείται η κυκλοφορία τους στον ΕΟΦ και οι υπεύθυνοι είναι υποχρεωμένοι να τηρούν φάκελλο με στοιχεία για την παραγωγή, τον έλεγχο και την ασφάλεια των προϊόντων τους, ο οποίος είναι στη διάθεση του ΕΟΦ σύμφωνα με τις σχετικές οδηγίες της Ευρωπαϊκής Επιτροπής.
- Για τα τρόφιμα ειδικής διατροφής και για ορισμένα συμπληρώματα διατροφής χορηγείται άδεια κυκλοφορίας από τον ΕΟΦ, με κριτήρια την ασφάλεια της χρήσης τους και την τεκμηρίωση των ενδείξεών τους.

- Για τα βιοκτόνα ακολουθούνται οι εθνικές ρυθμίσεις όπου απαιτούνται και χορηγείται άδεια κυκλοφορίας, ενώ η σχετική οδηγία της Ευρωπαϊκής Επιτροπής πρόκειται να ενσωματωθεί σύντομα στην Ελληνική νομοθεσία.
- Τα ιατρικά βοηθήματα κυκλοφορούν ελεύθερα, εφόσον τηρούνται ορισμένες διαδικασίες σύμφωνα με τις Ευρωπαϊκές Οδηγίες (σήμανση CE κλπ.).

#### 4.1.3.2 Μετεγκριτικοί έλεγχοι

##### 1. Φαρμακοεπαγρύπνηση

Το σύστημα φαρμακοεπαγρύπνησης συνίσταται στην συνεχή παρακολούθηση της ασφάλειας των φαρμάκων που κυκλοφορούν και είναι ενιαίο για όλες τις χώρες της Ευρωπαϊκής Ένωσης. Βασίζεται στη συλλογή και αξιολόγηση των ανεπιθύμητων ενεργειών και αλληλεπιδράσεων που καταγράφουν στο ειδικό εμπιστευτικό έντυπο της Κίτρινης Κάρτας οι υγειονομικοί (γιατροί, φαρμακοποιοί, οδοντίατροι, νοσηλευτές) καθώς και οι φαρμακευτικές εταιρίες μέσω των υπευθύνων φαρμακοεπαγρύπνησης που απασχολούν.

Η αξιολόγηση των ανεπιθύμητων ενεργειών γίνεται από την Επιτροπή Φαρμακοεπαγρύπνησης του ΕΟΦ και τα στοιχεία που προκύπτουν αποστέλλονται στον Παγκόσμιο Οργανισμό Υγείας και συζητούνται στην Ευρωπαϊκή Επιτροπή Φαρμακοεπαγρύπνησης, όπου λαμβάνονται αποφάσεις δεσμευτικές για όλα τα κράτη μέλη (π.χ. περιορισμός των ενδείξεων, προσθήκη νέων ανεπιθύμητων ενεργειών, αναστολή κυκλοφορίας).

##### 2. Έλεγχος μονάδων παραγωγής και κυκλοφορίας

Έλεγχος μονάδων παραγωγής (Επιθεωρήσεις). Στην Ελλάδα οι παραγωγικές μονάδες ανθρωπίνων και κτηνιατρικών φαρμάκων είναι περίπου 60, των καλλυντικών 100, των ζωοτροφών 200, των ιατρικών αερίων 15. Ο έλεγχος γίνεται σύμφωνα με τους κανόνες καλής παραγωγής που ισχύουν σε όλη την Ευρωπαϊκή Ένωση (Good Manufacturing Practice-GMP).

Ήδη έχει ξεκινήσει η διαδικασία διαπίστευσης σύμφωνα με το Ευρωπαϊκό Πρότυπο EN 45004. Με την διαπίστευση αποδεικνύεται η τεχνική ικανότητα των επιθεωρητών να ενεργούν σύμφωνα με συγκεκριμένες διαδικασίες, που περιγράφονται στο ως άνω Πρότυπο. Η διαδικασία διαπίστευσης ξεκίνησε τον Απρίλιο του 1997 και ευρίσκεται στη φάση της πρακτικής εφαρμογής. Η διαπίστευση είναι απαραίτητη μέσα στο κοινοτικό πλαίσιο, ώστε τα αποτελέσματα των διενεργούμενων ελέγχων να είναι αποδεκτά σε όλη την Ευρωπαϊκή Ένωση.

##### 3. Έλεγχος κυκλοφορίας

Συλλέγονται δείγματα από την αγορά με σκοπό τον εργαστηριακό έλεγχο ή τον έλεγχο της νομιμότητας της κυκλοφορίας τους. Εξετάζονται όλες οι καταγγελίες, που υποβάλλουν υγειονομικοί, νοσοκομεία ή καταναλωτές και αφορούν στην ποιότητα και την νόμιμη κυκλοφορία των προϊόντων.

Παρακολούθηση κατανάλωσης-Κάλυψη ελλείψεων-Ταινία γνησιότητας  
Παρακολουθείται σε μηνιαία βάση η παραγωγή, εισαγωγή και οι πωλήσεις των φαρμακευτικών προϊόντων. Σύμφωνα με τα στοιχεία που προκύπτουν ανιχνεύονται οι τάσεις της συνταγογράφησης και η επίδραση του ύψους των τιμών στην κατανάλωση. Ο ΕΟΦ διανέμει την ταινία γνησιότητας στις εταιρίες. Η ταινία γνησιότητας επικολλάται σε κάθε συσκευασία φαρμάκου και έτσι εξασφαλίζεται η αυθεντικότητα του κάθε φαρμάκου και η καλή λειτουργία του συστήματος οικονομικής απόδοσης των ταμείων.

Τέλος σημαντική κοινωνική προσφορά του ΕΟΦ είναι η κάλυψη των βασικών ελλείψεων της αγοράς σε φάρμακα.

#### 4. Εργαστηριακοί Έλεγχοι

Με σκοπό την προάσπιση της Δημόσιας Υγείας και την εξασφάλιση του Δημόσιου Συμφέροντος ,γενικότερα, στο χώρο του Φαρμάκου και των λοιπών συναφών Προϊόντων , ο ΕΟΦ εκτελεί –μέσω των Εργαστηρίων του-πλήθος Φυσικοχημικών, Μικροβιολογικών, Φαρμακοτεχνικών και λοιπών Ελέγχων σε Φαρμακευτικές Πρώτες Ύλες , Φαρμακευτικά Προϊόντα για Ανθρώπινη και Κτηνιατρική χρήση, καθώς και σε Καλλυντικά, Φαρμακούχες Ζωοτροφές, Συμπληρώματα Διατροφής και Ιατρικά Βοηθήματα. Αυτοί οι Εργαστηριακοί Έλεγχοι εντάσσονται στο ετήσιο τακτικό πρόγραμμα Μετεγκριτικού Ελέγχου της Εσωτερικής Αγοράς από τον ΕΟΦ ή της Ευρωπαϊκής Αγοράς από την Ευρωπαϊκή Διεύθυνση για την Ποιότητα των Φαρμάκων ((EDQM: European Department Quality Medicines), που λειτουργεί υπό την αιγίδα του Συμβουλίου της Ευρώπης) με την οποία συνεργάζεται ο ΕΟΦ. Παράλληλα πραγματοποιούνται Εργαστηριακοί Έλεγχοι σε Προϊόντα αρμοδιότητας ΕΟΦ επί τη βάσει Καταγγελιών προερχομένων κατά κύριο λόγο από τα Στελέχη του Υγειονομικού Συστήματος της χώρας (Ιατρούς, Φαρμακοποιούς κ.λπ.).

#### Διαπίστευση Εργαστηρίων

##### Καθιέρωση Ενιαίου Συστήματος Ποιότητας ISO 17025

Τα Εργαστήρια του ΕΟΦ προκειμένου να τεκμηριώσουν την Τεχνική τους Επάρκεια και να κατοχυρώσουν την αδιαμφισβήτητη Αξιοπιστία των αποτελεσμάτων-συμπερασμάτων των Εργαστηριακών Ελέγχων που εκτελούν, είχαν ξεκινήσει από το 1997 τη σχεδίαση και εφαρμογή κατάλληλου Συστήματος Ποιότητας.

Μετά από πολύχρονη και επίπονη προετοιμασία, υπέβαλαν, το Νοέμβριο του 2003, αίτηση στο Εθνικό Σύστημα Διαπίστευσης (ΕΣΥΔ) προκειμένου να επιτευχθεί η Διαπίστευση, σύμφωνα με τις απαιτήσεις του προτύπου ΕΛΟΤ EN ISO/IEC 17025, των σημαντικότερων από τις Εργαστηριακές Δοκιμές που πραγματοποιούν.

Το Μάρτιο του 2004 συντάχθηκε από την ορισθείσα Ομάδα Αξιολόγησης του ΕΣΥΔ η Προκαταρκτική Έκθεση και τον Ιούνιο του 2004 έλαβε χώρα η Προαξιολόγηση των Εργαστηρίων. Η Τελική Αξιολόγηση πραγματοποιήθηκε στις αρχές Μαρτίου 2005 και το σχετικό Πιστοποιητικό Διαπίστευσης κατά ΕΛΟΤ EN ISO/IEC 17025 χορηγήθηκε το Μάρτιο του 2006.

Επί πλέον, η συνεχής και συνεπής τήρηση υψηλών απαιτήσεων Ποιότητας ελέγχεται μεταξύ άλλων με τη συμμετοχή των Εργαστηρίων σε Διεθνή Διεργαστηριακά Προγράμματα Αξιολόγησης της Τεχνικής τους Επάρκειας καθώς και με την ετήσια επιτήρηση του Συστήματος Ποιότητας από το ΕΣΥΔ.

Τα Εργαστήρια του ΕΟΦ συμμετέχουν στο Δίκτυο των Επισήμων Ευρωπαϊκών Εργαστηρίων Ελέγχου Φαρμάκων (OMCL: Official Medicines Control Laboratories), το οποίο συντονίζεται από την Ευρωπαϊκή Διεύθυνση για την Ποιότητα των Φαρμάκων (EDQM). Η συνεργασία αυτή καλύπτει τον Μετεγκριτικό Έλεγχο Προϊόντων τα οποία κυκλοφορούν στην Ευρωπαϊκή Αγορά και έχουν εγκριθεί με την Κεντρική ή την Αμοιβαία Διαδικασία. Επίσης καλύπτει Διεργαστηριακές Μελέτες απόδειξης της Τεχνικής Επάρκειας του Εργαστηρίου καθώς και άλλες Μελέτες σχετικές με την Ευρωπαϊκή Φαρμακοποιία. Τόσο η επίτευξη της Διαπίστευσης των Εργαστηρίων του ΕΟΦ όσο και η διατήρηση και συνεχής ικανοποίηση των αυστηρών απαιτήσεων του Συστήματος Ποιότητας που συνεπάγεται, απαιτούν δέσμευση αυξημένων ανθρωπίνων και υλικών πόρων. Ωστόσο, οι όποιες αυξημένες απαιτήσεις αντισταθμίζονται πλήρως από την αποτελεσματική και φερέγγυα προώσπιση της Δημόσιας Υγείας.

#### *4.1.3.3 Πληροφόρηση-Εκδόσεις ΕΟΦ*

Είναι ο τομέας μέσα από τον οποίο ο ΕΟΦ υπηρετεί τη διαφάνεια και την ενημέρωση για την ορθολογική χρήση των φαρμάκων και τη φαρμακοεπαγρύπνηση.

Στις εκδόσεις του Οργανισμού περιλαμβάνονται το Εθνικό Συνταγολόγιο, το Κτηνιατρικό Συνταγολόγιο, η Ελληνική Φαρμακοποιία και η περιοδική έκδοση Φάρμακα και Συνταγογραφία.

Από το 2000 σε συνεννόηση με τον Πανελλήνιο Ιατρικό Σύλλογο (ΠΙΣ) ο Οργανισμός δημοσιεύει στο "Ιατρικό Βήμα", επίσημο όργανο του ΠΙΣ, τα "Νέα του ΕΟΦ", ενημερωτική

σελίδα με σύγχρονες πληροφορίες για τη λειτουργία του Οργανισμού και τα επείγοντα νέα για προϊόντα της αρομοδιότητας του.

## **4.2 DrugBank**

Η βάση δεδομένων DrugBank [10] που είναι διαθέσιμη στο Πανεπιστήμιο της Αλμπέρτα είναι ένας πόντος βιοπληροφορικής και χημειοπληροφορικής που συνδυάζει λεπτομερή δεδομένα (δηλαδή χημικές, φαρμακολογικές και φαρμακευτικές) φαρμάκων με εκτενείς πληροφορίες (δηλαδή ακολουθία, δομή, οδό χορήγησης) του στόχου-φαρμάκου. Η βάση δεδομένων περιέχει περίπου 4800 καταχωρίσεις φαρμάκων συμπεριλαμβανομένων:

- 1480 FDA-εγκεκριμένα μικρομοριακά φάρμακα,
- 128 FDA εγκεκριμένα βιοτεχνολογικά (πρωτεϊνών /πεπτιδίων) φάρμακα,
- 71 τροφοφάρμακα (nutraceuticals), και
- 3200 πειραματικά φάρμακα.

Περισσότερες από 2500 πρωτεϊνικές ακολουθίες συνδέονται με αυτές τις καταχωρίσεις φαρμάκων. Κάθε εγγραφή DrugCard περιέχει περισσότερα από 100 πεδία δεδομένων με τις μισές από τις πληροφορίες να αφορούν τα φαρμακολογικά / χημικά δεδομένα και τις άλλες μισές τον θεραπευτικό στόχο ή την πρωτεΐνη.

Το DrugBank συντηρείται από τους David Wishart και Craig Knox.

# 5

## *Αρχιτεκτονική .NET, η γλώσσα προγραμματισμού C# και ο SQL Server*

### *5.1 .NET framework*

#### *5.1.1 Εισαγωγή*

Το .NET framework [16] είναι ένα περιβάλλον που ελέγχει την εκτέλεση ενός προγράμματος. Ουσιαστικά είναι υπεύθυνο για να ξεκινήσει το πρόγραμμα, να αποκτήσει την απαραίτητη άδεια (permissions), να αποδώσει την απαραίτητη μνήμη για την αποθήκευση δεδομένων, να αποδώσει στο σύστημα τη μνήμη και τους υπόλοιπους πόρους που δεν χρησιμοποιούνται πλέον από το πρόγραμμα κτλ. Επίσης περιλαμβάνει μία βιβλιοθήκη κλάσεων (class library) που μπορούν να χρησιμοποιηθούν από προγράμματα που τρέχουν στο περιβάλλον .NET, οι οποίες μπορούν να εκτελέσουν ένα ευρύ φάσμα λειτουργιών.

Ακολουθούν κάποιοι ορισμοί που θα αποσαφηνίσουν περισσότερο την έννοια και τα χαρακτηριστικά του .NET framework.

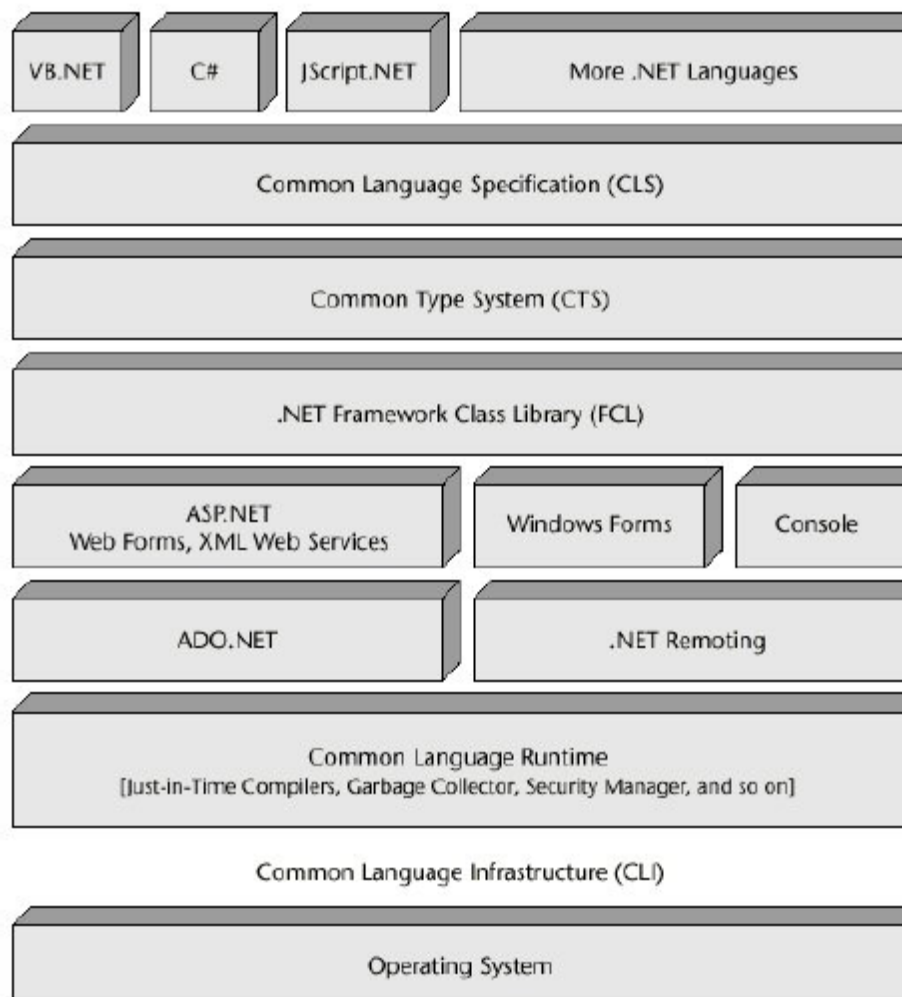
- .NET Runtime ή Common Language Runtime (CLR) : είναι αυτό ουσιαστικά που ελέγχει ένα κώδικα (φόρτωση, εκτέλεση κτλ).

- The .NET languages: Αυτές περιλαμβάνουν την Visual Basic, C#, JScript .NET ( μια server-side εκδοχή της JavaScript), J# (κλώνος της Java), και C++.
- Managed Code : είναι κάθε κομμάτι κώδικα που είναι σχεδιασμένος να εκτελεστεί στο περιβάλλον .NET. Κάθε άλλος κώδικας αναφέρεται ως unmanaged.
- Intermediate Language (IL) : Προκειμένου τα προγράμματα που είναι σχεδιασμένα για να τρέχουν σε .NET framework να είναι ανεξάρτητα από την αρχιτεκτονική του επεξεργαστή που τελικά θα εκτελεστούν, ο πηγαίος κώδικας ενός προγράμματος μεταφράζεται σε εντολές (assembly language) ενός εικονικού επεξεργαστή. Αργότερα, κατά την φόρτωση ενός προγράμματος από το CLR, όταν πλέον είναι γνωστή η ταυτότητα του επεξεργαστή και επομένως του ρεπερτορίου των εντολών, οι καταχωρητές του κτλ, ο ενδιάμεσος αυτός κώδικας και αφού γίνουν κάποιες βελτιστοποιήσεις μετατρέπεται σε τελικό προκειμένου να εκτελεστεί ορθά.
- Common Type System (CTS) : Στο περιβάλλον .NET υπάρχει διασυνδεσιμότητα (interoperability) μεταξύ των διαφόρων γλωσσών προγραμματισμού. Δηλαδή αφού ένα κομμάτι κώδικα έχει μεταφραστεί σε Intermediate Language δεν μπορεί να πλέον να αναγνωρισθεί η υψηλού επιπέδου γλώσσα που το δημιούργησε. Έτσι μία εφαρμογή που είναι γραμμένη σε μία γλώσσα προγραμματισμού Α μπορεί χωρίς κανένα πρόβλημα να καλεί συναρτήσεις, ή να δημιουργεί κλάσεις παράγωγες μίας που έχει δημιουργηθεί από μία γλώσσα προγραμματισμού Β. Για να επιτευχθεί αυτό θα πρέπει να κάποιιο κοινός βασικοί τύποι δεδομένων. Το CTS παρέχει αυτούς τους τύπους καθώς και τους κανόνες για να δημιουργηθούν νέοι custom τύποι.
- .NET Base Classes : πρόκειται για μία βιβλιοθήκη κλάσεων που παρέχουν στους χρήστες των μεγάλο αριθμό λειτουργιών π.χ. controls σε user interfaces, ζωγραφική στην οθόνη, απόκτηση πρόσβασης στο Internet, εγγραφή – ανάγνωση σε αρχεία κτλ.
- Assemblies : είναι το μέρος αποθήκευσης ενός compiled managed κώδικα, δηλαδή είτε κάποιο εκτελέσιμο αρχείο (exe) είτε κάποια βιβλιοθήκη dll. Εκτός από τον εκτελέσιμο κώδικα περιλαμβάνει και πληροφορία για τον κώδικα., π.χ. ποιοι τύποι, μέθοδοι κ.τ.λ δηλώνονται. Η πληροφορία αυτή ονομάζεται metadata.
- Common Language Specification (CLS) : πρόκειται για κανόνες που πρέπει να ακολουθεί κάποιο πρόγραμμα προκειμένου να μπορεί να εκτελεστεί στο περιβάλλον .NET. Ουσιαστικά αφορά κατασκευαστές compilers που σκοπεύουν να παραξουν IL του .NET.
- Garbage Collection : Το CLR είναι υπεύθυνο για την απόδοση στο σύστημα των πόρων που πλέον δεν χρησιμοποιούνται από μία εφαρμογή. Έτσι μία εφαρμογή δεν



χρειάζεται να μεριμνά για αυτή τη διαδικασία αφού αυτή πραγματοποιείται αυτόματα από το σύστημα. [1]

Τα προγράμματα μεταγλωττίζονται μέσω του .NET Framework σε δύο βήματα: πρώτα, το πρόγραμμα μεταγλωττίζεται σε ενδιάμεσο κώδικα (bytecode) γραμμένο σε Microsoft Intermediate Language (MSIL), ο οποίος ουσιαστικά ορίζει οδηγίες για το CLR που λειτουργεί ως εικονική μηχανή (virtual machine). Κατόπιν, ένας άλλος μεταγλωττιστής στο CLR μεταφράζει τη MSIL σε κώδικα μηχανής για μια συγκεκριμένη πλατφόρμα, δημιουργώντας έτσι μια μόνο εφαρμογή. Η αρχιτεκτονική αυτή των δύο βημάτων παρέχει διαλειτουργικότητα ανάμεσα στις γλώσσες και έλεγχο πάνω στα χαρακτηριστικά διαχείρισης της εκτέλεσης, όπως στη διαχείριση μνήμης και την ασφάλεια. [3]



Εικόνα 4.1– Η αρχιτεκτονική του .NET Framework

### 5.1.2 Πλεονεκτήματα χρήσης του .NET framework

Μερικά από πλεονεκτήματα από τη χρήση του .NET framework είναι [14]:

- Ασφάλεια : Αφού κάθε assembly που είναι γραμμένο σε IL μεταφέρει και πληροφορία για τους τύπους και τις μεθόδους που περιλαμβάνει, το CLR γνωρίζει με απόλυτη ακρίβεια τις λειτουργίες που δύναται να εκτελεσει αυτό. Έτσι θέτοντας ένας χρήστης ένα επίπεδο ασφάλειας που πιστεύει ότι χρειάζεται να αποκτήσει μία εφαρμογή (π.χ. μία που εκτελείται από το internet δεν πρέπει να διαβάσει πληροφορία από ένα σκληρό δίσκο) το CLR αποφασίζει αν θα εκτελεστεί τελικά η εφαρμογή. Η ασφάλεια μπορεί να είναι role-based (χρησιμοποιώντας την ταυτότητα της διεργασίας που τρέχει ο κώδικας) ή code-based (βασισμένη στο κατά πόσο ο κώδικας αυτός καθ'αυτός κρίνεται ασφαλείς).
- Garbage Collection : Η εφαρμογή δεν είναι πλέον υπεύθυνη για τη διαχείριση της μνήμης αφού αυτή πραγματοποιείται αυτόματα από το σύστημα (αν και υπάρχει μία μικρή επιβάρυνση του υπολογιστικού συστήματος από αυτή τη λειτουργία).
- Language interoperability : Σε μία σύνθετη εφαρμογή μπορούν να χρησιμοποιηθούν διάφορες γλώσσες προγραμματισμού για τα διάφορα assemblies που μπορεί να αποτελείται. Τα assemblies αυτά μπορούν να επικοινωνήσουν μεταξύ τους, π.χ. κλάσεις οι οποίες είναι αποθηκευμένες σε ένα assembly σε μία γλώσσα προγραμματισμού μπορούν να χρησιμοποιηθούν άμεσα από ένα άλλο (instantiation, inheritance κτλ), το οποίο είναι γραμμένο σε διαφορετική γλώσσα προγραμματισμού.
- Η αυτοεπεξηγηματική (self-describing) δομή των assemblies εξαλείφει την πιθανότητα λαθών (bugs) λόγω διαφορετικού versioning.
- Οι βασικές κλάσεις (.NET base classes) που παρέχονται αποτελούν πολύτιμη βοήθεια για την ανάπτυξη πολύπλοκων εφαρμογών. Πρόκειται για μία ολοκληρωμένη βιβλιοθήκη κλάσεων - πολύ πιο πλήρης από κάθε άλλη π.χ. MFC (Visual C++) ή VCL (Borland – Imprise Delphi) – που παρέχει έτοιμες λειτουργίες όπως διαχείριση XML και βάσεων δεδομένων, networking, drawing κτλ.
- Αντικειμενοστραφής προγραμματισμός — Τόσο το .NET Framework όσο και η C# που θα χρησιμοποιήσουμε είναι αποκλειστικά βασισμένες σε αντικειμενοστραφείς αρχές από την βάση τους.

### 5.1.3 Ιστορία του .NET Framework

- .NET Framework Version 1.0 το έτος 2002.
- .NET Framework Version 1.1 το έτος 2003.
- .NET Framework Version 2.0 το έτος 2005.
- .NET Framework Version 3.0 το έτος 2006.
- .NET Framework Version 3.5 το έτος 2007.

Το .NET Framework 3 θεωρείται μια σημαντική έκδοση του προϊόντος με πολλά νέα χαρακτηριστικά (αυτά που προστεθηκαν linq to sql).

### 5.1.4 LINQ (Language Integrated Query)

Η LINQ [11] είναι κομμάτι του Microsoft .NET Framework που προσθέτει εγγενείς data querying δυνατότητες στις .NET γλώσσες.

Η LINQ ορίζει ένα σύνολο δεσμευμένων μεθόδων (ονομάζονται standard query τελεστές, η standard sequence τελεστές), μαζί με κανόνες μετάφρασης από τις αποκαλούμενες query εκφράσεις σε εκφράσεις που χρησιμοποιούν αυτές τις μεθόδους, λάμδα εκφράσεις και ανώνυμους τύπους. Αυτές μπορούν για παράδειγμα να χρησιμοποιηθούν για να προβάλλουν και να φιλτράρουν δεδομένα σε πίνακες, αριθμήσιμες κλάσεις, xml (LINQ to XML), σχεσιακή βάση δεδομένων και άλλες πηγές δεδομένων. Άλλες χρήσεις περιλαμβάνουν την κατασκευή event handlers ή monadic parsers.

Πολλές από τις αρχές που εισήγαγε η LINQ δοκιμάστηκαν για πρώτη φορά στο ερευνητικό έργο Cω της Microsoft. Η LINQ κυκλοφόρησε ως μέρος του .NET Framework 3.5 στις 19 Νοεμβρίου 2007.

#### 5.1.4.1 Αρχιτεκτονική της LINQ στο .NET Framework

Delegate είναι ένας τύπος δεδομένων (reference type) όπου παρέχει έναν ασφαλή τρόπο για να ενθυλακώσουμε συναρτήσεις (είτε static είτε instance). Ένα Delegate μπορεί να αναφέρεται μόνο σε συναρτήσεις που είναι συμβατές με το signature που περιγράφει.

Τυπικοί Query Τελεστές:

#### **Select**

Ο τελεστής Select πραγματοποιεί προβολή στη συλλογή για να επιλέξει ενδιαφέρουσες πτυχές των στοιχείων. Ο χρήστης εισάγει μια αυθαίρετη συνάρτηση, σαν τύπο delegate ή λάμδα έκφραση, η οποία προβάλλει τα data members.

### **Where**

Ο τελεστής Where επιτρέπει τον ορισμό ενός συνόλου κατηγορηματικών κανόνων οι οποίοι αποτιμώνται για κάθε αντικείμενο στη συλλογή, ενώ αντικείμενα που δεν υπόκεινται στον κανόνα αποκλείονται. Το κατηγορηματικό χρησιμοποιείται στον τελεστή σαν τύπος delegate.

### **OrderBy**

Ο τελεστής OrderBy χρησιμοποιείται για να ορίσει την ταξινόμηση των στοιχείων της συλλογής με βάση κάποιο κλειδί. Η προκαθορισμένη ταξινόμηση είναι κατά αύξουσα σειρά, για να αντιστραφεί η σειρά, χρησιμοποιείται ο τελεστής OrderByDescending.

### **Distinct**

Ο Distinct τελεστής αφαιρεί τις διπλοεγγραφές μιας ενός γνωρίσματος-κλειδιού από τη συλλογή.

### **Single**

Ο τελεστής Single παίρνει ένα κατηγορηματικό και επιστρέφει το στοιχείο που ικανοποιεί το κατηγορηματικό. Εγείρει exception αν κανένα ή περισσότερα από ένα στοιχεία ικανοποιούν το κατηγορηματικό.

### **ElementAt**

Ο τελεστής ElementAt επιστρέφει το στοιχείο σε ένα συγκεκριμένο index στη συλλογή.

### **Contains**

Ο τελεστής Contains ελέγχει αν η συλλογή περιέχει ένα καθορισμένο σύνολο στοιχείων.

### **Count**

Ο τελεστής Count επιστρέφει το πλήθος στοιχείων της δοσμένης συλλογής.

Επίσης ορίζονται κάποιοι τελεστές που μετετρέπουν μια συλλογή σε ένα άλλο τύπο.

**AsEnumerable**: μετατρέπει τη συλλογή σε IEnumerable<T> τύπο.

**ToArray**: μετατρέπει τη συλλογή σε πίνακα.

#### *5.1.4.2 LINQ to SQL*

Ο LINQ to SQL provider επιτρέπει τη LINQ να χρησιμοποιηθεί για ερωταποκρίσεις στις SQL Server βάσεις δεδομένων, συμπεριλαμβανόμενων [SQL Server Compact](#) βάσεων δεδομένων. Από την στιγμή που τα SQL Server δεδομένα μπορούν να υπάρχουν σε ένα απομονωμένο server, και επειδή ο SQL Server έχει την δική του μηχανή ερωταποκρίσεων, η LINQ to SQL δεν χρησιμοποιεί την μηχανή ερωταποκρίσεων της LINQ. Αντίθετα, μετατρέπει ένα LINQ query σε ένα [SQL](#) query το οποίο τότε στέλνεται στο SQL Server για επεξεργασία. Όμως από την στιγμή που ο SQL Server αποθηκεύει τα δεδομένα ως σχεσιακά

και η LINQ δουλεύει με δεδομένα ενθυλακωμένα σε αντικείμενα, οι δυο αναπαραστάσεις πρέπει να αντιστοιχούν η μια στην άλλη. Για αυτόν το λόγο η LINQ to SQL επίσης ορίζει ένα framework αντιστοίχισης. Η αντιστοίχιση γίνεται ορίζοντας κλάσεις που αντιστοιχούν στους πίνακες στην βάση δεδομένων, και περιλαμβάνοντας όλα η ένα υποσύνολο των στηλών στον πίνακα ως data members. Η αντιστοίχιση, μαζί με άλλες ιδιότητες του σχεσιακού μοντέλου όπως τα κύρια κλειδιά, ορίζονται χρησιμοποιώντας LINQ to SQL-defined ιδιότητες.

### **5.1.5 Χρήση Master Pages**

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

Αυτό αρχικά μπορεί να γίνει με το χέρι, όμως από τη στιγμή που ο αριθμός των σελίδων αυξάνει και ειδικά αν χρειάζονται αλλαγές η εργασία που χρειάζεται είναι πάρα πολύ μεγάλη. Για αυτό και υπάρχουν διάφορες τεχνικές οι οποίες υποστηρίζουν κάποιο είδος «template» με τη βοήθεια του οποίου όλες οι σελίδες θα έχουν κοινά χαρακτηριστικά. Το «template» αυτό στην περίπτωση του ASP.NET ονομάζεται «Master Page» [12]. Μια master page είναι μια απλή ASP.NET σελίδα η οποία περιέχει μια σειρά από κανονικά στοιχεία τα οποία θα εμφανίζονται σε όλες τις άλλες σελίδες που την χρησιμοποιούν ως template. Επίσης, περιέχει μια σειρά από στοιχεία της μορφής <asp:contentplaceholder>.

Αυτά τα content place holders θα χρησιμοποιηθούν από τις σελίδες που χρησιμοποιούν τη master page ώστε να τοποθετήσουν το περιεχόμενο τους εκεί. Έτσι, η δομή όλων των σελίδων θα είναι η ίδια ακριβώς και το μόνο που θα αλλάζει θα είναι το περιεχόμενο του content place holder. Κάθε σελίδα που χρησιμοποιεί τη master page ως template θα πρέπει να το δηλώνει στην επικεφαλίδα της με κάποια γραμμή της μορφής masterPageFile="MasterPage.master". Για να χρησιμοποιηθεί ένας content place holder χρησιμοποιείται το στοιχείο <asp:content> του οποίου τα περιεχόμενα θα τοποθετηθούν στον αντίστοιχο content place holder. Η τεχνική αυτή χρησιμοποιήθηκε και εδώ έτσι ώστε να υπάρχει μια ομοιόμορφη δομή σε όλες τις σελίδες που αναπτύχθηκαν και φυσικά να είναι ευκολότερη ανάπτυξη τους.

### **5.1.6 Ajax**

Το Ajax [19] (συντομογραφία για Asynchronous JavaScript and XML) είναι ένα σύνολο αλληλένδετων μεθόδων ανάπτυξης ιστοσελίδων που χρησιμοποιούνται στην πλευρά του client για τη δημιουργία δυναμικών web εφαρμογών. Με τη χρήση του Ajax, οι web

εφαρμογές μπορούν να ανακτούν δεδομένα από τον Server ασύγχρονα στο παρασκήνιο χωρίς να αλληλεπιδρά με την εμφάνιση και τη συμπεριφορά της υπάρχουσας σελίδας. Τα δεδομένα συνήθως ανακτώνται χρησιμοποιώντας το XMLHttpRequest αντικείμενο. Παρά το όνομα, η χρήση της XML δεν είναι απαραίτητη, ενώ τα request δεν χρειάζεται να είναι ασύγχρονα.

Το Ajax δεν είναι μια τεχνολογία, αλλά μια ομάδα τεχνολογιών. Το Ajax χρησιμοποιεί ένα συνδυασμό HTML και CSS για τη στυλιστική απεικόνιση των πληροφοριών. Το DOM είναι προσβάσιμο με Javascript για να εμφανίζει δυναμικά, και να επιτρέπει στο χρήστη να αλληλεπιδρά με τις πληροφορίες που παρουσιάζονται. Η JavaScript και το XMLHttpRequest αντικείμενο παρέχουν μια μέθοδο για την ασύγχρονα ανταλλαγή δεδομένων μεταξύ του browser και του server για να αποφευχθούν οι πλήρεις ανανεώσεις σελίδας.

## **Ιστορία**

Στη δεκαετία του 1990, οι περισσότερες ιστοσελίδες βασίζονταν σε αμιγώς σελίδες HTML. Κάθε ενέργεια χρήστη απαιτούσε την ανανέωση της σελίδας από το server (ή τη φόρτωση μιας νέας σελίδας). Αυτή η διαδικασία είναι μη αποδοτική. Κάθε φορά που μια σελίδα ανανεώνεται που οφείλεται σε μερική τροποποίηση, όλο το περιεχόμενο θα πρέπει να αποσταλεί ξανά, αντί μόνο των τροποποιημένων πληροφοριών. Αυτό μπορεί να προσθέσει επιπλέον φορτίο στον server και να κάνει χρήση υπερβολικού εύρους ζώνης.

Ασύγχρονη φόρτωση του περιεχομένου έγινε για πρώτη φορά πρακτική όταν εισήχθησαν τα Java applets στην πρώτη έκδοση της γλώσσας το 1995. Αυτά επιτρέπουν στον μεταγλωττισμένο client-side κώδικα να φορτώσει δεδομένα ασύγχρονα από το web server μετά την φόρτωση μιας ιστοσελίδας. Το 1996, ο Internet Explorer εισήγαγε το στοιχείο iframe στην HTML, το οποίο επέτρεψε την ασύγχρονα φόρτωσης. Το 1999, η Microsoft δημιούργησε το XMLHttpRequest ActiveX control στον Internet Explorer 5, το οποίο υιοθετήθηκε αργότερα από το Mozilla, Safari, Opera και άλλους browsers, όπως το XMLHttpRequest JavaScript αντικείμενο. Η Microsoft έχει υιοθετήσει το εγγενές XMLHttpRequest μοντέλο, για τον Internet Explorer 7, αν και το ActiveX υποστηρίζεται ακόμα. Η δυνατότητα των HTTP request στον server και οι ασύγχρονες τεχνολογίες web παρέμειναν αρκετά ασαφής έως ότου άρχισαν να εμφανίζονται σε ευρεία κλίμακα online εφαρμογές όπως το Outlook Web Access (2000) και Oddpost (2002), και αργότερα, η Google εφάρμοσε εκτενώς το Ajax με το Gmail (2004) και το Google Maps (2005).

Ο όρος Ajax επινοήθηκε στις 18 Φλεβάρη του 2005 από Jesse James Garrett σε ένα άρθρο με τίτλο Ajax: A New Approach to Web Applications. Στις 5 Απριλίου του 2006 το World Wide Web Consortium (W3C) εξέδωσε ένα προσχέδιο προδιαγραφής για το

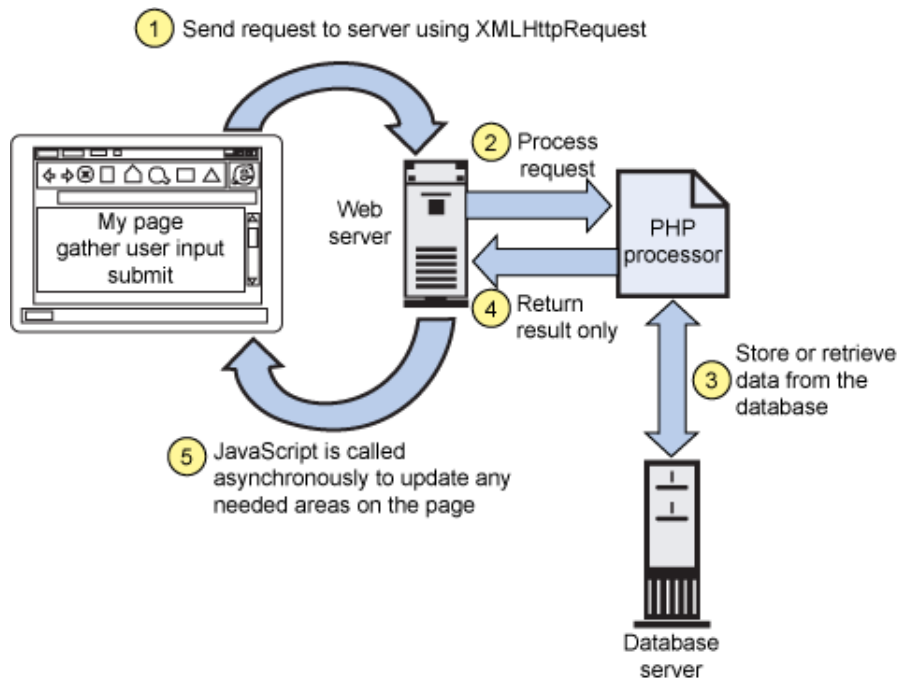
XMLHttpRequest αντικείμενο σε μια προσπάθεια να δημιουργηθεί ένα επίσημο web πρότυπο.

### Τεχνολογίες

Ο όρος Ajax αντιπροσωπεύει μια ευρεία ομάδα δικτυακών τεχνολογιών που μπορούν να χρησιμοποιηθούν για την υλοποίηση μιας διαδικτυακής εφαρμογής που επικοινωνεί με ένα server στο παρασκήνιο, χωρίς να αλληλεπιδρά με την τρέχουσα κατάσταση της σελίδας. Στο άρθρο που επινοήθηκε τον όρο Ajax, Jesse James Garrett εξήγησε ότι ενσωματώνονται οι ακόλουθες τεχνολογίες:

- HTML ή XHTML και CSS για την παρουσίασή.
- Το Document Object Model (DOM) για δυναμική εμφάνιση και αλληλεπίδραση με δεδομένα.
- XML για την ανταλλαγή δεδομένων, και XSLT για το χειρισμό τους.
- Το XMLHttpRequest αντικείμενο την ασύγχρονη επικοινωνία.
- Javascript για να φέρει αυτές τις τεχνολογίες μαζί.

Έκτοτε, ωστόσο, υπήρξαν ορισμένες εξελίξεις στις τεχνολογίες που χρησιμοποιούνται σε μια εφαρμογή Ajax, καθώς και στον ορισμό του όρου Ajax. Ειδικότερα, διαπιστώθηκε ότι η JavaScript δεν είναι η μόνη γλώσσα προγραμματισμού client-side που μπορεί να χρησιμοποιηθούν για την υλοποίηση μιας αίτησης Ajax, καθώς και άλλες γλώσσες, όπως η VBScript είναι ικανές να παρέχουν την απαιτούμενη λειτουργικότητα. (Ωστόσο, η JavaScript είναι η πιο δημοφιλής γλώσσα για τον προγραμματισμό Ajax λόγω ότι περιλαμβάνεται και είναι συμβατό με την πλειοψηφία των σύγχρονων web browsers.). Επίσης, η XML δεν είναι απαραίτητη για την ανταλλαγή δεδομένων και, συνεπώς, η XSLT δεν απαιτείται για το χειρισμό των δεδομένων. Το JavaScript Object Notation (JSON) χρησιμοποιείται συχνά ως εναλλακτικό πρότυπο για την ανταλλαγή δεδομένων, αν και άλλα πρότυπα, όπως η προ-προσωποποιημένη HTML ή το απλό κείμενο μπορούν επίσης να χρησιμοποιηθούν



Εικόνα 4.2 – Ασύγχρονη επικοινωνία με το Server

### Μειονεκτήματα

Σελίδες που έχουν δημιουργηθεί δυναμικά με χρήση διαδοχικών Ajax request δεν αποθηκεύονται στο ιστορικό του browser, έτσι όταν πατήσεις το back button του browser, δεν μπορεί να επιστρέψει ο browser σε μια προηγούμενη κατάσταση της Ajax-enabled σελίδας, αλλά μπορεί αντ' αυτού μπορεί να επιστρέψει στην προηγούμενη σελίδα που είχε επισκέφθει.

- Τα URLs δεν αλλάζουν καθώς αλλάζει η σελίδα.
- Μπορεί να υπάρχει ασυμβατότητα μεταξύ των browsers.
- Η Javascript μπορεί να καταναλώνει πολύ υπολογιστική ισχύ (ειδικά σε παλιότερους υπολογιστές).
- Η Javascript μπορεί να είναι απενεργοποιημένη για λόγους ασφαλείας και επομένως η εφαρμογή να μη λειτουργεί.
- Το Debugging είναι αρκετά δύσκολο.
- Κάθε χρήστης του οποίου το πρόγραμμα περιήγησης δεν υποστηρίζει JavaScript ή XMLHttpRequest, ή απλά έχει την λειτουργία αυτή απενεργοποιημένη, δεν θα είναι σε θέση να χρησιμοποιεί σωστά τις σελίδες που εξαρτώνται από το Ajax. Ομοίως, οι συσκευές όπως κινητά τηλέφωνα, PDAs, και screen readers μπορεί να μην υποστηρίζουν τις απαιτούμενες τεχνολογίες. Ο μόνος τρόπος για να εκμεταλλευτεί ο χρήστης την λειτουργικότητα είναι να καταφύγει σε μη JavaScript μεθόδους.



- Οι Ajax-powered διεπαφές μπορεί να αυξήσουν δραματικά τον αριθμό των αιτήσεων των χρηστών στους web servers και τα back-ends τους (βάσεις δεδομένων, ή άλλο). Αυτό μπορεί να οδηγήσει σε μεγαλύτερο χρόνο απόκρισης ή / και πρόσθετες ανάγκες υλικού.

## 5.2 Η γλώσσα προγραμματισμού C#

Η C# [15] είναι μία γλώσσα προγραμματισμού υψηλού επιπέδου. Αυτό σημαίνει ότι όταν προγραμματίζεις σε C# δεν χρειάζεται να γνωρίζεις τις λεπτομέρειες του υλικού (hardware) – ποιους καταχωρητές π.χ. προσφέρει ο επεξεργαστής, πώς γίνεται η διαχείριση της μνήμης κτλ – αλλά βασιζόμενος στο πλούσιο συντακτικό της επικεντρώνεσαι στις λειτουργίες ενός προγράμματος, όπως άλλωστε συμβαίνει και με όλες τις σύγχρονες γλώσσες προγραμματισμού. Η C# είναι σχεδιασμένη έτσι ώστε να μπορεί κάποιος να εκμεταλευτεί πλήρως τις υπηρεσίες που προσφέρει το .NET framework.

Μερικά από τα χαρακτηριστικά της C # είναι τα εξής:

- Πρόκειται για μία object-oriented γλώσσα προγραμματισμού. Αυτό σημαίνει ότι υποστηρίζει κλάσεις (classes), objects (δηλαδή instances των classes), ιεραρχίες κλάσεων, πολυμορφισμό (polymorfism), ενθυλάκωση (encapsulation), καθώς και interfaces. Επειδή πρόκειται για μία νέα γλώσσα προγραμματισμού έχει δανειστεί στοιχεία από διάφορες άλλες γλώσσες όπως C++, Java, Delphi και VB.
- Έχει συνεκτικό και καλά ορισμένο σύνολο βασικών τύπων.
- Ενσωματωμένη - υποστήριξη για αυτόματη παραγωγή XML εγγράφων.
- Αυτόματη εκκαθάριση της δυναμικά κατανεμημένης μνήμης.
- Πλήρης πρόσβαση στη .NET base class library,, καθώς και εύκολη πρόσβαση στο Windows API (εάν είναι αναγκαίο, κάτι όμως που δεν συμβαίνει πολύ συχνά)
- Δείκτες και η άμεση πρόσβαση στην μνήμη είναι διαθέσιμοι , αν απαιτείται, αλλά η γλώσσα έχει σχεδιαστεί κατά τέτοιο τρόπο ώστε να μπορεί να λειτουργήσει χωρίς αυτούς σε όλες σχεδόν τις περιπτώσεις.
- Υποστήριξη για τις ιδιότητες και τα γεγονότα στο στυλ της Visual Basic.

- Η C # μπορεί να χρησιμοποιηθεί για να γράφει κώδικας ASP.NET δυναμικών ιστοσελίδων και υπηρεσιών Web XML.

## ***5.3 Γλώσσα Μορφοποίησης Ιστοσελίδων Cascading Style***

### ***Sheets (CSS)***

Η Cascading Style Sheets (CSS) [17] είναι μια γλώσσα μορφοποίησης ιστοσελίδων που δημιουργήθηκε από το W3C, και καθορίζει το πως θα εμφανίζονται τα στοιχεία της HTML.

Γράφοντας τις σελίδες μας μόνο με HTML κώδικα, μπορούμε να ορίσουμε το χρώμα και το μέγεθος του κειμένου αλλά και άλλων στοιχείων της σελίδας (όπως πίνακες, links, λίστες κτλ). Για να αλλάξουμε το χρώμα κάποιου κειμένου ή το χρώμα ενός πίνακα, θα πρέπει να βρούμε το χρώμα αυτό μέσα στον κώδικα και να το αλλάξουμε. Η διαδικασία αυτή μπορεί να φαντάζει εύκολη όταν έχουμε να διαχειριστούμε μια μόνο σελίδα, αλλά ένα site αποτελείται από δεκάδες σελίδες τις οποίες χρειάζεται να διαχειριζόμαστε εύκολα και γρήγορα.

Με την χρήση CSS μπορούμε να ορίζουμε χρώματα και μεγέθη οργανωμένα σε στυλ και έπειτα να εφαρμόζουμε τα στυλ αυτά στα στοιχεία των σελίδων του site μας. Με αυτόν τον τρόπο, κάθε φορά που αλλάζουμε το χρώμα ενός στυλ, αλλάζει το χρώμα όλων των στοιχείων που έχουν αναφορά στο στυλ αυτό. Έτσι αν έχουμε ορίσει ένα στυλ για το κύριο μενού του site, τότε θα χρειάζεται να αλλάξουμε το χρώμα του στυλ αυτού και αυτόματα θα εφαρμοστεί σε όλες τις σελίδες.

Εκτός από την ευκολία στην διαχείριση ενός site, ένα άλλο σημαντικό πλεονέκτημα της χρήσης CSS στις σελίδες είναι ο "καθαρότερος" κώδικας, χωρίς πολλές ιδιότητες στις ετικέτες οι οποίες τον κάνουν δυσανάγνωστο. Επιπλέον κάνει γρηγορότερη την πλοήγηση καθώς το αρχείο, μέσα στο οποίο ορίζονται τα στυλ, "διαβάζεται" από τον browser μόνο μια φορά και έπειτα αποθηκεύεται στην cache memory, μειώνοντας έτσι το μέγεθος της πληροφορίας που γίνεται download από τους browsers.

## ***5.4 Microsoft SQL Server***

Ο SQL Server [13] είναι ένα Σχεσιακό Μοντέλο Διαχείρισης Δεδομένων RDBMS το οποίο αναπτύσσεται από την Microsoft. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Ο SQL Server βγήκε για πρώτη φορά στην αγορά το 1989 σε συνεργασία με την Sybase.

Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα.

#### **5.4.1 Αποθήκευση**

Η κεντρική βάση δεδομένων του SQL υποστηρίζει διαφορετικούς τύπους, συμπεριλαμβανομένων των ακεραίων αριθμών, αριθμών κινητής υποδιαστολής, δεκαδικών, αλφαριθμητικών, Varchar (σειρές χαρακτήρων μεταβλητού μήκους), δυαδικών αριθμών (για τα μη δομημένα δεδομένα), κειμένων (για κείμενα).

Επιτρέπει επίσης καθορισμένους από το χρήστη σύνθετους τύπους δεδομένων (UDTs), δηλαδή τύπους που βασίζονται στους βασικούς τύπους αλλά μπορούν να τροποποιηθούν. Τα στοιχεία στη βάση δεδομένων αποθηκεύονται σε ένα (ή περισσότερα) αρχεία με επέκταση .mdf.

Τα δευτεροβάθμια στοιχεία αποθηκεύονται στο αρχείο με επέκταση .ndf. Το αρχείο καταγραφής το οποίο περιέχει όλες τις πρόσφατες αλλαγές στη βάση δεδομένων αποθηκεύεται σε αρχείο με επέκταση .ldf.

Ο χώρος αποθήκευσης που διατίθεται σε μια βάση δεδομένων διαιρείται σε διαδοχικά αριθμημένες σελίδες, κάθε μία από τις οποίες έχει μέγεθος 8 KB.

#### **5.4.2 Ενδιάμεση μνήμη**

Οι σελίδες αποθηκεύονται στην ενδιάμεση (buffer) μνήμη RAM για να ελαχιστοποιηθεί η μεταφορά δεδομένων προς και από τον σκληρό δίσκο. Οποιαδήποτε σελίδα 8 KB μπορεί να είναι αποθηκευμένη στη μνήμη, και το σύνολο όλων των σελίδων που αποθηκεύονται σε μία περίοδο καλείται λανθάνουσα μνήμη (cache).

Το ποσό μνήμης που είναι διαθέσιμο στον κεντρικό διακομιστή SQL αποφασίζει πόσες σελίδες θα εναποθηκευθούν στη λανθάνουσα μνήμη. Ο SQL Server έχει διαφορετικούς και έξυπνους αλγόριθμους για την καλύτερη απόδοση της λανθάνουσας μνήμης.

Ο κεντρικός διακομιστής SQL εξασφαλίζει ότι οποιαδήποτε αλλαγή στα στοιχεία είναι "ACID", δηλ. χρησιμοποιεί τις "συναλλαγές" (transactions) που εξασφαλίζουν ότι οποιαδήποτε λειτουργία είτε ολοκληρώνεται συνολικά είτε αποτυγχάνει συνολικά, αλλά δεν αφήνει ποτέ τη βάση δεδομένων σε μία ενδιάμεση κατάσταση.

Χρησιμοποιώντας τις συναλλαγές αυτές, μια ακολουθία εντολών μπορεί να αποτελέσει μία ομάδα, με την προϋπόθεση ότι είτε όλες οι εντολές θα πετύχουν ή καμία δεν θα πετύχει. Ο κεντρικός διακομιστής SQL επιτρέπει σε πολλούς πελάτες να χρησιμοποιούν την ίδια βάση δεδομένων ταυτόχρονα.

Υπό αυτήν τη μορφή, θα πρέπει να ελέγξει την ταυτόχρονη πρόσβαση στα κοινά στοιχεία, για να εξασφαλίσει την ακεραιότητα των στοιχείων – Π.χ όταν πολλαπλάσιοι πελάτες ενημερώνουν τα ίδια στοιχεία, ή οι πελάτες προσπαθούν να διαβάσουν ένα στοιχείο το οποίο αλλάζεται από έναν άλλο πελάτη.

### **5.4.3 Ταυτοχρονισμός**

Ο κεντρικός διακομιστής SQL παρέχει δύο τρόπους ελέγχου του ταυτοχρονισμού: απαισιόδοξος ταυτοχρονισμός και αισιόδοξος ταυτοχρονισμός.

Όταν ο απαισιόδοξος έλεγχος ταυτοχρονισμού χρησιμοποιείται, ο κεντρικός διακομιστής SQL ελέγχει την ταυτόχρονη πρόσβαση με τη χρησιμοποίηση κλειδώματος (locks). Τα κλειδώματα μπορούν είτε να διαμοιράζονται είτε να είναι αποκλειστικά. Το αποκλειστικό κλειδωμα χορηγεί στο χρήστη την αποκλειστική πρόσβαση στα στοιχεία (συνήθως κατά την αλλαγή στοιχείων) - κανένας άλλος χρήστης δεν μπορεί να έχει πρόσβαση στα στοιχεία εφόσον υπάρχει κλειδωμα. Το κοινό κλειδωμα χρησιμοποιείται όταν διαβάζεται κάποιο στοιχείο - πολλαπλοί χρήστες μπορούν να διαβάσουν από τα στοιχεία που κλειδώνονται με ένα κοινό κλειδωμα, αλλά να μην αποκτούν αποκλειστικό.

Το κλειδωμα μπορεί να εφαρμοστεί σε διαφορετικά επίπεδα - σε ολόκληρους πίνακες, σελίδες ή ακόμα και σε διαφορετικές γραμμές. Το επίπεδο που χρησιμοποιείται καθορίζεται σε μια βάση δεδομένων από περίπλοκους αλγόριθμους του SQL Server, οι οποίοι έχουν σκοπό την γρηγορότερη και καλύτερη λειτουργία από χιλιάδες χρήστες η ακόμη και εκατοντάδες χιλιάδες την ίδια στιγμή. Ένας SQL Server είναι δυνατόν να πραγματοποιήσει μέχρι και 1.000.000 transactions το δευτερόλεπτο με το ανάλογο υλικό. Ο SQL Server λειτουργεί σε 32 η 64 bit και υποστηρίζει μέχρι και 256 πυρήνες (CPU) Τυπική χρήση είναι οι 8-16 επεξεργαστές.

### **5.4.4 Ανάκτηση δεδομένων**

Η ερώτηση (query) είναι ο κύριος τρόπος για την ανάκτηση στοιχείων από μια βάση δεδομένων. Η ερώτηση εκφράζεται χρησιμοποιώντας μια παραλλαγή της αποκαλούμενου SQL T-SQL, είναι μια διάλεκτος SQL που αναπτύχθηκε απο την Microsoft και Sybase. Η T-

SQL είναι πολύ κοντά στα ANSI standards που έχουν καθιερωθεί διεθνώς, σε αντιδιαστολή με άλλες διαλέκτους όπως η PL-SQL της Oracle που διαφέρουν περισσότερο από το ANSI standards.

Η ερώτηση διευκρινίζει επεξηγηματικά αυτό που είναι να ανακτηθεί από την βάση δεδομένων. Η ερώτηση υποβάλλεται σε επεξεργασία από τον επεξεργαστή ερώτησης, ο οποίος υπολογίζει τη σειρά των βημάτων που θα είναι απαραίτητα για να ανακτηθούν τα ζητούμενα στοιχεία. Η ακολουθία ενεργειών απαραίτητων να εκτελέσουν μια ερώτηση καλείται σχέδιο ερώτησης.

Υπάρχουν πολλοί τρόποι για να υποβληθεί σε επεξεργασία η ίδια ερώτηση. Π.χ, μια ερώτηση που ενώνει 2 πίνακες A & B μαζί θα μπορούσε να γίνει ή πηγαίνοντας πρώτα στο A βρίσκοντας τα στοιχεία και μετά ενώνοντας με τα στοιχεία του πίνακα B, ή πρώτα από το B και μετά πίνακα A. Ο SQL Server και όχι ο Προγραμματιστής παίρνει την απόφαση μέσω στατιστικών στοιχείων και άλλων στοιχείων που διαθέτει για τους 2 πίνακες να ακολουθήσει την διαδρομή A-B γιατί αυτός είναι ο γρηγορότερος τρόπος. Στις procedural γλώσσες προγραμματισμού πχ C, Pascal, .NET, ο προγραμματιστής θα πρέπει να κάνει την απόφαση το οποίο πολλές φορές σημαίνει λάθος απόφαση. Αυτό καλείται βελτιστοποίηση ερώτησης και εκτελείται από τον επεξεργαστή ερώτησης. Ο SQL παίρνει πολύ περίπλοκες αποφάσεις βασισμένος σε αλγόριθμους που έχουν αναπτυχθεί τα τελευταία 20-30 χρόνια πάνω στο αντικείμενο. Μια παρόμοια ερώτηση σε μια προηγούμενη έκδοση του SQL Server θα μπορούσε να πάρει 10-20 φορές περισσότερο χρόνο λόγω του διαφορετικού αλγόριθμου. Γενικά κάθε καινούργια έκδοση βελτιώνει την αποτελεσματικότητα, και άρα τον χρόνο εκτέλεσης. Εδώ ακριβώς είναι ένα από τα πλεονεκτήματα προγραμματισμού ενός προβλήματος σε μια βάση δεδομένων σε σύγκριση με C, Pascal, C++ κτλ. Βέβαια όλα τα προβλήματα δεν είναι κατάλληλα για εφαρμογή σε βάση δεδομένων.

Ο SQL Server επιτρέπει επίσης αποθηκευμένες ερωτήσεις (stored procedures). Οι αποθηκευμένες ερωτήσεις είναι ερωτήσεις με παραμέτρους T-SQL, οι οποίες αποθηκεύονται στον SQL Server. Μια stored procedure τυπικά είναι από 100-1000 γραμμές κώδικα SQL. Οι αποθηκευμένες ερωτήσεις μπορούν να δεχτούν τιμές που στέλνονται από τον πελάτη ως παράμετροι εισαγωγής, και να στείλουν τα αποτελέσματα ως παραμέτρους παραγωγής. Μπορούν να καλέσουν functions, και άλλες stored procedures. Οι stored procedures είναι γρηγορότερες από κατασκευή σε σχέση με απλές ακολουθίες SQL.

#### 5.4.5 CLR

Ο Microsoft SQL Server 2005 – 2008 περιλαμβάνει ένα module που λέγεται SQL CLR μέσω του οποίου ενσωματώνει το .NET μέσα στον SQL Server.

Με το SQL CLR, οι αποθηκευμένες διαδικασίες μπορούν να γραφτούν σε οποιαδήποτε γλώσσα .NET συμπεριλαμβανομένου C# και VB.NET και να δημιουργήσουν μια stored procedure. Αυτό σημαίνει ότι ο SQL Server έχει όλες τις βιβλιοθήκες και πλεονεκτήματα του .NET, αυτόχθονα μέσα στο περιβάλλον του, τα οποία μπορεί να τα καλέσει οποιαδήποτε στιγμή.

#### 5.4.6 Υπηρεσίες

- Ο Μεσίτης Υπηρεσιών (Service Broker) που τρέχει ως μέρος της μηχανής βάσεων δεδομένων, παρέχει μια αξιόπιστη πλατφόρμα μηνυμάτων και αναμονής μηνυμάτων για τις εφαρμογές κεντρικών SQL Server. Ο Service Broker έχει ανώτερες δυνατότητες προγραμματισμού στο μέλλον (scheduling).
- Οι υπηρεσίες συγχρονισμού (Replication) του SQL Server χρησιμοποιούνται από τον SQL Server για και να συγχρονίσουν βάσεις δεδομένων, είτε ολοκληρωτικά είτε ένα υποσύνολο των αντικειμένων. Υπάρχουν τουλάχιστον 4 διαφορετικοί μέθοδοι (log shipping, mirroring, replication, back-ups, clustering) για τον συγχρονισμό. Αυτό βοηθάει επίσης και σε περιπτώσεις ολοκληρωτικής καταστροφής της μητρικής βάσης δεδομένων, δίνοντας την δυνατότητα ακόμη και για αυτόματη λειτουργία της δευτερεύουσας συγχρονισμένης βάσης δεδομένων.
- Οι υπηρεσίες ανάλυσης (Analysis Services) SQL Server προσθέτουν τις ικανότητες OLAP και περίπλοκης ανάκτησης δεδομένων για τις βάσεις δεδομένων SQL, με περίπλοκη ανάλυση χωρίς την ανάγκη για δύσκολες SQL queries. Υποστηρίζεται το MOLAP, ROLAP και HOLAP για τα στοιχεία.
- Ο SQL Server έχει υπηρεσίες Reporting για στοιχεία που συγκεντρώνονται από τις βάσεις δεδομένων SQL Server. Με αυτόν τον τρόπο είναι δυνατό να παραχθεί μια έκθεση Ιστού (Web Report) σχεδόν αυτόματα χωρίς προγραμματισμό HTML, CSS ή JavaScript, μέσα από το Visual Studio.
- Οι υπηρεσίες ανακοίνωσης (Notification Services) δίνουν την δυνατότητα στον χρήστη να παράγει μηνύματα ηλεκτρονικού ταχυδρομείου με πολύ λίγο κόπο.<sup>[5] [6]</sup>

- Το Integration Services του SQL χρησιμοποιούνται για να ενσωματώσουν στοιχεία από διαφορετικές πηγές δεδομένων. Χρησιμοποιεί τις ικανότητες ETL. Οι Integration Services περιλαμβάνουν εργαλεία GUI για να χτίσουν διάφορες λειτουργίες ροών δεδομένων, όπως η εξαγωγή στοιχείων από διάφορες πηγές, η συζήτηση στοιχείων, ο μετασχηματισμός των στοιχείων συμπεριλαμβανομένης της συνάθροισης, του διπλασιασμού και η συγχώνευσης.
- Ο SQL Server μπορεί επίσης να αναζητήσει μέσα σε κείμενα (Full Text Search Service ). Ο δείκτης αναζήτησης μπορεί να δημιουργηθεί σε οποιαδήποτε στήλη και έτσι η αναζήτηση πραγματοποιείται άμεσα. Για παράδειγμα, κείμενα Word και έγγραφα pdf αποθηκευμένα στον SQL Server μπορούν αναζητηθούν πολύ γρήγορα.

# 6

## Βάση δεδομένων της εφαρμογής

### 6.1 Σχέσεις της βάσης δεδομένων της εφαρμογής


Κατά τον σχεδιασμό της βάσης δεδομένων της συγκεκριμένης ιστοσελίδας δημιουργήθηκαν οι παρακάτω οντότητες:

#### Σχέση `dbo.ATC_Table`

Γνωρίσματα (attributes)	Περιγραφή
<b>ATC_Id</b>	Η ATC κωδικοποίηση, αποτελεί το πρωτεύον κλειδί της σχέσης.
<b>Ingredient</b>	Το όνομα της κύριας δραστικής ουσίας.

Πίνακας 6.1 -attributes της σχέσης `dbo.ATC_Table`

Στην σχέση `dbo.ATC_Table` αποθηκεύονται όλοι οι κωδικοί ATC με την κύρια δραστική ουσία τους.

	Column Name	Data Type	Allow Nulls
	ATC_Id	nchar(10)	<input type="checkbox"/>
	Ingredient	nvarchar(MAX)	<input checked="" type="checkbox"/>

Εικόνα 6.1 - τύποι πεδίων της σχέσης `dbo.ATC_Table`





### Σχέση dbo.ATC\_DRUG\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>ATC_Id, DRUG_Id</b>	Αποτελούν το κύριο κλειδί του πίνακα αυτού, μέσω του οποίου συσχετίζεται ο ATC_Table με τον DRUG_Table.

Πίνακας 6.2 -attributes της σχέσης dbo.ATC\_DRUG\_Table

Η σχέση dbo. ATC\_DRUG\_Table συσχετίζει τις οντότητες dbo.ATC\_Table και dbo.DRUG\_Table.

	Column Name	Data Type	Allow Nulls
	DRUG_Id	int	<input type="checkbox"/>
	ATC_Id	nchar(10)	<input type="checkbox"/>



Εικόνα 6.2 - τύποι πεδίων της σχέσης dbo ATC\_DRUG\_Table

### Σχέση dbo. ATC\_DrugStore\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>ATC_Id, DRUG_Id</b>	Αποτελούν το κύριο κλειδί του πίνακα αυτού, μέσω του οποίου συσχετίζεται ο ATC_Table με τον DrugStore_Table.

Πίνακας 6.3 -attributes της σχέσης dbo.ATC\_DrugStore\_Table

Η σχέση dbo.ATC\_DrugStore\_Table συσχετίζει τις οντότητες dbo.ATC\_Table και dbo.DrugStore\_Table.

	Column Name	Data Type	Allow Nulls
	ATC_Id	nchar(10)	<input type="checkbox"/>
	Drug_Id	int	<input type="checkbox"/>



Εικόνα 6.3- τύποι πεδίων της σχέσης dbo.ATC\_DrugStore\_Table

### Σχέση dbo.ATC\_ICD\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>ATC_Id, ICD_Id</b>	Αποτελούν το κύριο κλειδί του πίνακα αυτού, μέσω του οποίου συσχετίζεται ο ATC_Table με τον ICD_Table.

Πίνακας 6.4-attributes της σχέσης dbo.ATC\_ICD\_Table

Η σχέση dbo.ATC\_ICD\_Table συσχετίζει τις οντότητες dbo.ATC\_Table και dbo.ICD\_Table.

	Column Name	Data Type	Allow Nulls
	ATC_Id	nchar(10)	<input type="checkbox"/>
	ICD_Id	nchar(10)	<input type="checkbox"/>



Εικόνα 6.4- τύποι πεδίων της σχέσης dbo.ATC\_ICD\_Table

### Σχέση dbo.CATEGORY\_ATC\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>ATC_Id</b>	Η ATC κωδικοποίηση.
<b>CATEGORY_Name</b>	Προσδιορίζει κατηγορίες φαρμάκων.
Μαζί αποτελούν το κύριο κλειδί του πίνακα αυτού, μέσω του οποίου συσχετίζεται με τον ATC_Table.	

Πίνακας 6.5 -attributes της σχέσης dbo.CATEGORY\_ATC\_Table

Η σχέση dbo.CATEGORY\_ATC\_Table προσδιορίζει τις κατηγορίες των φαρμάκων.

	Column Name	Data Type	Allow Nulls
	CATEGORY_Name	varchar(60)	<input type="checkbox"/>
	ATC_Id	nchar(10)	<input type="checkbox"/>

Εικόνα 6.5- τύποι πεδίων της σχέσης dbo. CATEGORY\_ATC\_Table

### Σχέση dbo.CATEGORY\_ATC\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>Drug_Id</b>	Είναι ο μοναδικός αριθμός αναγνώρισης κάθε ξεχωριστού φαρμάκου.
<b>Quantity</b>	Προσδιορίζει την ποσότητα του φαρμάκου στην αποθήκη του νοσοκομείου.

Πίνακας 6.6-attributes της σχέσης dbo. DrugStore\_Table

Στην σχέση dbo. DrugStore\_Table αποθηκεύονται όλα τα φάρμακα που υπάρχουν στην αποθήκη του νοσοκομείου.

	Column Name	Data Type	Allow Nulls
	Drug_Id	int	<input type="checkbox"/>
	Quantity	int	<input type="checkbox"/>


Εικόνα 6.6- τύποι πεδίων της σχέσης dbo. DrugStore\_Table

### Σχέση dbo.DRUG\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>DRUG_Id</b>	Είναι ο μοναδικός αριθμός αναγνώρισης κάθε ξεχωριστού φαρμάκου, αποτελεί το πρωτεύον κλειδί της σχέσης και έχει τεθεί auto increment.
<b>DRUG_Name</b>	Είναι το όνομα του φαρμάκου.
<b>DRUG_Supplier</b>	Είναι το όνομα της εταιρίας-προμηθευτή του φαρμάκου.
<b>DRUG_Dosage</b>	Η δοσολογία του φαρμάκου.
<b>DRUG_Package</b>	Η συσκευασία του φαρμάκου.
<b>ROUTE_Id</b>	Προσδιορίζει το χαρακτηριστικό κωδικό της οδού χορήγησης και του τύπου του συγκεκριμένου φαρμάκου.

Πίνακας 6.7-attributes της σχέσης dbo.DRUG\_Table

Στην σχέση dbo.DRUG\_Table αποθηκεύονται όλα τα φάρμακα του Ε.Ο.Φ με τα χαρακτηριστικά τους.

	Column Name	Data Type	Allow Nulls
	DRUG_Id	int	<input type="checkbox"/>
	DRUG_Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
	DRUG_Supplier	nvarchar(MAX)	<input checked="" type="checkbox"/>
	DRUG_Dosage	nvarchar(MAX)	<input checked="" type="checkbox"/>
	DRUG_Package	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ROUTE_Id	int	<input checked="" type="checkbox"/>


Εικόνα 6.7- τύποι πεδίων της σχέσης dbo. DRUG\_Table

### Σχέση dbo. CONTAINER\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>Drug_Id</b>	Είναι ο μοναδικός αριθμός αναγνώρισης κάθε ξεχωριστού φαρμάκου.
<b>Quantity</b>	Προσδιορίζει την ποσότητα του φαρμάκου στο container.

Πίνακας 6.8-attributes της σχέσης dbo. CONTAINER\_Table

Στην σχέση dbo. DrugStore\_Table αποθηκεύονται όλα τα φάρμακα που υπάρχουν στο container.

	Column Name	Data Type	Allow Nulls
	Drug_Id	int	<input type="checkbox"/>
	Quantity	int	<input type="checkbox"/>

Εικόνα 6.8- τύποι πεδίων της σχέσης dbo. CONTAINER\_Table




### Σχέση dbo.DrugsPerVisit\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>Visit_Id</b>	Είναι ο μοναδικός αριθμός αναγνώρισης κάθε ξεχωριστού φαρμάκου.

<b>Drug_Id</b>	Προσδιορίζει το χαρακτηριστικό κωδικό κάθε επίσκεψης κάθε ασθενούς.
<b>Drug_Quantity</b>	Προσδιορίζει την ποσότητα του φαρμάκου που συνταγογράφησε ο γιατρός.
Το Visit_Id με το Drug_Id αποτελούν το κύριο κλειδί του πίνακα αυτού, μέσω του οποίου συσχετίζεται με τον VISITS_Table.	

Πίνακας 6.9-attributes της σχέσης dbo.DrugsPerVisit\_Table

Στην σχέση dbo.DrugsPerVisit\_Table αποθηκεύονται τα φάρμακα ανά επίσκεψη καθώς και η συνταγογραφούμενη ποσότητα για το καθένα.

	Column Name	Data Type	Allow Nulls
	Visit_Id	int	<input type="checkbox"/>
	Drug_Id	int	<input type="checkbox"/>
	Drug_Quantity	int	<input type="checkbox"/>


Εικόνα 6.9- τύποι πεδίων της σχέσης dbo.DrugsPerVisit\_Table

### Σχέση dbo. DrugOrder\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>Drug_Id</b>	Είναι ο μοναδικός αριθμός αναγνώρισης κάθε ξεχωριστού φαρμάκου.
<b>Quantity</b>	Προσδιορίζει την ποσότητα της παραγγελίας του φαρμάκου.

Πίνακας 6.10-attributes της σχέσης dbo. DrugOrder\_Table

Στην σχέση dbo. DrugStore\_Table αποθηκεύεται η ποσότητα της παραγγελίας των φαρμάκων.

	Column Name	Data Type	Allow Nulls
	Drug_Id	int	<input type="checkbox"/>
	Quantity	int	<input type="checkbox"/>


Εικόνα 6.10- τύποι πεδίων της σχέσης dbo. DrugOrder\_Table

### Σχέση dbo. ICD\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>ICD_Id</b>	Η ICD-10 κωδικοποίηση, αποτελεί το πρωτεύον κλειδί της σχέσης.
<b>Diseases</b>	Προσδιορίζει το όνομα της αντίστοιχης ασθένειας.

Πίνακας 6.11-attributes της σχέσης dbo.ICD\_Table

Στην σχέση dbo. ICD\_Table αποθηκεύονται όλες οι ICD-10 ασθένειες με τα ονόματά τους.

	Column Name	Data Type	Allow Nulls
	ICD_Id	nchar(10)	<input type="checkbox"/>
	Diseases	nvarchar(MAX)	<input checked="" type="checkbox"/>

Εικόνα 6.11- τύποι πεδίων της σχέσης dbo.ICD\_Table

### Σχέση dbo. LOGIN\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>Username</b>	Προσδιορίζει το όνομα με το οποίο συνδέεται ο χρήστης στην σελίδα.  (doctor ή warehouseman)
<b>Password</b>	Προσδιορίζει τον χαρακτηριστικό κωδικό που πρέπει να επιβεβαιώσει ο χρήστης για να πιστοποιηθεί η ταυτότητά του

Πίνακας 6.12-attributes της σχέσης dbo. LOGIN\_Table

Στην σχέση dbo. LOGIN\_Table αποθηκεύονται όλοι οι χρήστες που έχουν πρόσβαση στο site.

	Column Name	Data Type	Allow Nulls
	Username	nvarchar(MAX)	<input type="checkbox"/>
	Password	nvarchar(MAX)	<input type="checkbox"/>







Εικόνα 6.12- τύποι πεδίων της σχέσης dbo.LOGIN\_Table

### Σχέση dbo. PATIENT\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>Patient_Id</b>	Είναι ο μοναδικός αριθμός αναγνώρισης κάθε ξεχωριστού ασθενούς, αποτελεί το πρωτεύον κλειδί της σχέσης και έχει τεθεί auto increment.
<b>ID_Number</b>	Είναι ο αριθμός της ταυτότητας του ασθενούς.
<b>Patient_Name</b>	Είναι το ονοματεπώνυμο του ασθενούς.
<b>Gender</b>	Προσδιορίζει το φύλο του ασθενούς (male ή female).
<b>Birth_Date</b>	Ημερομηνία γέννησης του ασθενούς.
<b>Phone_Number</b>	Ο αριθμός τηλεφώνου του ασθενούς.

Πίνακας 6.13-attributes της σχέσης dbo. PATIENT\_Table

Στην σχέση dbo. PATIENT\_Table αποθηκεύονται οι πληροφορίες των ασθενών.

	Column Name	Data Type	Allow Nulls
	Patient_Id	int	<input type="checkbox"/>
	ID_Number	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Patient_Name	nvarchar(MAX)	<input type="checkbox"/>
	Gender	nchar(10)	<input checked="" type="checkbox"/>
	Birth_Date	nchar(10)	<input checked="" type="checkbox"/>
	Phone_Number	nvarchar(MAX)	<input checked="" type="checkbox"/>

Εικόνα 6.13- τύποι πεδίων της σχέσης dbo.PATIENT\_Table


#### Σχέση dbo. ROUTE\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>ROUTE_Id</b>	Προσδιορίζει το χαρακτηριστικό κωδικό της οδού χορήγησης και του τύπου του συγκεκριμένου φαρμάκου, αποτελεί το πρωτεύον κλειδί της σχέσης και έχει τεθεί auto increment.
<b>ROUTE_Code</b>	Αποτελεί την συντομογραφία του τύπου του φαρμάκου.

<b>ROUTE_Name</b>	Αποτελεί την αναλυτική ονομασία του τύπου φαρμάκου.
<b>ROUTE_Type</b>	Προσδιορίζει την οδό χορήγησης του φαρμάκου.

Πίνακας 6.14-attributes της σχέσης dbo. ROUTE\_Table

Στην σχέση dbo. ROUTE\_Table αποθηκεύονται οι πληροφορίες της οδού χορήγησης και του τύπου του συγκεκριμένου φαρμάκου.

	Column Name	Data Type	Allow Nulls
	ROUTE_Id	int	<input type="checkbox"/>
	ROUTE_Code	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ROUTE_Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ROUTE_Type	nvarchar(MAX)	<input checked="" type="checkbox"/>


Εικόνα 6.14- τύποι πεδίων της σχέσης dbo. ROUTE\_Table

#### Σχέση dbo.TreeView\_Table

Γνωρίσματα (attributes)	Περιγραφή
<b>id</b>	Αποτελεί το πρωτεύον κλειδί της σχέσης και έχει τεθεί auto increment.
<b>atc</b>	Η κωδικοποίηση ATC με δενδρική δομή πέντε επιπέδων με τον τίτλο τους.

Πίνακας 6.15-attributes της σχέσης dbo.TreeView\_Table

Στην σχέση dbo.TreeView\_Table αποθηκεύονται όλοι οι κωδικοί ATC με τους τίτλους τους σε δενδρική δομή.

	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	atc	varchar(MAX)	<input type="checkbox"/>

Εικόνα 6.15- τύποι πεδίων της σχέσης dbo.TreeView\_Table


#### Σχέση dbo.VISITS\_Table



Γνωρίσματα (attributes)	Περιγραφή
<b>Visit_Id</b>	Προσδιορίζει το χαρακτηριστικό κωδικό κάθε επίσκεψης κάθε ασθενούς, αποτελεί το πρωτεύον κλειδί της σχέσης και έχει τεθεί auto increment.
<b>Patient_Id</b>	Είναι ο μοναδικός αριθμός αναγνώρισης κάθε ξεχωριστού ασθενούς.
<b>ICD_Id</b>	Η ICD-10 κωδικοποίηση της ασθένειας του ασθενούς.
<b>Prescription_Date</b>	Ορίζει την ημερομηνία συνταγογράφησης.

Πίνακας 6.16-attributes της σχέσης dbo.VISITS\_Table

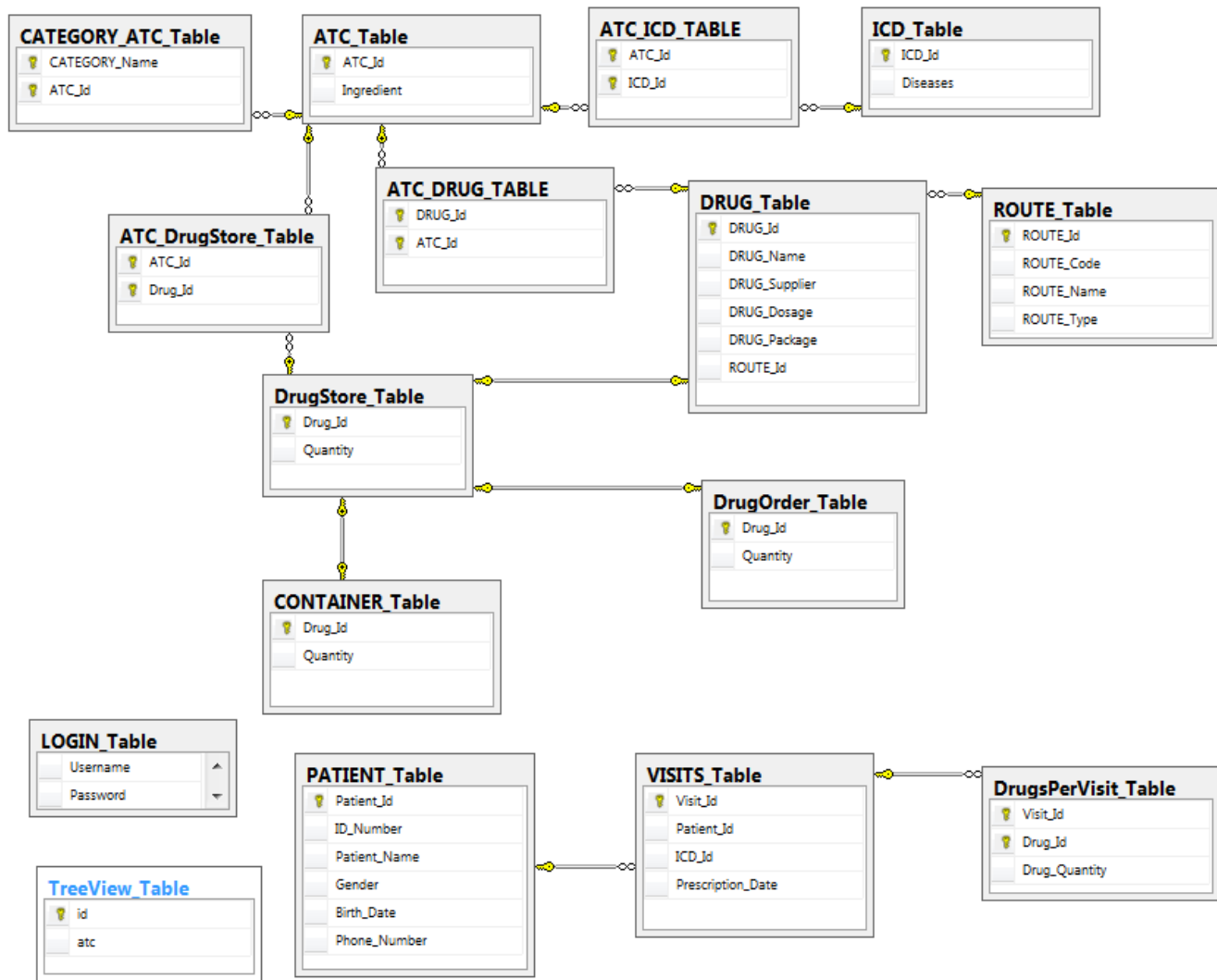
Στην σχέση dbo.VISITS\_Table αποθηκεύονται οι πληροφορίες κάθε επίσκεψης κάθε ασθενούς.

	Column Name	Data Type	Allow Nulls
	Visit_Id	int	<input type="checkbox"/>
	Patient_Id	int	<input type="checkbox"/>
	ICD_Id	nchar(10)	<input type="checkbox"/>
	Prescription_Date	nchar(10)	<input type="checkbox"/>

Εικόνα 6.16- τύποι πεδίων της οντότητας dbo.VISITS\_Table

## 6.2 Σχεσιακό Μοντέλο

Το σχεσιακό μοντέλο της βάσης δεδομένων φαίνεται παρακάτω :



### 6.3 Περιεχόμενα της βάσης δεδομένων

Η βάση δεδομένων περιέχει το σύνολο των παρακάτω εγγραφών:

- 4123 διαφορετικές ATC κωδικοποιήσεις με το όνομα των κύριων δραστικών ουσιών τους.
- 9254 διαφορετικές ICD κωδικοποιήσεις με το αντίστοιχο όνομα της ασθένειας.
- 4665 έγγραφες φαρμάκων μαζί με χαρακτηριστικά όπως το όνομα, τον προμηθευτή, την δοσολογία, το είδος της συσκευασίας και την οδό χορήγησης του φαρμάκου [18].
- 416 διαφορετικές συσκευασίες φαρμάκων.
- 20 διαφορετικοί οδοί χορήγησης.
- 471 διαφορετικοί κατηγορίες φαρμάκων (σύμφωνα με την DrugBank).

# 7

## *Η Ολοκληρωμένη Εφαρμογή Διαχείρισης Ηλεκτρονικού Φαρμακείου*

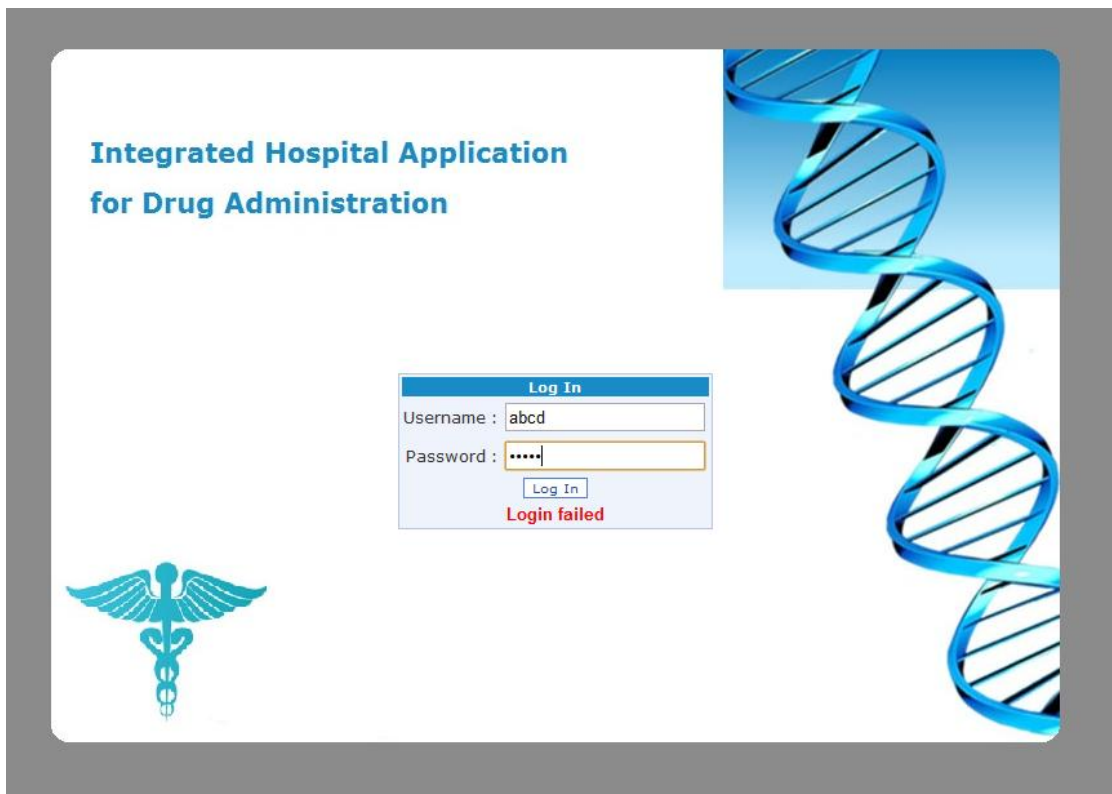
### *7.1 Σενάριο σύνδεσης χρήστη στην ιστοσελίδα*

#### *7.1.1 Σελίδα Login.aspx*

Κατά την επίσκεψη στην ιστοσελίδα η πρώτη οθόνη που συναντά ο χρήστης είναι η οθόνη εισόδου (Login.aspx), όπου του ζητείται να εισάγει το προσωπικό του όνομα χρήστη (Username) και κωδικό χρήστη (Password) για να εισέλθει στην ιστοσελίδα.



Εικόνα 7.1– Η σελίδα Login.aspx στην αρχική της μορφή



Εικόνα 7.2– Ανεπιτυχές Login και μήνυμα σφάλματος

### **Επιτυχημένο σενάριο**

- Ο χρήστης εισάγει τα προσωπικά του στοιχεία στα απαιτούμενα πεδία.
- Ο server επικοινωνεί με την βάση δεδομένων επιβεβαιώνοντας ότι τα στοιχεία ανήκουν σε χρήστη και τον βαθμό πρόσβασης του χρήστη (γιατρός ή αποθηκάριος).
- Ο server στέλνει πίσω στον χρήστη την αρχική οθόνη Patients.aspx εάν είναι γιατρός ή Warehouse.aspx εάν είναι αποθηκάριος.

### **Αποτυχημένο σενάριο**

- Ο χρήστης εισάγει τα προσωπικά του στοιχεία στα απαιτούμενα πεδία.
- Ο server επικοινωνεί με την βάση δεδομένων όμως τα στοιχεία που έχει εισάγει ο χρήστης (password ή username) δεν υπάρχουν καταχωρημένα μέσα στην βάση.
- Επιστρέφεται μήνυμα στον επισκέπτη ότι έχει εισάγει λάθος στοιχεία (Login failed).

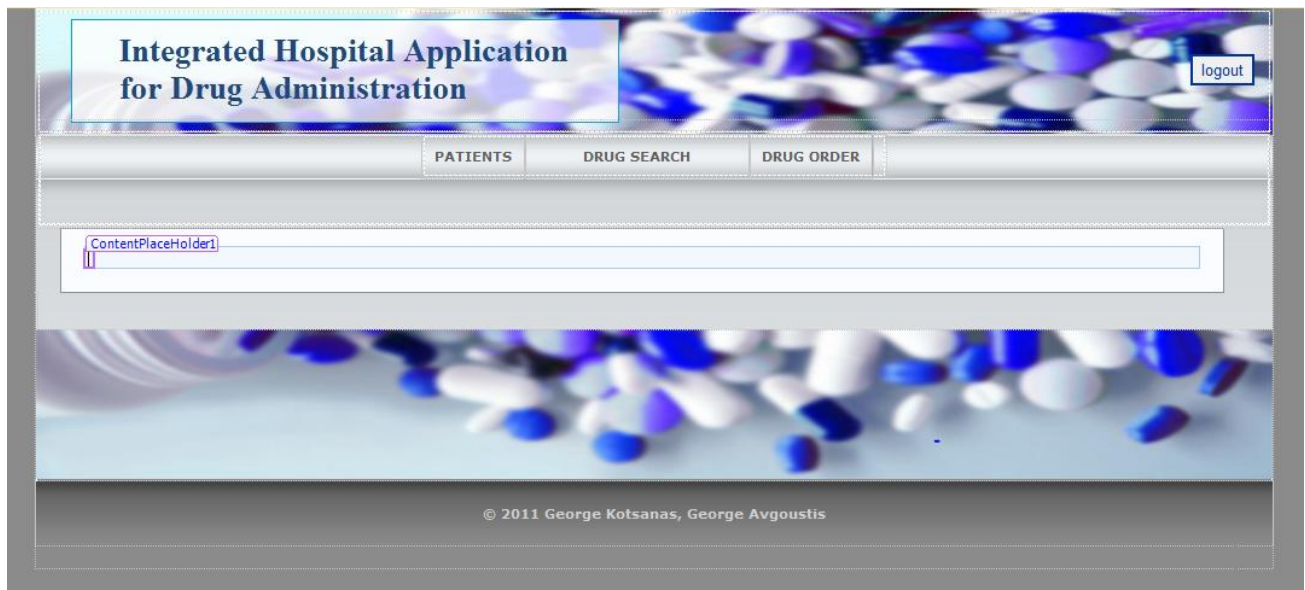
Με την εισαγωγή σωστού κωδικού, δημιουργείται Session με τιμή αντίστοιχη του σεναρίου χρήσης. Κατά την φόρτωση κάθε μιας από τις σελίδες που ακολουθούν ελέγχεται αν η τιμή του Session είναι η κατάλληλη αλλιώς ο χρήστης προτρέπεται για login στην αρχική σελίδα.

## **7.2 Σενάριο λογαριασμού γιατρού**

### **7.2.1 Σελίδα DoctorMasterPage.Master**

Στο σενάριο χρήσης του γιατρού χρησιμοποιήσαμε την παρακάτω εικονιζόμενη masterpage.

Master page είναι το κεντρικό template που θα συναντά ο γιατρός σε όλες τις σελίδες του ενώ στα placeholders (συγκεκριμένα στο ContentPlaceHolder1) θα εμφανίζονται τα περιεχόμενα της εκάστοτε σελίδας.

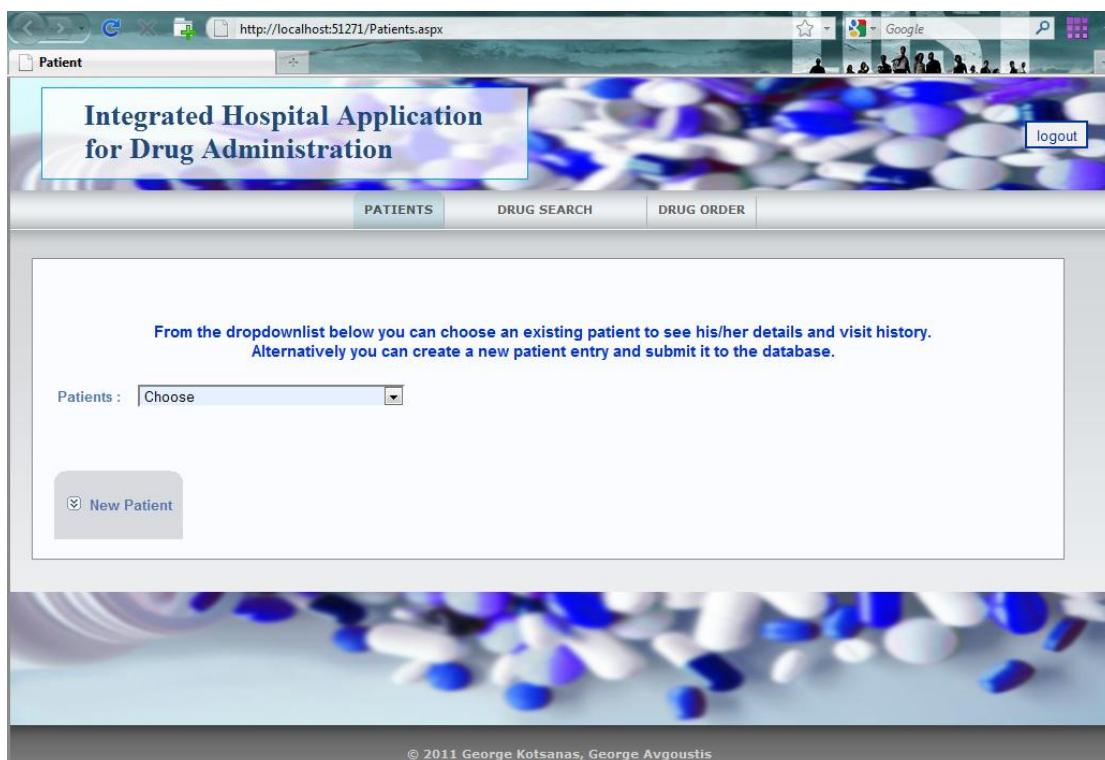


Εικόνα 7.3– Το masterpage για το σενάριο γιατρού

Επιπλέον, στα masterpage βρίσκεται υπάρχει το κουμπί `logout` με το οποίο ο χρήστης μπορεί να αποσυνδεθεί και αυτομάτως να προωθηθεί στην αρχική σελίδα της εφαρμογής `Login.aspx`.

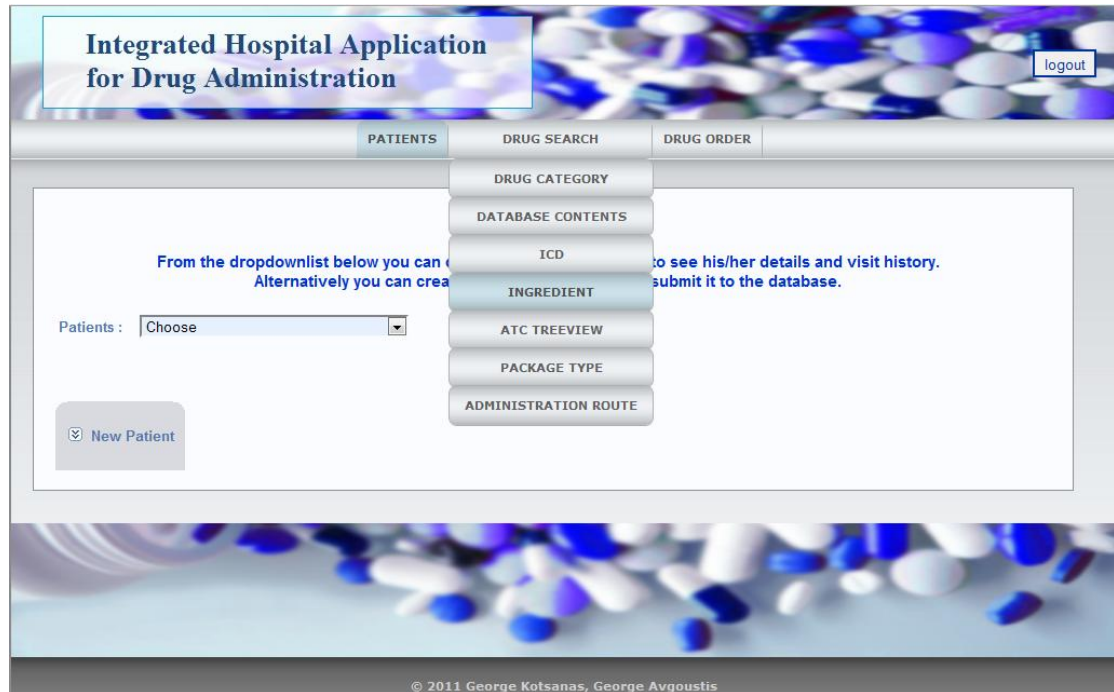
### 7.2.2 Οριζόντιο μενού πλοήγησης

Το οριζόντιο μενού πλοήγησης έχει δημιουργηθεί αποκλειστικά με CSS και ανάλογα με ποια σελίδα `aspx` βρίσκεται ο χρήστης, το αντίστοιχο `tab` ενεργοποιείται (γίνεται πιο έντονο). Αυτό έχει επιτευχθεί με χρήση JavaScript. Το μενού φαίνεται παρακάτω :



Εικόνα 7.4– Η σελίδα `Patients.aspx` με ενεργοποιημένο το αντίστοιχο `tab` (`PATIENTS`)

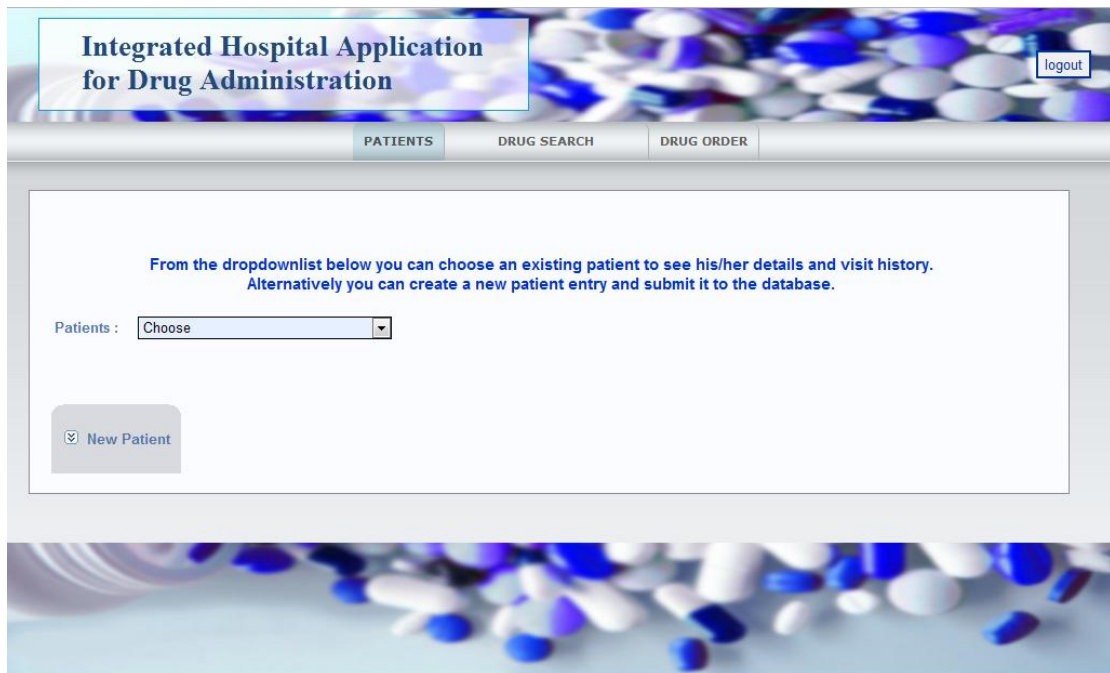
Όταν ο χρήστης τοποθετήσει το δείκτη του ποντικιού πάνω στο tab DRUG SEARCH, ανοίγει το υπομενού του, όπως αυτό φαίνεται παρακάτω. Επιλέγοντας κάποιο tab ή στοιχείο του υπομενού του tab DRUG SEARCH, ο χρήστης μεταβαίνει στην αντίστοιχη ιστοσελίδα.



Εικόνα 7.5– Εμφάνιση του μενού του tab DRUG SEARCH

### 7.2.3 Σελίδα *Patients.aspx*

Αφού γίνει login στην σελίδα Login.aspx με το username και το password του γιατρού ο χρήστης μεταβαίνει αυτόματα στην σελίδα Patients.aspx. Παρακάτω φαίνεται η αρχική σελίδα του προφίλ γιατρός.

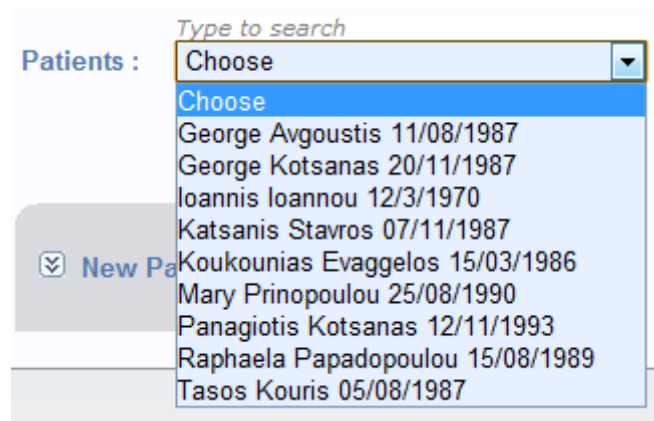


Εικόνα 7.6– Η σελίδα Patients.aspx στην αρχική της μορφή

Ο γιατρός έχει δυο δυνατότητες:

- Να επιλέξει κάποιον καταχωρημένο ασθενή από το Dropdownlist (είναι ενσωματωμένος ο ListSearchExtender για γρήγορη επιλογή-αναζήτηση πληκτρολόγηση των πρώτων χαρακτήρων)
- Να πραγματοποιήσει νέα εγγραφή ασθενούς

Παρακάτω φαίνονται οι υπάρχοντες ασθενείς. Γίνεται η ανάκτηση τους από την βάση δεδομένων και ταξινόμηση τους κατά αλφαβητική σειρά. Για αποφυγή σύγχυσης μεταξύ ασθενών με συνωνυμία αναγράφεται δίπλα στο όνομα και η ημερομηνία γέννησης.

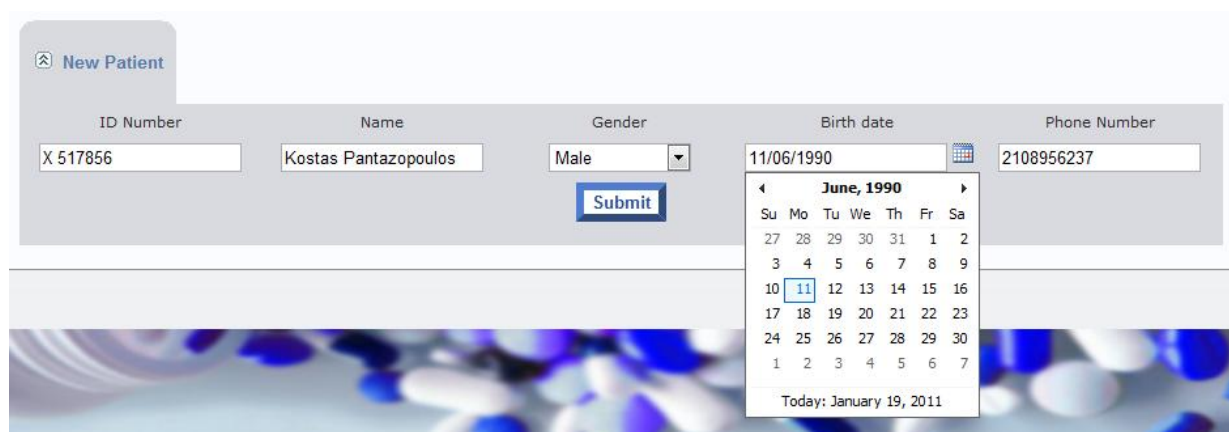


Εικόνα 7.7– Πάτημα του DropDownList και εμφάνιση της λίστας των ασθενών



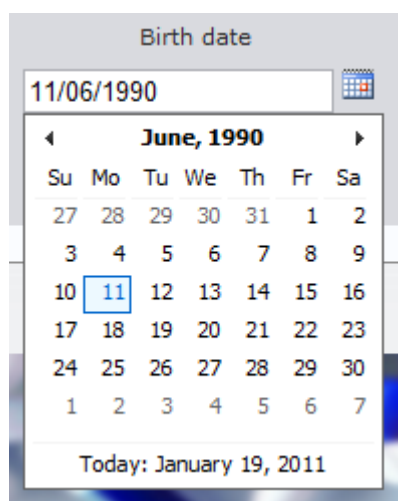
Για την δημιουργία εγγραφής νέου ασθενούς και εισαγωγή του στην βάση ο γιατρός πρέπει να εισάγει τα εξής στοιχεία (βλ. Εικόνα ):

- Αριθμός Ταυτότητας (ID Number)
- Ονοματεπώνυμο (Name)
- Γένος (Gender- Male or Female)
- Ημερομηνία Γέννησης (Birth date)
- Αριθμός Τηλεφώνου (Phone Number)




The image shows a web form titled "New Patient" with the following fields: ID Number (X 517856), Name (Kostas Pantazopoulos), Gender (Male), Birth date (11/06/1990), and Phone Number (2108956237). A "Submit" button is located below the Gender field. A calendar popup is open over the Birth date field, showing the month of June 1990. The date 11 is selected. The calendar also shows the current date as January 19, 2011.

Εικόνα 7.8– Εισαγωγή νέας εγγραφής ασθενούς



A close-up of the calendar popup from the previous image. It shows the month of June 1990. The date 11 is selected. The calendar also shows the current date as January 19, 2011.

Στην εικόνα χρησιμοποιείται ένα στοιχείο του Ajax Control Toolkit που ονομάζεται Calendar extender το οποίο επιτρέπει στους χρήστες να επιλέγουν γρήγορα, μια ημερομηνία χρησιμοποιώντας ένα επαγγελματικής εμφάνισης date picker με σύστημα διαχείρισης γεγονότων, client-side. Όταν ο χρήστης πατήσει το εικονίδιο  εμφανίζεται το pop up παράθυρο, όπου η συγκεκριμένη ημερομηνία που θα επιλεγεί θα αναγραφεί στο διπλανό textbox.

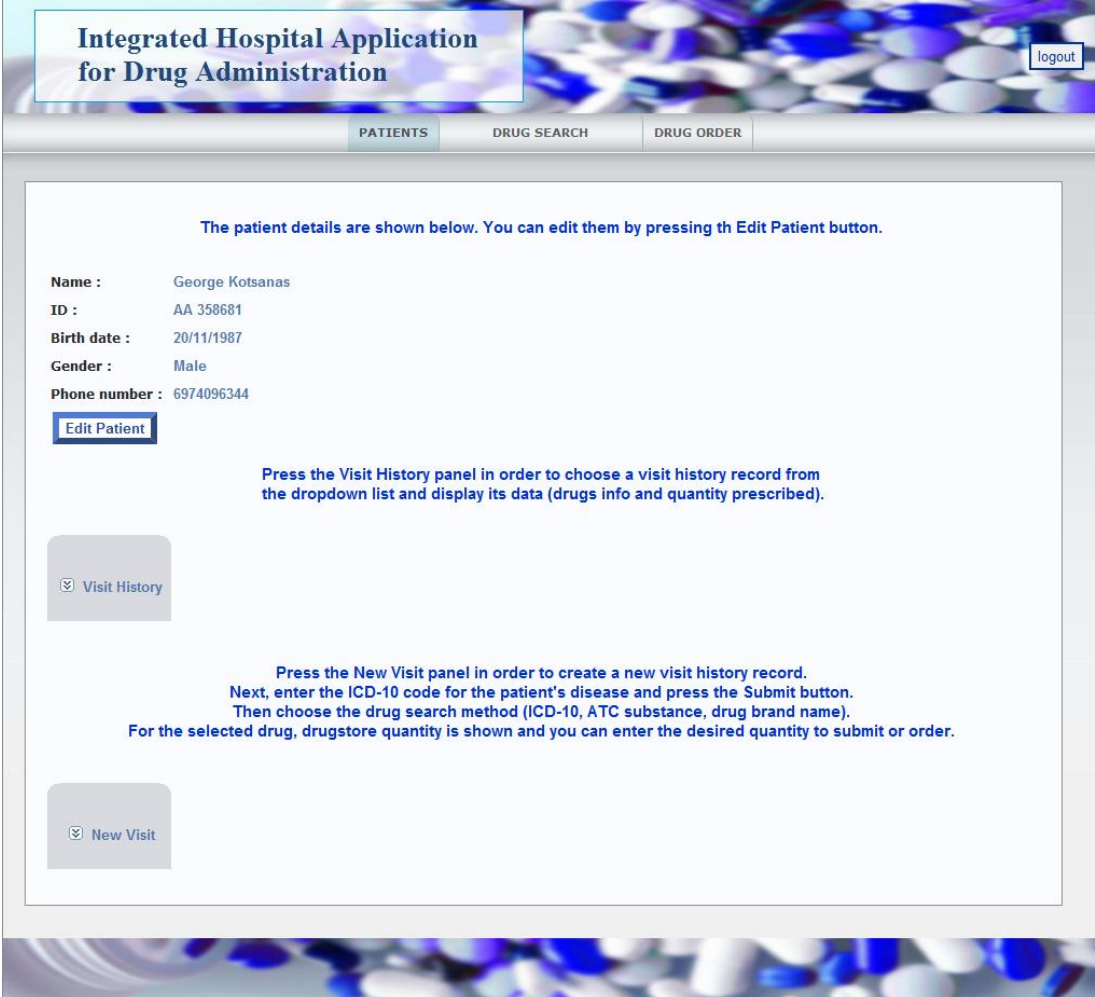
Εικόνα 7.9– Calendar extender

Με την επιλογή ενός ασθενούς ή την δημιουργία ενός νέου, το πρόγραμμα αυτομάτως προωθεί τον γιατρό στην σελίδα Visit\_History.aspx. Η προώθηση γίνεται με την μέθοδο GET π.χ. [http://localhost:51271/Visit\\_History.aspx?Patient\\_Id=10](http://localhost:51271/Visit_History.aspx?Patient_Id=10). Όπου Patient\_Id είναι ο μοναδικός αριθμός αναγνώρισης (κύριο κλειδί) του ασθενούς που έχει επιλεγεί και ανακτηθεί από την βάση δεδομένων.

## 7.2.4 Σελίδα *Visit\_History.aspx*

Η σελίδα αυτή περιέχει πληροφορίες και αρχείο ιστορικού για τον επιλεγμένο ασθενή. Ο γιατρός μπορεί είτε να εποπτεύσει κάποια προηγούμενη επίσκεψη του ασθενούς είτε να δημιουργήσει μια καινούρια.

Η σελίδα αυτή συνδέεται άμεσα με την σελίδα *Patients.aspx*, από την οποία με την μέθοδο GET (π.χ. [http://localhost:51271/Visit\\_History.aspx?Patient\\_Id=10](http://localhost:51271/Visit_History.aspx?Patient_Id=10)) ανακτώνται πληροφορίες (από την βάση) σχετικά με το αρχείο ασθενούς που έχει επιλεγθεί.



**Integrated Hospital Application  
for Drug Administration** logout

PATIENTS   DRUG SEARCH   DRUG ORDER

The patient details are shown below. You can edit them by pressing the Edit Patient button.

**Name :** George Kotsanas  
**ID :** AA 358681  
**Birth date :** 20/11/1987  
**Gender :** Male  
**Phone number :** 6974096344

Press the Visit History panel in order to choose a visit history record from the dropdown list and display its data (drugs info and quantity prescribed).

Visit History

Press the New Visit panel in order to create a new visit history record.  
Next, enter the ICD-10 code for the patient's disease and press the Submit button.  
Then choose the drug search method (ICD-10, ATC substance, drug brand name).  
For the selected drug, drugstore quantity is shown and you can enter the desired quantity to submit or order.

New Visit

Εικόνα 7.10– Η σελίδα *Visit\_History.aspx* στην αρχική της μορφή

Στην αρχή της σελίδας εμφανίζονται τα στοιχεία του ασθενούς, τα οποία ο γιατρός μπορεί να επεξεργαστεί, πατώντας το κουμπί *Edit Patient*.

<b>Name :</b>	George Kotsanas
<b>ID :</b>	AA 358681
<b>Birth date :</b>	20/11/1987
<b>Gender :</b>	Male
<b>Phone number :</b>	6974096344

[Edit Patient](#)

Εικόνα 7.11– Εμφάνιση στοιχείων του ασθενούς

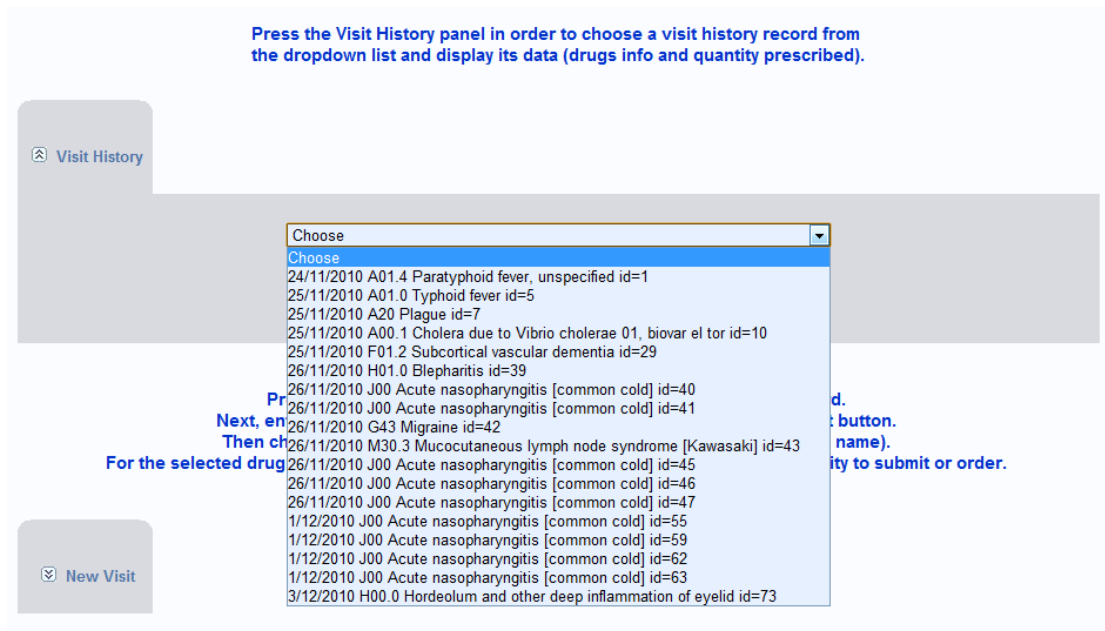
Αφού ο γιατρός αλλάξει τα στοιχεία, με το πάτημα του κουμπιού Update, ανανεώνεται η βάση δεδομένων.

<b>Name :</b>	<input type="text" value="George Kotsanas"/>
<b>ID :</b>	<input type="text" value="AA 358681"/>
<b>Birth date :</b>	<input type="text" value="20/11/1987"/>
<b>Gender :</b>	<input type="text" value="Male"/>
<b>Phone number :</b>	<input type="text" value="6974096344"/>

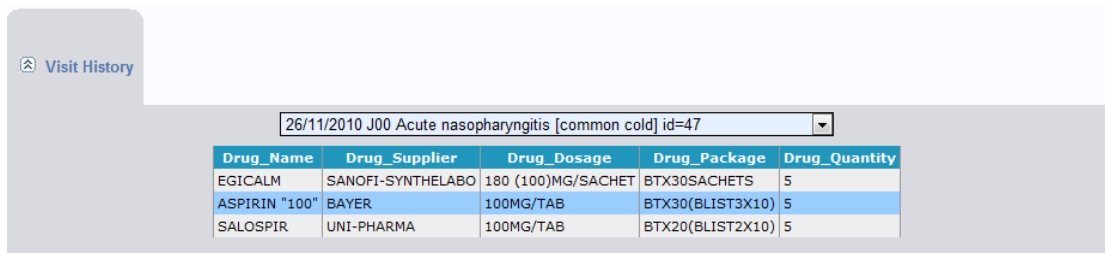
[Update](#)

Εικόνα 7.12– Τα στοιχεία του ασθενούς σε επεξεργάσιμη μορφή

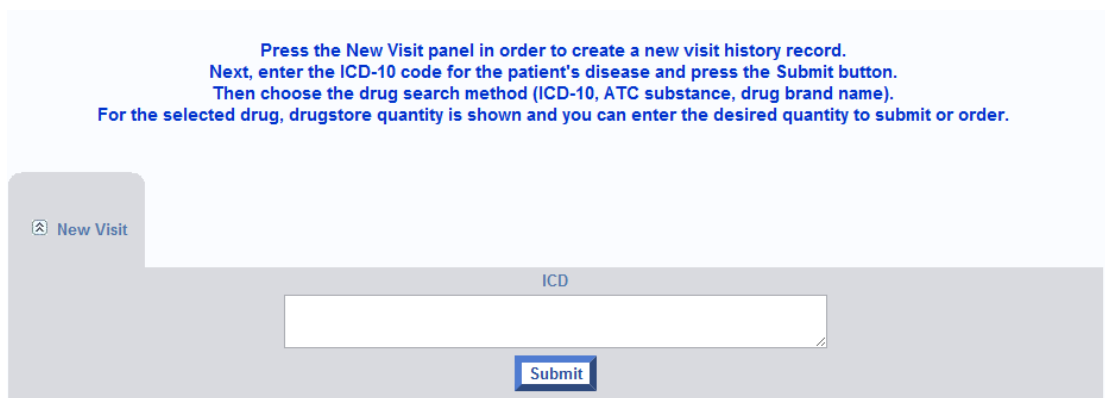
Πατώντας το Visit History panel εμφανίζεται (από το dropdown list) το ιστορικό του ασθενούς (ταξινομημένο κατά ημερομηνία επίσκεψης) και ο γιατρός έχει τη δυνατότητα να επιλέξει ένα από αυτά και να προβάλει τις πληροφορίες του (φάρμακα και ποσότητες που χορηγήθηκαν).



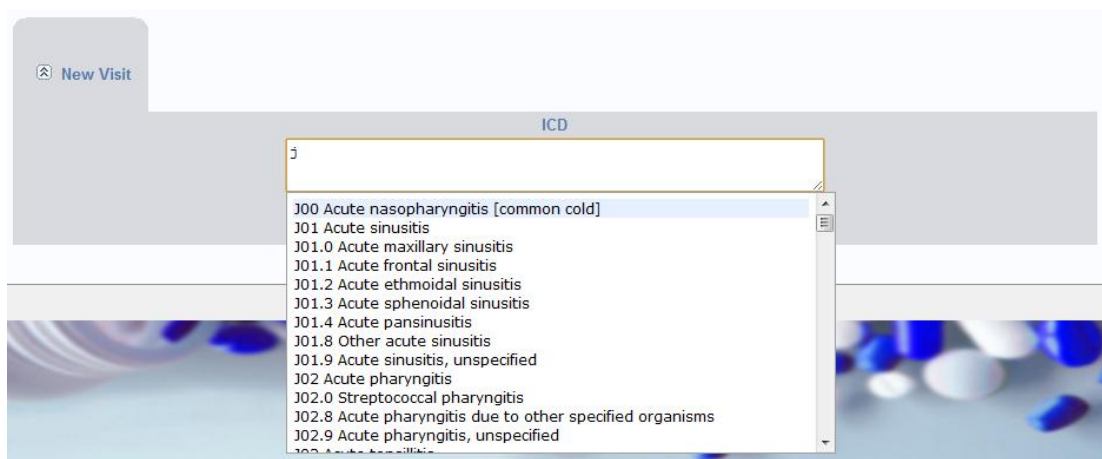
Εικόνα 7.13– Το ιστορικό του ασθενούς



Εικόνα 7.14– Τα φάρμακα και οι ποσότητες που χορηγήθηκαν στην συγκεκριμένη επίσκεψη Πατώντας το New Visit panel, ο γιατρός μπορεί να δημιουργήσει μια νέα εγγραφή ιστορικού (για νέα επίσκεψη ασθενούς). Αρχικά, με αυτόματη συμπλήρωση από τη βάση δεδομένων, δίνει τον κωδικό ICD-10 μαζί με το όνομα της ασθένειας του ασθενούς και πατώντας το κουμπί Submit εισάγονται στη βάση ο κωδικός του ασθενούς, ο ICD-10 κωδικός και η ημερομηνία επίσκεψης.



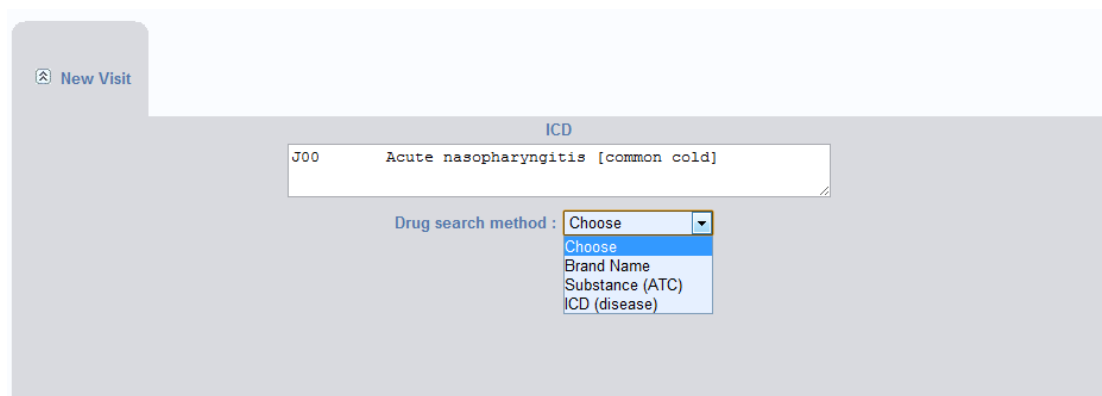
Εικόνα 7.15– Η φόρμα συμπλήρωσης του ICD-10 κωδικού



Εικόνα 7.16– Επιλογή ενός ICD-10 κωδικού με αυτόματη συμπλήρωση

Υπάρχουν τρεις εναλλακτικές μέθοδοι αναζήτησης φαρμάκων (από την αποθήκη) :

- Κατά Ονομασία σκευάσματος (Brand Name)
- Κύρια δραστική ουσία ATC
- ICD-10 κωδικοποίηση



Εικόνα 7.17– Επιλογή μεθόδου αναζήτησης φαρμάκων

**Σενάριο αναζήτησης κατά Ονομασία σκευάσματος (Brand Name) (βλ. Εικόνες 7.18-7.19).**

ICD

J00 Acute nasopharyngitis [common cold]

Drug search method : Brand Name

Search drug a

Next

- ASPIRIN "100" BAYER 100MG/TAB  
BTX30(BLIST3X10) id=471
- ALGOFREN UNI-PHARMA 400MG/TAB  
BTX24(BLIST 2X12) id=3334

Εικόνα 7.18– Επιλογή φαρμάκου με αυτόματη συμπλήρωση και πάτημα του κουμπιού Next

ICD

J00 Acute nasopharyngitis [common cold]

Drug search method : Brand Name

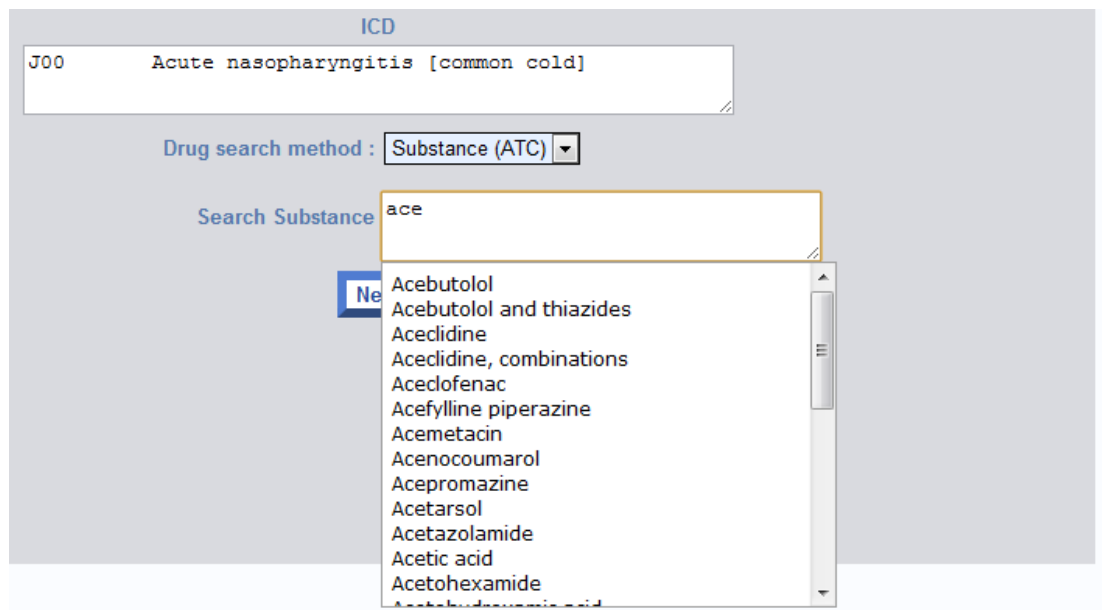
Search drug ASPIRIN "100" BAYER 100MG/TAB  
BTX30 (BLIST3X10) id=471

Enter quantity : 1 Drug store quantity: 1

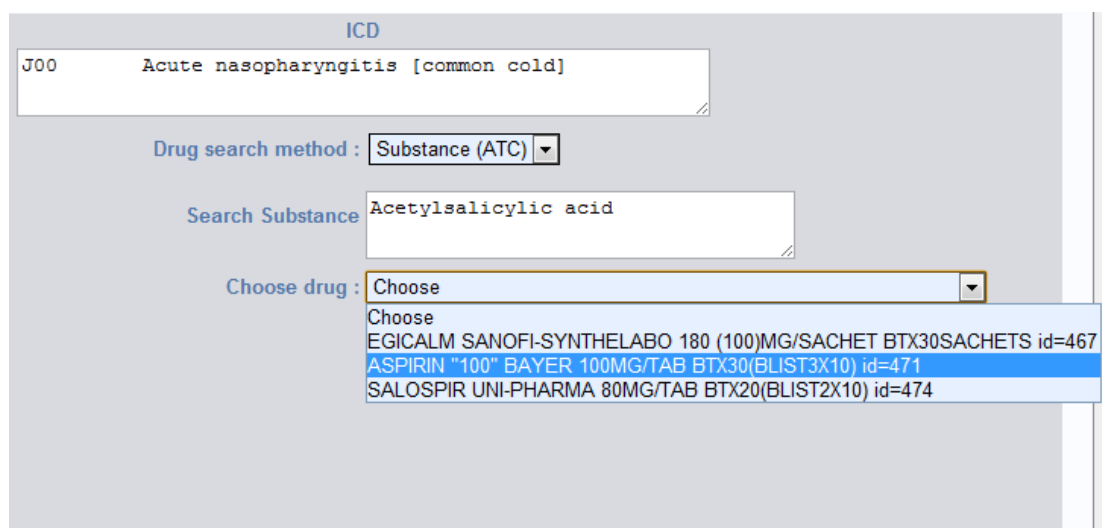
Submit Drug Order Drug

Εικόνα 7.19– Εμφάνιση ποσότητας φαρμάκου στην αποθήκη και εισαγωγή επιθυμητής ποσότητας για χορήγηση ή παραγγελία

**Σενάριο αναζήτησης κατά κύρια δραστική ουσία ATC (Ingredient (ATC)) (βλ. Εικόνες 7.20-7.22).**



Εικόνα 7.20– Επιλογή κύριας δραστικής ουσίας ATC με αυτόματη συμπλήρωση και πάτημα του κουμπιού Next



Εικόνα 7.21– Επιλογή φαρμάκου με την παραπάνω κύρια δραστική ουσία από το dropdown list

ICD

J00 Acute nasopharyngitis [common cold]

Drug search method : Substance (ATC) ▾

Search Substance Acetylsalicylic acid

Choose drug : ASPIRIN "100" BAYER 100MG/TAB BTX30(BLIST3X10) id=471 ▾

Enter quantity : 1 Drug store quantity: 2

Εικόνα 7.22– Εμφάνιση ποσότητας φαρμάκου στην αποθήκη και εισαγωγή επιθυμητής ποσότητας για χορήγηση ή παραγγελία

**Σενάριο αναζήτησης κατά ICD-10 (ICD) (βλ. Εικόνες 7.23-7.24).**

Ο γιατρός έχει επιλέξει έναν ICD-10 κωδικό για την ασθένεια του επισκέπτη. Κάθε κωδικός ICD-10 αντιστοιχεί σε κάποιους (έναν ή περισσότερους) κωδικούς ATC που έχουν κύρια δραστική ουσία που θεραπεύει την εν λόγω ασθένεια. Κάθε ATC κωδικός αντιστοιχεί σε κάποια (ένα ή περισσότερα) φάρμακα. Επομένως, έχοντας ως βάση κάποιον κωδικό ICD-10 καταλήγουμε σε ένα υποσύνολο φαρμάκων από την αποθήκη που θεραπεύουν τη συγκεκριμένη ασθένεια.

ICD

J00 Acute nasopharyngitis [common cold]

Drug search method : ICD (disease) ▾

Choose drug : Choose ▾

Choose  
EGICALM SANOFI-SYNTHELABO 180 (100)MG/SACHET BTX30SACHETS id=467  
ASPIRIN "100" BAYER 100MG/TAB BTX30(BLIST3X10) id=471  
SALOSPIR UNI-PHARMA 80MG/TAB BTX20(BLIST2X10) id=474  
ALGOFREN UNI-PHARMA 400MG/TAB BTX24(BLIST 2X12) id=3334

Εικόνα 7.23– Επιλογή ενός φαρμάκου από το υποσύνολο που θεραπεύουν την παραπάνω ασθένεια



ICD

J00 Acute nasopharyngitis [common cold]

Drug search method : ICD (disease) ▼

Choose drug : EGICALM SANOFI-SYNTHELABO 180 (100)MG/SACHET BTX30€ ▼

Enter quantity :  Drug store quantity: 13

Εικόνα 7.24– Εμφάνιση ποσότητας φαρμάκου στην αποθήκη και εισαγωγή επιθυμητής ποσότητας για χορήγηση ή παραγγελία

**Τελικό στάδιο για όλα τα σενάρια**

ICD

J00 Acute nasopharyngitis [common cold]

Drug search method : ICD (disease) ▼

Choose drug : EGICALM SANOFI-SYNTHELABO 180 (100)MG/SACHET BTX30€ ▼

Enter quantity :  Drug store quantity: 13

Order successful

Εικόνα 7.25– Πατώντας το κουμπί Order Drug καταχωρείται στη βάση η παραγγελία και εμφανίζεται το μήνυμα Order successful

ICD

J00 Acute nasopharyngitis [common cold]

Drug search method : ICD (disease) ▼

Choose drug : EGICALM SANOFI-SYNTHELABO 180 (100)MG/SACHET BTX30€ ▼

Enter quantity :  Drug store quantity: 13 Press Order Drug button or enter less quantity to submit

Εικόνα 7.26– Αν η ποσότητα που θέλει να χορηγήσει ο γιατρός είναι μεγαλύτερη από την διαθέσιμη ποσότητα (13 συσκευασίες) που υπάρχει στην αποθήκη το εμφανίζεται το μήνυμα σφάλματος Press Order Drug button or enter less quantity to submit

ICD

J00 Acute nasopharyngitis [common cold]

Drug search method : ICD (disease)

Choose drug : EGICALM SANOFI-SYNTHELABO 180 (100)MG/SACHET BTX30E

Enter quantity : 3 Drug store quantity: 10

Submit Drug Order Drug

Εικόνα 7.27– Με το πάτημα του κουμπιού Submit Drug η ποσότητα που χορηγεί ο γιατρός αφαιρείται αυτομάτως από την αποθήκη (πλέον είναι 10 συσκευασίες)

### 7.2.5 Σελίδα DatabaseContents.aspx

Ο γιατρός έχει την δυνατότητα να επιλέξει και να προβάλει όλους τους ATC κωδικούς, τους ICD-10 κωδικούς ή τα φάρμακα της βάσης δεδομένων. Επίσης επιστρέφεται το πλήθος των αποτελεσμάτων.

PATIENTS DRUG SEARCH DRUG ORDER

Choose below to display all ATC codes, ICD-10 codes or drugs.

Display all : Choose

Εικόνα 7.28– Η σελίδα DatabaseContents.aspx στην αρχική της μορφή

PATIENTS DRUG SEARCH DRUG ORDER

Choose below to display all ATC codes, ICD-10 codes or drugs.

Display all :

4123 Results

ATC Id	Ingredient
A01AA01	Sodium fluoride
A01AA02	Sodium monofluorophosphate
A01AA03	Oloflur
A01AA04	Stannous fluoride
A01AA30	Combinations
A01AA51	Sodium fluoride, combinations
A01AB02	Hydrogen peroxide
A01AB03	Chlorhexidine
A01AB04	Amphotericin B
A01AB05	Polynoxylin
A01AB06	Domiphen
A01AB07	Oxyquinoline
A01AB08	Neomycin
A01AB09	Miconazole
A01AB10	Natamycin
A01AB11	Various
A01AB12	Hexetidine
A01AB13	Tetracycline
A01AB14	Benzoxonium chloride
A01AB15	Tibezonium iodide
A01AB16	Mepartricin
A01AB17	Metronidazole
A01AB18	Clotrimazole
A01AB19	Sodium perborate
A01AB21	Chlortetracycline
A01AB22	Doxycycline
A01AB23	Mincycline
A01AC01	Triamcinolone
A01AC02	Dexamethasone
A01AC03	Hydrocortisone
A01AC54	Prednisolone, combinations
A01AD01	Epinephrine
A01AD02	Benzylamine
A01AD05	Acetylsalicylic acid
A01AD06	Adrenalone
A01AD07	Amlexanox
A01AD11	Various
A02AA01	Magnesium carbonate
A02AA02	Magnesium oxide
A02AA03	Magnesium peroxide
A02AA04	Magnesium hydroxide
A02AA05	Magnesium silicate

Εικόνα 7.29– Επιλογή ATC κωδικοποίησης

### 7.2.6 Σελίδα *Category.aspx*

Ο γιατρός έχει την δυνατότητα να επιλέξει μια κατηγορία φαρμάκων σύμφωνα με την κατηγοριοποίηση της DrugBank και να του επιστραφούν τα αποτελέσματα καθώς και το πλήθος τους.

PATIENTS DRUG SEARCH DRUG ORDER

Choose category below to display drugs (based on DrugBank categorisation).

Choose Drug Category :

Εικόνα 7.30– Η σελίδα *Category.aspx* στην αρχική της μορφή

PATIENTS    DRUG SEARCH    DRUG ORDER

Choose category below to display drugs (based on DrugBank categorisation).

Choose Drug Category :

**18 Results**

DRUG_Name	DRUG_Supplier	DRUG_Dosage	DRUG_Package	ROUTE_Name
CYTOTEC	PHARMACIA	200MCG/TAB	BTX42(BLIST 3X14)	tablet
EMTHEXATE	CHEMIPHARM	5MG/2ML VIAL	BTX1VIALX2ML	solution for injection
EMTHEXATE	CHEMIPHARM	50MG/2ML VIAL	BTX1VIALX2 ML	solution for injection
EMTHEXATE	CHEMIPHARM	5MG/2ML VIAL	BTX25VIALSX2ML	solution for injection
EMTHEXATE	CHEMIPHARM	500MG/20ML VIAL	BTX1VIALX20ML	solution for injection
EMTHEXATE	CHEMIPHARM	50MG/2ML VIAL	BTX25VIALS X2ML	solution for injection
EMTHEXATE	CHEMIPHARM	1000MG/40ML VIAL	BTX1VIALX40ML	solution for injection
EMTHEXATE	CHEMIPHARM	500MG/20ML VIAL	BTX10VIALSX20ML	solution for injection
EMTHEXATE	CHEMIPHARM	1000MG/40ML VIAL	BTX10VIALSX40ML	solution for injection
EMTHEXATE	CHEMIPHARM	2.5MG/TAB	BTX100(BLIST10X10)	tablet
METHOBION	MEDICUS	50MG/2ML VIAL	BTX1VIALX2ML	suspension for injection
METHOBION	MEDICUS	500MG/20ML VIAL	BTX1VIALX20ML	suspension for injection
METHOBION	MEDICUS	1000MG/40ML VIAL	BTX1VIALX40ML	suspension for injection
METHOTREXATE/DAVID BULL	GEROLYMATOS	500MG/20ML VIAL	BTX1VIALX20ML	solution for injection
METHOTREXATE/DAVID BULL	GEROLYMATOS	50MG/2ML VIAL	BTX5VIALSX2ML	solution for injection
METHOTREXATE/DAVID BULL	GEROLYMATOS	100MG/4ML VIAL	BTX5VIALSX4ML	solution for injection
METHOTREXATE/LEDERLE	WYETH	2.5MG/TAB	BTX100	tablet
METHOTREXATE/LEDERLE	WYETH	1000MG/40ML	1BOTTLX40ML	suspension for injection

Εικόνα 7.31– Επιλογή Abortifacient Agents

### 7.2.7 Σελίδα ICD.aspx

Ο γιατρός έχει τη δυνατότητα να αναζητήσει φάρμακα με την πληκτρολόγηση των πρώτων χαρακτήρων(αυτόματη συμπλήρωση) του ICD κωδικού τους.

PATIENTS    DRUG SEARCH    DRUG ORDER

Search drugs by ICD code :

Εικόνα 7.32– Η σελίδα ICD.aspx στην αρχική της μορφή

PATIENTS    DRUG SEARCH    DRUG ORDER

Search drugs by ICD code :

J00    Acute nasopharyngitis [common cold]

**Search**

28 Results

DRUG_Name	DRUG_Supplier	DRUG_Dosage	DRUG_Package	ROUTE_Name
EGICALM	SANOFI-SYNTHELABO	180 (100)MG/SACHET	BTX30SACHETS	single-dose powder for oral liquid
EGICALM	SANOFI-SYNTHELABO	288(160)MG/SACHET	BTX30 SACHETS	single-dose powder for oral liquid
EGICALM	SANOFI-SYNTHELABO	540(300)MG/SACHET	BTX30 SACHETS	single-dose powder for oral liquid
ASPIRIN	BAYER	100MG/TAB	BTX20(BLIST2X10)	tablet
ASPIRIN "100"	BAYER	100MG/TAB	BTX30(BLIST3X10)	gastro-resistant tablet
ASPIRIN "100"	BAYER	100MG/TAB	BTX90(BLIST9X10)	gastro-resistant tablet
ASPIRIN "300"	BAYER	300MG/TAB	BTX30(BLISTERS)	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	80MG/TAB	BTX20(BLIST2X10)	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	100MG/TAB	BT20(BLIST2X10)	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	160MG/TAB	BTX20(BLIST2X10)	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	325MG/TAB	BTX20	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	75MG/TAB	BTX20(BLIST2X10)	tablet
SALOSPIR	UNI-PHARMA	100MG/TAB	BTX20(BLIST2X10)	tablet
UPSALGIN-N	BRISTOL MYERS SQUIBB	0.325G/TAB	BTX10	effervescent tablet
ALGOFREN	UNI-PHARMA	100MG/SML	FL X 150ML	syrup
ALGOFREN	UNI-PHARMA	400MG/TAB	BTX24(BLIST 2X12)	film coated tablet
ALGOFREN	UNI-PHARMA	600MG/TAB	BT X24(BLIST 2X12)	film coated tablet
ALGOFREN	UNI-PHARMA	200MG/TAB	BTX30(BLIST 3X10)	film coated tablet
ALGOFREN	UNI-PHARMA	500MG/SUP	BT X12(STRIP 2X6)	suppository
BRUFEN	VIANEX	400MG/TAB	BTX24(BLIST3X8)	coated tablet
BRUFEN	VIANEX	600MG/TAB	BTX24(BLIST 3X8)	film coated tablet
BRUFEN	VIANEX	200MG/TAB	BTX30(BLIST3X10)	coated tablet
BRUFEN	VIANEX	500MG/SUP	BTX12	suppository
BRUFEN	VIANEX	100MG/SML	FLX150ML	syrup
SALOSPIR	UNI-PHARMA	1000MG/TAB	BTX30	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	650MG/TAB	BTX20	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	500MG/TAB	BTX20(BLIST2X10)	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	500MG/TAB	BTX50(BLIST5X10)	gastro-resistant tablet

Εικόνα 7.33– Εμφάνιση φαρμάκων με ICD-10 κωδικό J00

### 7.2.8 Σελίδα *Ingredient.aspx*

Ο γιατρός έχει τη δυνατότητα να αναζητήσει φάρμακα με την ηλεκτρολόγηση των πρώτων χαρακτήρων(αυτόματη συμπλήρωση) της ATC κύριας δραστικής ουσίας.

PATIENTS    DRUG SEARCH    DRUG ORDER

Search drugs by ATC substance :

**Search**

Εικόνα 7.34– Η σελίδα *Ingredient.aspx* στην αρχική της μορφή

PATIENTS    DRUG SEARCH    DRUG ORDER

Search drugs by ATC substance :

Acetylsalicylic acid

Search

18 Results

DRUG_Name	DRUG_Supplier	DRUG_Dosage	DRUG_Package	ROUTE_Name
EGICALM	SANOFI-SYNTHELABO	180 (100)MG/SACHET	BTX30SACHETS	single-dose powder for oral liquid
EGICALM	SANOFI-SYNTHELABO	288(160)MG/SACHET	BTX30 SACHETS	single-dose powder for oral liquid
EGICALM	SANOFI-SYNTHELABO	540(300)MG/SACHET	BTX30 SACHETS	single-dose powder for oral liquid
ASPIRIN	BAYER	100MG/TAB	BTX20(BLIST2X10)	tablet
ASPIRIN "100"	BAYER	100MG/TAB	BTX30(BLIST3X10)	gastro-resistant tablet
ASPIRIN "100"	BAYER	100MG/TAB	BTX90(BLIST9X10)	gastro-resistant tablet
ASPIRIN "300"	BAYER	300MG/TAB	BTX30(BLISTERS)	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	80MG/TAB	BTX20(BLIST2X10)	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	100MG/TAB	BT20(BLIST2X10)	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	160MG/TAB	BTX20(BLIST2X10)	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	325MG/TAB	BTX20	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	75MG/TAB	BTX20(BLIST2X10)	tablet
SALOSPIR	UNI-PHARMA	100MG/TAB	BTX20(BLIST2X10)	tablet
UPSALGIN-N	BRISTOL MYERS SQUIBB	0.325G/TAB	BTX10	effervescent tablet
SALOSPIR	UNI-PHARMA	1000MG/TAB	BTX30	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	650MG/TAB	BTX20	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	500MG/TAB	BTX20(BLIST2X10)	gastro-resistant tablet
SALOSPIR	UNI-PHARMA	500MG/TAB	BTX50(BLIST5X10)	gastro-resistant tablet

Εικόνα 7.35– Εμφάνιση φαρμάκων με ATC κύρια δραστική ουσία Acetylsalicylic acid

### 7.2.9 Σελίδα RouteName.aspx

Ο γιατρός έχει τη δυνατότητα να αναζητήσει φάρμακα με βάση τον τύπο της συσκευασίας (επιλογή από dropdownlist).

PATIENTS    DRUG SEARCH    DRUG ORDER

Search drugs by package type :

Choose

Εικόνα 7.36– Η σελίδα RouteName.aspx στην αρχική της μορφή

PATIENTS DRUG SEARCH DRUG ORDER

Search drugs by package type :

lotion

14 Results

DRUG_Name	DRUG_Supplier	DRUG_Dosage	DRUG_Package	ROUTE_Type
DAKTARIN	JANSSEN-CILAG	2% W/W	FLX30G	Subcutaneous use
TOLMICIL	PHARMACIA	1%	FLX30ML	Subcutaneous use
FLOGOZYME	NORMA	0.10% W/V	FLX50ML	Subcutaneous use
GALINOCORT	VILCO	0.1%	FLX60ML	Subcutaneous use
LOCASON SCALP	PROEL	0.1%	FLX60ML	Subcutaneous use
MOVITHIOL	FARMANIC	0.1% W/V	FLX50ML	Subcutaneous use
OSMORAN	RAFARM	0.1% W/V(BET.VAL)	FLX60ML	Subcutaneous use
SANORVIL	DEKAZ	0.1%	FLX70ML	Subcutaneous use
ELOCON	SCHERING PLOUGH	0.1% W/W	FLX30ML	Subcutaneous use
ELOCON	SCHERING PLOUGH	0.1% W/W	FLX20ML	Subcutaneous use
PROPIOSALIC	SCHERING PLOUGH	0.05%+2% W/W	FLX30ML	Subcutaneous use
CLIDACIN	VIANEX	1%	FLX30ML	Subcutaneous use
DALACIN C	PHARMACIA	1%	FLX30ML	Subcutaneous use
TOLIKEN	NORMA	1%	FLX25ML	Subcutaneous use

Εικόνα 7.37– Εμφάνιση φαρμάκων τύπου συσκευασίας lotion

### 7.2.10 Σελίδα RouteType.aspx

Ο γιατρός έχει τη δυνατότητα να αναζητήσει φάρμακα με βάση την οδό χορήγησης (επιλογή από dropdownlist).

PATIENTS DRUG SEARCH DRUG ORDER

Search drugs by administration route :

Choose

Εικόνα 7.38– Η σελίδα RouteType.aspx στην αρχική της μορφή

Search drugs by administration route :

Rectal

28 Results

DRUG_Name	DRUG_Supplier	DRUG_Dosage	DRUG_Package	ROUTE_Name
CLINDAGEL	VOCATE	2%W/W	TUBX40G+7DOS.EFARMOSTES	vaginal cream
DALACIN C	PHARMACIA	2%	TUBX40G+7DOS.EFARMOSTES	vaginal cream
FOUCH	RAFARM	2%	TUBX40G+7DOS.EFARMOSTES	vaginal cream
YGIELLE	LAMDA	2%	TUBX40G+7DOS.EFARMOSTES	vaginal cream
UPDERM	GENEPHARM	2%	TUBX40G+7DOS.EFARMOSTES	vaginal cream
VELDOM	VIOFAR	2% W/W	TUBX40G+7DOS.EFARMOSTES	vaginal cream
VELKADERM	VELKA	2%	TUBX40G+7DOS.EFARMOSTES	vaginal cream
SULTRIN	JANSSEN-CILAG		TUBX78G+1 APPLICATOR	vaginal cream
FLAGYL	AVENTIS	500MG/SUP	BTX10	pessary
CANESTEN	BAYER	0.5G/TAB	BTX1TAB+APPLICATOR	vaginal tablet
CANESTEN	BAYER	2%	TUBX20G+3APPL.	vaginal cream
CANESTEN	BAYER	0.1G/TAB	BTX6(AL FORM PACK)+APPLIQUEUR	vaginal tablet
DAKTARIN	JANSSEN-CILAG	200MG/SUP	BTX7	pessary
DAKTARIN	JANSSEN-CILAG	400MG/SUP	BTX3	pessary
DAKTARIN	JANSSEN-CILAG	2%	TUBX78G+16PROOTH.	vaginal cream
BISMULTIN	RAFARM	150MG/SUP	BTX3	pessary
MYCOBACTER	BIO SPRAY	1%	TUBX78G+1APPLIQUEUR	vaginal cream
PEVARYL	JANSSEN-CILAG	1% W/W	TUBX78G+16 APPLICAT.	vaginal cream
PEVARYL	JANSSEN-CILAG	150MG/SUP	BTX3	pessary
LOMEXIN	GALENICA	600MG/SUP	BTX1	pessary
LOMEXIN	GALENICA	200MG/SUP	BTX3	pessary
LOMEXIN	GALENICA	2%	TUBX35G+16SYSK.	vaginal cream
PROPESS	CHEMI PHARM	10MG/DEVICE	BTX5VAG.DEVICE	vaginal device
PROSTIN E2	PHARMACIA	3MG/TAB	BTX4	vaginal tablet
VAGIFEM	NOVO NORDISK	25MCG/TAB	BTX15(BLIST3X5)	vaginal tablet
PREMARIN	WYETH	0.0625% W/W	TUBX42.5G	vaginal ointment
CRINONE	SERONO	8% (W/W)	BTX15APPL.X1,125G(SYSKEYASIA MONO GIA EXOSOMATIKI)	vaginal gel
CRINONE	SERONO	8% (W/W)	BTX6APPL.X1,125G	vaginal gel

Εικόνα 7.39– Εμφάνιση φαρμάκων με οδό χορήγησης Rectal

### 7.2.11 Σελίδα *TreeView.aspx*

Αυτό είναι το δέντρο της ATC κωδικοποίησης 5-επιπέδων. Ο γιατρός επιλέγει ένα κόμβο του δένδρου και πατώντας το κουμπί Search Drug εμφανίζονται όλα τα φάρμακα με τον ATC κωδικό(ενός από τα 5-επίπεδα) του επιλεγμένου κόμβου καθώς και το πλήθος τους.

PATIENTS    DRUG SEARCH    DRUG ORDER

**The 5-level ATC treeview is shown below.**  
Select one tree node and press the Search Drug button to display all drugs with the selected tree node ATC code.

- A ALIMENTARY TRACT AND METABOLISM
- B BLOOD AND BLOOD FORMING ORGANS
- C CARDIOVASCULAR SYSTEM
- D DERMATOLOGICALS
- G GENITO URINARY SYSTEM AND SEX HORMONES
- H SYSTEMIC HORMONAL PREPARATIONS, EXCL. SEX HORMONES AND INSULINS
- J ANTIINFECTIVES FOR SYSTEMIC USE
- L ANTINEOPLASTIC AND IMMUNOMODULATING AGENTS
- M MUSCULO-SKELETAL SYSTEM
- N NERVOUS SYSTEM
- P ANTIPARASITIC PRODUCTS, INSECTICIDES AND REPELLENTS
- R RESPIRATORY SYSTEM
- S SENSORY ORGANS
- V VARIOUS

Εικόνα 7.40– Η σελίδα *TreeView.aspx* στην αρχική της μορφή



PATIENTS DRUG SEARCH DRUG ORDER

The 5-level ATC treeview is shown below.  
Select one tree node and press the Search Drug button to display all drugs with the selected tree node ATC code.

- ▣ A ALIMENTARY TRACT AND METABOLISM
- ▣ B BLOOD AND BLOOD FORMING ORGANS
- ▣ C CARDIOVASCULAR SYSTEM
- ▣ D DERMATOLOGICALS
- ▣ G GENITO URINARY SYSTEM AND SEX HORMONES
- ▣ H SYSTEMIC HORMONAL PREPARATIONS, EXCL. SEX HORMONES AND INSULINS
- ▣ J ANTIINFECTIVES FOR SYSTEMIC USE
- ▣ L ANTINEOPLASTIC AND IMMUNOMODULATING AGENTS
- ▣ M MUSCULO-SKELETAL SYSTEM
- ▣ N NERVOUS SYSTEM
- ▣ P ANTIPARASITIC PRODUCTS, INSECTICIDES AND REPELLENTS
- ▣ R RESPIRATORY SYSTEM
- ▣ S SENSORY ORGANS
- ▣ V VARIOUS
  - ▣ V01 ALLERGENS
  - ▣ V03 ALL OTHER THERAPEUTIC PRODUCTS
    - ▣ V03A ALL OTHER THERAPEUTIC PRODUCTS
      - ▣ V03AB Antidotes
        - ▣ V03AC Iron chelating agents
          - ▣ V03AC01 Deferoxamine
          - ▣ V03AC02 Deferiprone
        - ▣ V03AE Drugs for treatment of hyperkalemia and hyperphosphatemia
        - ▣ V03AF Detoxifying agents for antineoplastic treatment
        - ▣ V03AG Drugs for treatment of hypercalcemia
        - ▣ V03AH Drugs for treatment of hypoglycemia
        - ▣ V03AK Tissue adhesives
        - ▣ V03AM Drugs for embolisation
        - ▣ V03AN Medical gases
        - ▣ V03AX Other therapeutic products
        - ▣ V03AZ Nerve depressants
      - ▣ V04 DIAGNOSTIC AGENTS
      - ▣ V06 GENERAL NUTRIENTS
      - ▣ V07 ALL OTHER NON-THERAPEUTIC PRODUCTS
      - ▣ V08 CONTRAST MEDIA
      - ▣ V09 DIAGNOSTIC RADIOPHARMACEUTICALS
      - ▣ V10 THERAPEUTIC RADIOPHARMACEUTICALS
      - ▣ V20 SURGICAL DRESSINGS

**Search Drug**

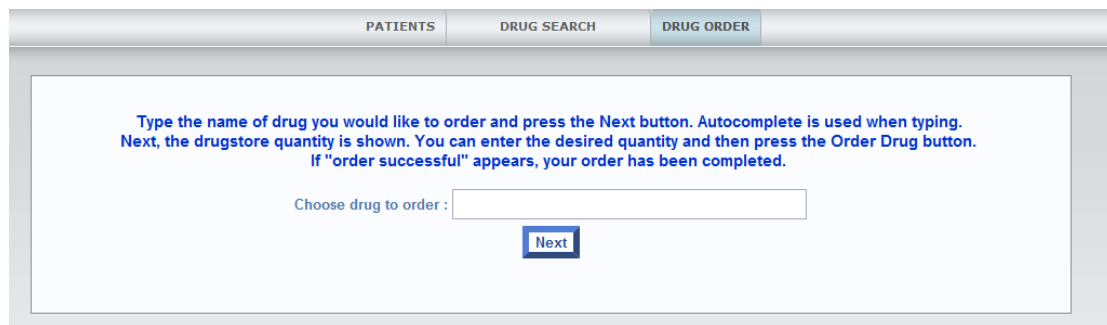
3 Results

DRUG_Name	DRUG_Supplier	DRUG_Dosage	DRUG_Package	ROUTE_Name
DESFERAL	NOVARTIS	500MG/VIAL	BTX10VIAL	lyophilized powder for injection
FERRIPROX	APOTEX	500MG/TAB	FLX100	film coated tablet
KELFER	VIANEX	500MG/CAP	BTX100	capsule, hard

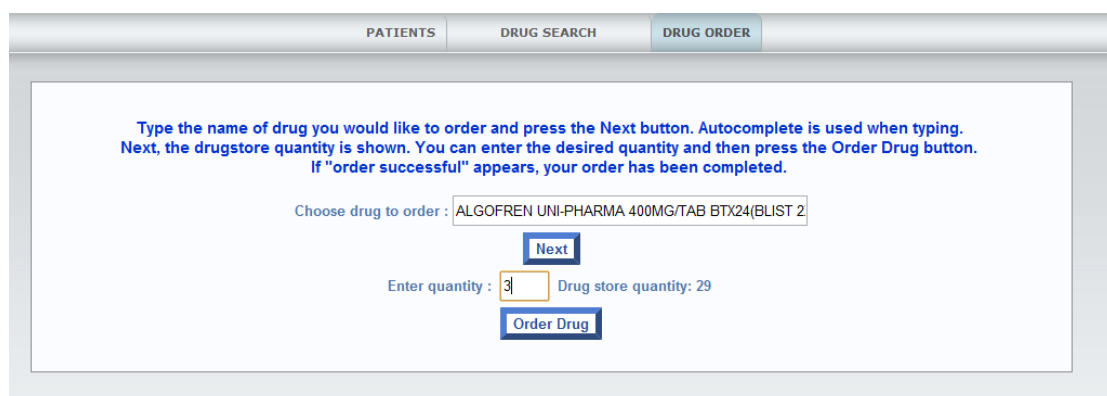
Εικόνα 7.41– Εμφάνιση φαρμάκων με ATC κωδικό (4ου-επιπέδου) V03AC

### 7.2.12 Σελίδα *Orders.aspx*

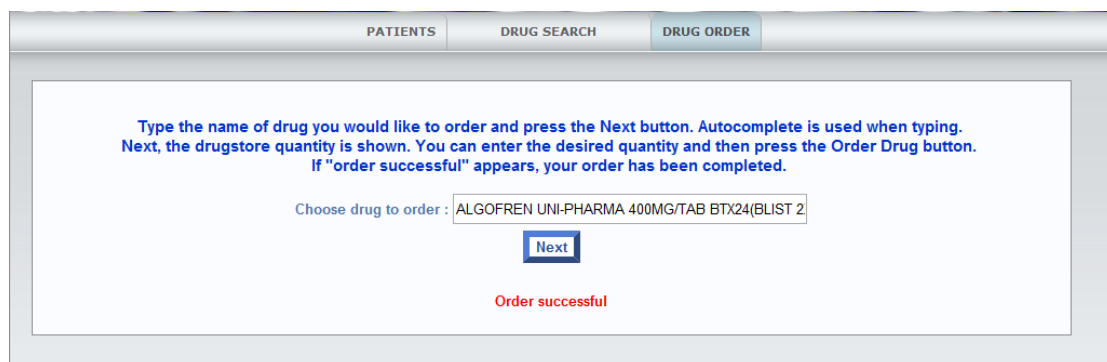
Ο γιατρός έχει τη δυνατότητα να αναζητήσει κάποιο φάρμακο με την πληκτρολόγηση των πρώτων χαρακτήρων (αυτόματη συμπλήρωση), το οποίο βρίσκεται στην αποθήκη του νοσοκομείου. Ύστερα πιέζοντας το κουμπί Next εμφανίζεται η υπάρχουσα ποσότητα του φαρμάκου στην αποθήκη του νοσοκομείου. Ο γιατρός πληκτρολογεί την επιθυμητή ποσότητα για παραγγελία και πατώντας το κουμπί Order Drug η παραγγελία του καταχωρείται προς εκπλήρωση της από τον αποθηκάριο. Αν η παραγγελία είναι επιτυχής εμφανίζεται το μήνυμα Order successful.



Εικόνα 7.42– Η σελίδα Orders.aspx στην αρχική της μορφή



Εικόνα 7.43– Παραγγελία 3 συσκευασιών ALGOFREN και εμφάνιση διαθέσιμης ποσότητας (29 συσκευασίες)

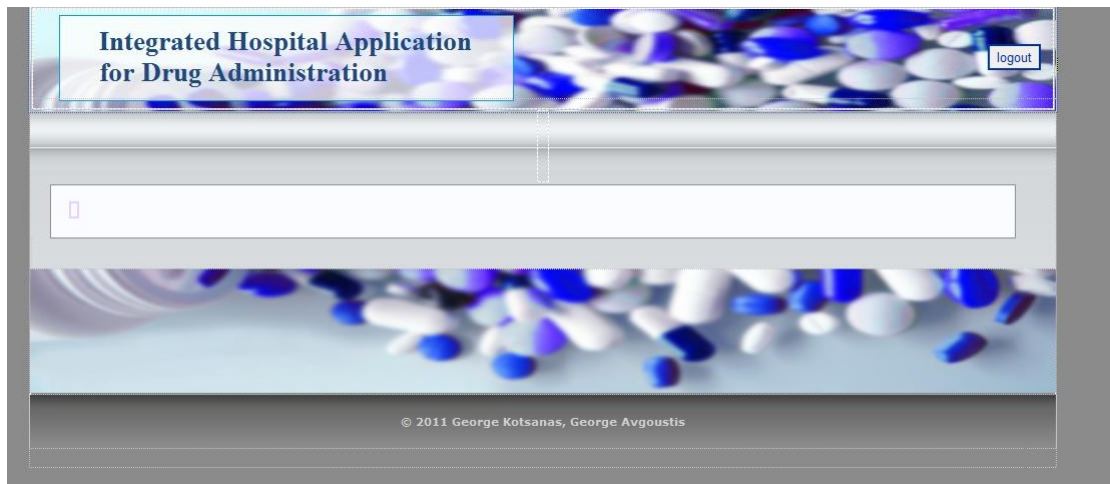


Εικόνα 7.44– Εμφάνιση μηνύματος επιβεβαίωσης παραγγελίας

## 7.3 Σενάριο λογαριασμού αποθηκάρου

### 7.3.1 Σελίδα Warehouse.Master

Στο σενάριο χρήσης του αποθηκάρου χρησιμοποιήσαμε την παρακάτω εικονιζόμενη masterpage.



Εικόνα 7.3– Το masterpage για το σενάριο αποθηκάριου

Επιπλέον, στα masterpage βρίσκεται υπάρχει το κουμπί logout με το οποίο ο χρήστης μπορεί να αποσυνδεθεί και αυτομάτως να προωθηθεί στην αρχική σελίδα της εφαρμογής Login.aspx.

### 7.3.2 Σελίδα Warehouse.aspx

Ο αποθηκάριος έχει πρόσβαση σε όλες τις παραγγελίες που έχουν γίνει και σκοπός του είναι να τις ικανοποιήσει. Επιλέγει λοιπόν μια ή περισσότερες και πατώντας το κουμπί Commit. Επιπλέον, προς διευκόλυνση του μπορεί να πατήσει το κουμπί Check All για να επιλέξει όλες τις παραγγελίες. Η ικανοποίηση μια παραγγελίας προϋποθέτει την αφαίρεση της ποσότητας από το container και την προσθήκη της στην αποθήκη. Η ποσότητα της παραγγελίας φαίνεται στην στήλη Quantity. Οι παραγγελίες μπορούν να προέλθουν από τις σελίδες Visit\_History.aspx και Orders.aspx.

In the table below all unsatisfied orders are shown.  
Select one or more orders and press the commit button  
In order to select all orders press the Check All button.

Check	DRUG_Name	DRUG_Supplier	DRUG_Dosage	DRUG_Package	Quantity	Drug_Id
<input type="checkbox"/>	EGICALM	SANOFI-SYNTHELABO	180 (100)MG/SACHET	BTX30SACHETS	20	467
<input type="checkbox"/>	ASPIRIN "100"	BAYER	100MG/TAB	BTX30(BLIST3X10)	25	471
<input type="checkbox"/>	SALOSPIR	UNI-PHARMA	80MG/TAB	BTX20(BLIST2X10)	10	474
<input type="checkbox"/>	ALGOFREN	UNI-PHARMA	400MG/TAB	BTX24(BLIST 2X12)	3	3334

Εικόνα – Η σελίδα Warehouse.aspx στην αρχική της μορφή

In the table below all unsatisfied orders are shown.  
Select one or more orders and press the commit button  
In order to select all orders press the Check All button.

Check	DRUG_Name	DRUG_Supplier	DRUG_Dosage	DRUG_Package	Quantity	Drug_Id
<input type="checkbox"/>	EGICALM	SANOFI-SYNTHELABO	180 (100)MG/SACHET	BTX30SACHETS	20	467
<input type="checkbox"/>	ASPIRIN "100"	BAYER	100MG/TAB	BTX30(BLIST3X10)	25	471
<input checked="" type="checkbox"/>	SALOSPIR	UNI-PHARMA	80MG/TAB	BTX20(BLIST2X10)	10	474
<input checked="" type="checkbox"/>	ALGOFREN	UNI-PHARMA	400MG/TAB	BTX24(BLIST 2X12)	3	3334

Εικόνα – Επιλογή δύο παραγγελιών προς ικανοποίηση

In the table below all unsatisfied orders are shown.  
Select one or more orders and press the commit button  
In order to select all orders press the Check All button.

Check	DRUG_Name	DRUG_Supplier	DRUG_Dosage	DRUG_Package	Quantity	Drug_Id
<input type="checkbox"/>	EGICALM	SANOFI-SYNTHELABO	180 (100)MG/SACHET	BTX30SACHETS	20	467
<input type="checkbox"/>	ASPIRIN "100"	BAYER	100MG/TAB	BTX30(BLIST3X10)	25	471

Εικόνα – Πάτημα του κουμπιού Commit και ικανοποίηση των παραγγελιών

# 8

## *Επίλογος*

Το παρόν κεφάλαιο αποτελεί τον επίλογο του τόμου της διπλωματικής εργασίας. Παρατίθενται τα συμπεράσματα από την εκπόνηση και χρήση του λογισμικού συστήματος καθώς επίσης και οι πιθανές μελλοντικές επεκτάσεις που θα μπορούσαν να εφαρμοσθούν σε αυτό.

### *8.1 Σύνοψη και συμπεράσματα*

Η ολοκληρωμένη εφαρμογή διαχείρισης ηλεκτρονικού φαρμακείου όπως παρουσιάστηκε στο αντίστοιχο κεφάλαιο μπορεί κάλλιστα να βρει εφαρμογή σε νοσοκομειακές εγκαταστάσεις. Κατά αυτόν τον τρόπο επιτυγχάνεται αυστηρή προτυποποίηση δημιουργίας και διαχείρισης ηλεκτρονικών φακέλων και διατήρησης ιστορικού ασθενών. Επίσης, κάθε γιατρός απολαμβάνει ελεύθερη πρόσβαση και στα ιστορικά των ασθενών των άλλων γιατρών του νοσοκομείου και συνεπώς αυτή η ενιαία βάση δεδομένων αποκτά πλέον και εκπαιδευτικό χαρακτήρα για το ιατρικό προσωπικό του νοσοκομείου.

Επιπρόσθετα, εγγυάται την συμφωνία των φαρμακευτικών σκευασμάτων με τα πρότυπα του Ε.Ο.Φ. και την συσχέτιση τους με τις διεθνείς τυποποιήσεις του Ευρωπαϊκού τμήματος της Παγκόσμιας Οργάνωσης Υγείας (WHO) για τις κωδικοποιήσεις ασθενειών ICD-10 και φαρμάκων ATC.

Τέλος, η παραγγελία καθορισμένων ποσοτήτων φαρμακευτικών σκευασμάτων καθίσταται πλέον εφικτή με ημιαυτόματο τρόπο. Η δημιουργία λογαριασμού αποθηκάρου τείνει προς αυτή την κατεύθυνση.

## **8.2 Μελλοντικές Επεκτάσεις**

- Το παρόν σύστημα, όπως είναι υλοποιημένο, παρέχει μερική συσχέτιση (ορισμένη χειροκίνητα) μεταξύ ICD-10 και ATC κωδικοποιήσεων καθώς αυτό δεν είναι ακόμα εφικτό και βρίσκεται σε ερευνητικό στάδιο. Επομένως μελλοντικός στόχος είναι η επέκταση του παρόντος συστήματος ώστε να υποστηρίζει αυτόματα και πλήρως την συσχέτιση των παραπάνω κωδικοποιήσεων.
- Δημιουργία σελίδας διαχειριστή όπου θα μπορούν να προστίθενται νέα φάρμακα.
- Δημιουργία διακριτών λογαριασμών γιατρών και διατήρηση ιστορικών τους.
- Ανάθεση διακριτών ρόλων στους γιατρούς(κατηγοριοποίηση τους με βάση ειδικότητα). Το σύστημα έτσι αποκτά δομή και αυξημένη λειτουργικότητα.
- Αναγραφή τιμής κάθε φαρμακευτικού σκεύασματος ώστε να διαφαίνεται ο προϋπολογισμός.
- Δυνατότητα καταγραφής και εμφάνισης τυχόν αλλεργιών που υπάρχουν σε διάφορα φαρμακευτικά σκεύασματα, για αποτροπή σχετικών αλλεργικών αντιδράσεων στον ασθενή.
- Καταχώριση ημερομηνίας λήξης κάθε παρτίδας παραλαβής φαρμάκων και συνεπώς εντοπισμός και αφαίρεσης των ληγμένων φαρμάκων.

# 9

## *Βιβλιογραφία*

- [1] A new drug classification for computer systems: the ATC extension code  
G. C. Miller and H. Britt
- [2] Comparability of Cause of Death Between ICD–9 and ICD–10: Preliminary Estimates  
by Robert N. Anderson, Ph.D.; Arialdi M. Miniño, M.P.H.; Donna L. Hoyert, Ph.D.; and  
Harry M. Rosenberg, Ph.D.
- [3] Coding Algorithms for Defining Comorbidities in ICD-9-CM and ICD-10 Administrative  
Data Quan, Hude MD, PhD; Sundararajan, Vijaya MD, MPH, FACP; Halfon, Patricia MD;  
Fong, Andrew BCOMM; Burnand, Bernard MD, MPH; Luthi, Jean-Christophe MD, PhD;  
Saunders, L Duncan MBBCh, PhD; Beck, Cynthia A. MD, MASc; Feasby, Thomas E. MD;  
Ghali, William A. MD, MPH
- [4] Guidelines for ATC classification and DDD assignment 14th edition WHO Collaborating  
Centre for Drug Statistics Methodology Norwegian Institute of Public Health
- [5] <http://www.whocc.no/>
- [6] <http://www.who.int/classifications/icd/en/>
- [7] <http://www.yyka.gov.gr/prokurixeis/ICD1020082guide.pdf>
- [8] <http://en.wikipedia.org/wiki/ICD-10>
- [9] <http://www.eof.gr/web/guest>
- [10] <http://en.wikipedia.org/wiki/DrugBank>

- [11] [http://en.wikipedia.org/wiki/Language\\_Integrated\\_Query](http://en.wikipedia.org/wiki/Language_Integrated_Query)
- [12] <http://wiki.asp.net/page.aspx/106/master-pages/>
- [13] [http://el.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://el.wikipedia.org/wiki/Microsoft_SQL_Server)
- [14] Beginning-ASP.NET-3.5-in-C-Sharp-From-Novice-to-Professional
- [15] Professional C 2008 Wrox – Mar 2008
- [16] Microsoft Net Framework, <http://www.microsoft.com/NET/>
- [17] [http://www.wlearn.gr/index.php?option=com\\_content&view=article&id=78:home&catid=25:-css&Itemid=83](http://www.wlearn.gr/index.php?option=com_content&view=article&id=78:home&catid=25:-css&Itemid=83)
- [18] <http://www.ifet.gr/>
- [19] [http://en.wikipedia.org/wiki/Ajax\\_%28programming%29](http://en.wikipedia.org/wiki/Ajax_%28programming%29)



# 10

Π

## αράρτημα

### Login.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Login.aspx.cs"
Inherits="WebApplication1.Login" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
  <link rel="stylesheet" href="andreas09.css" type="text/css" media="screen" />
  <link href="Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
</head>
<body style="width: 790px; margin: 0 auto;">
  <form id="form1" runat="server">
    <asp:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
    </asp:ToolkitScriptManager>
    <div align="center">
      <asp:Panel ID="Panel1" runat="server" Height="546px" BackImageUrl="~/img/login6.jpg">
        <table>
          <tr>
            <td>
              <table class="login">
                <tr>
                  <td align="center" colspan="2" style="color: White; background-color: #198BC6; font-size:
0.9em;
font-weight: bold;">
                    Log In
                </td>
                <tr>
                  <td align="right" bgcolor="#EFF3FB" class="style3">
                    <asp:Label ID="Label1" runat="server" BackColor="#EFF3FB" Text="Username
:"></asp:Label>
                </td>
                <td bgcolor="#EFF3FB" class="style4">
                    <asp:TextBox ID="TextBox1" runat="server" autocomplete="off"></asp:TextBox>
                </td>
                </tr>
              </table>
            </td>
          </tr>
        </table>
      </div>
    </asp:Panel>
  </form>
</body>
</html>
```

```

        <td align="right" bgcolor="#EFF3FB" class="style5">
            <asp:Label ID="Label2" runat="server" Text="Password :"></asp:Label>
        </td>
        <td bgcolor="#EFF3FB" class="style6">
            <asp:TextBox ID="TextBox2" runat="server" TextMode="Password"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td align="center" bgcolor="#EFF3FB" class="style1" colspan="2">
            <asp:Button ID="Button1" runat="server" BackColor="White" BorderColor="#507CD1"
BorderStyle="Solid"
                BorderWidth="1px" Font-Names="Verdana" Font-Size="0.8em" ForeColor="#284E98"
                OnClick="Button1_Click" Text="Log In" />
        </td>
    </tr>
    <tr>
        <td align="center" bgcolor="#EFF3FB" class="style1" colspan="2">
            <asp:Label ID="Label3" runat="server" CssClass="resullabel"></asp:Label>
        </td>
    </tr>
</table>
</td>
</tr>
</table>
</asp:Panel>
</div>
</form>
</body>
</html>

```

#### Login.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1
{
    public partial class Login : System.Web.UI.Page
    {
        Class1 ab = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Session["Login"] = ab.CheckLogin(TextBox1.Text, TextBox2.Text);
            if (Session["Login"].ToString().Equals("doctor"))
                Response.Redirect("Patients.aspx");
            else if (Session["Login"].ToString().Equals("warehouseman"))
                Response.Redirect("Warehouse.aspx");
            else if (Session["Login"].ToString().Equals("invalid"))
                Label3.Text = "Login failed";
        }
    }
}

```

#### Patients.aspx

```

<%@ Page Language="C#" MasterPageFile="~/DoctorMasterPage.Master" AutoEventWireup="true"
CodeBehind="Patients.aspx.cs" Inherits="WebApplication1.Patients" %>

```

```

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
  <link href="Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
  <title>Patient</title>
  <style type="text/css">
    .style129
    {
      width: 922px;
      height: 81px;
    }
    .style132
    {
      width: 154px;
      height: 22px;
    }
    .style134
    {
      height: 21px;
      width: 287px;
    }
    .style135
    {
      width: 154px;
      height: 21px;
    }
    .style136
    {
      width: 153px;
      height: 22px;
    }
    .style137
    {
      height: 22px;
      width: 287px;
    }
    .style138
    {
      height: 34px;
    }
    .style139
    {
      width: 153px;
      height: 21px;
    }
    .style140
    {
      width: 184px;
      height: 21px;
    }
    .style141
    {
      width: 184px;
      height: 22px;
    }
  </style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
  <span id="tabid" style="display: none">PatientsTab</span>
  <div align="left">
    <asp:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
    </asp:ToolkitScriptManager>
    <br />
    <br />
    <asp:UpdatePanel ID="UpdatePanel2" runat="server">
      <ContentTemplate>
        <table style="width:100%;">

```



```

        ID Number
    </td>
    <td align="center" class="style140">
        Name
    </td>
    <td align="center" class="style140">
        Gender
    </td>
    <td align="center" class="style140">
        Birth date
    </td>
    <td align="center" class="style140">
        Phone Number
    </td>
</tr>
<tr>
    <td align="center" class="style141">
        <asp:TextBox ID="TextBox1" runat="server" autocomplete="off" ></asp:TextBox>
    </td>
    <td align="center" class="style141">
        <asp:TextBox ID="TextBox2" runat="server" autocomplete="off"></asp:TextBox>
    </td>
    <td align="center" class="style141">
        <asp:DropDownList ID="DropDownList2" runat="server" Height="22px" Style="margin-left:
0px"
            Width="109px">
            <asp:ListItem Selected="True">Male</asp:ListItem>
            <asp:ListItem>Female</asp:ListItem>
        </asp:DropDownList>
    </td>
    <td align="center" class="style141">
        <asp:TextBox ID="TextBox4" runat="server" Style="margin-right: 0px"
            ></asp:TextBox>
        <asp:CalendarExtender ID="TextBox4_CalendarExtender" runat="server" Enabled="True"
            TargetControlID="TextBox4" PopupButtonID="Image1" Format="dd/MM/yyyy">
        </asp:CalendarExtender>
        <asp:Image ID="Image1" runat="server" AlternateText="Click to show calendar"
            ImageUrl="images/Calendar_scheduleHS.png" />
    </td>
    <td align="center" class="style141">
        <asp:TextBox ID="TextBox5" runat="server" autocomplete="off" ></asp:TextBox>
    </td>
</tr>
<tr>
    <td align="center" colspan="5" class="style138">
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Submit"
        CssClass="button" />
    </td>
</tr>
</table>
<br />
</asp:Panel>
<br />
<asp:CollapsiblePanelExtender ID="cpeDescription" runat="Server"
TargetControlID="description_ContentPanel"
ExpandControlID="description_HeaderPanel" CollapseControlID="description_HeaderPanel"
Collapsed="True" ImageControlID="description_ToggleImage"
ExpandedImage="images/collapse.jpg"
CollapsedImage="images/expand.jpg" />
</div>
</ContentTemplate>
</asp:UpdatePanel>
</div>
</asp:Content>

```

**Patients.aspx.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1
{
    public partial class Patients : System.Web.UI.Page
    {
        Class1 ab = new Class1();

        protected void Page_Load(object sender, EventArgs e)
        {
            if(!Page.IsPostBack)
            {
                if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))
                    Response.Redirect("Login.aspx");
                DropDownList1.DataSource = ab.GetAllPatientsName();
                DropDownList1.DataBind();
                DropDownList1.Items.Insert(0, new ListItem("Choose", "Choose"));
            }
        }

        protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
        {
            Response.Redirect("Visit_History.aspx?Patient_Id=" +
ab.GetPatient_id_by_Name_And_Birth_Date(DropDownList1.SelectedItem.Text));
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("Visit_History.aspx?Patient_Id=" + ab.SetPatientAndGetPatient_Id(TextBox1.Text,
TextBox2.Text, DropDownList2.SelectedItem.Text, TextBox4.Text, TextBox5.Text));
        }
    }
}

```

### Visit\_History.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="~/DoctorMasterPage.Master"
CodeBehind="Visit_History.aspx.cs" Inherits="WebApplication1.Visit_History" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
    <link href="Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
    <link href="Styles/StyleSheet1.css" rel="stylesheet" type="text/css" />
    <title>Visit History and Drug Administration</title>
    <style type="text/css">
        .style1
        {
            height: 17px;
        }
        .style2
        {
            height: 20px;
        }
    </style>

```

```

.style3
{
    background-color: #D9DADF;
    overflow: hidden;
}
.style125
{
    height: 20px;
    width: 107px;
}
.style129
{
    width: 107px;
}
</style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
<span id="tabid" style="display: none">PatientsTab</span>
<asp:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
</asp:ToolkitScriptManager>
<asp:UpdatePanel ID="UpdatePanel12" runat="server">
<ContentTemplate>
<table style="width: 100%;">
<tr>
<td align="center" colspan="2">
<asp:Label ID="Label20" runat="server"
Text="The patient details are shown below. You can edit them by pressing th Edit Patient
button."
CssClass="label2"></asp:Label>
<br />
<br />
<br />
</td>
</tr>
<tr>
<td class="style129">
<asp:Label ID="Label15" runat="server" CssClass="patientlabel" Text="Name : "></asp:Label>
</td>
<td>
<asp:TextBox ID="TextBox5" runat="server" CssClass="textbox"
ReadOnly="True"></asp:TextBox>
</td>
</tr>
<tr>
<td class="style125">
<asp:Label ID="Label16" runat="server" Text="ID : " CssClass="patientlabel"></asp:Label>
</td>
<td class="style2">
<asp:TextBox ID="TextBox6" runat="server" CssClass="textbox"
ReadOnly="True"></asp:TextBox>
</td>
</tr>
<tr>
<td class="style129">
<asp:Label ID="Label17" runat="server" Text="Birth date : "
CssClass="patientlabel"></asp:Label>
</td>
<td>
<asp:TextBox ID="TextBox7" runat="server" CssClass="textbox"
ReadOnly="True"></asp:TextBox>
</td>
</tr>
<tr>
<td class="style129">

```

```

        <asp:Label ID="Label18" runat="server" Text="Gender : " CssClass="patientlabel"></asp:Label>
    </td>
    <td>
        <asp:TextBox ID="TextBox8" runat="server" CssClass="textbox"
ReadOnly="True"></asp:TextBox>
    </td>
</tr>
<tr>
    <td class="style129">
        <asp:Label ID="Label19" runat="server" Text="Phone number : "
CssClass="patientlabel"></asp:Label>
    </td>
    <td>
        <asp:TextBox ID="TextBox9" runat="server" CssClass="textbox"
ReadOnly="True"></asp:TextBox>
    </td>
</tr>
<tr>
    <td colspan="2">
        <asp:Button ID="Button100" runat="server" CssClass="button" OnClick="Button100_Click"
Text="Edit Patient" />
    </td>
</tr>
<tr>
    <td colspan="2" align="center">
        <br />
        <asp:Label ID="Label21" runat="server" CssClass="label2"
Text="Press the Visit History panel in order to choose a visit history record from"></asp:Label>
        <br />
        <asp:Label ID="Label22" runat="server" CssClass="label2"
Text=" the dropdown list and display its data (drugs info and quantity
prescribed)."></asp:Label>
    </td>
</tr>
</table>
<br />
<asp:Label ID="Label4" runat="server" Text="Label" Visible="False"></asp:Label>
<br />
<asp:Panel ID="Panel2" runat="server" Style="cursor: pointer;" CssClass="panel" Width="115px">
    <div class="heading">
        &nbsp;&nbsp;&nbsp;<asp:UpdatePanel ID="UpdatePanel2" runat="server">
            <ContentTemplate>
                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<asp:ImageButton ID="Description_ToggleImage1" runat="server"
AlternateText="collapse"
ImageUrl="images/expand.jpg" />
                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Visit History
            </ContentTemplate>
        </asp:UpdatePanel>
    </div>
</asp:Panel>
<asp:RoundedCornersExtender ID="Panel2_RoundedCornersExtender" runat="server" Corners="Top"
Enabled="True" Radius="11" TargetControlID="Panel2">
</asp:RoundedCornersExtender>
<asp:Panel ID="Panel3" runat="server" Style="overflow: hidden;" Height="157px" Width="100%"
CssClass="panel">
    <table style="width: 100%; height: 127px;">
        <tr>
            <td align="center">
                <asp:DropDownList ID="DropDownList3" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="DropDownList3_SelectedIndexChanged"
CssClass="dropdownlist">
            </asp:DropDownList>
            </td>
        </tr>
    </table>
    <td align="center">

```



```

<asp:GridView ID="GridView1" runat="server" BackColor="White" BorderColor="#999999"
  BorderStyle="None" BorderWidth="1px" CellPadding="3" GridLines="Vertical">
  <AlternatingRowStyle BackColor="#99CCFF" />
  <FooterStyle BackColor="#CCCCCC" ForeColor="Black" />
  <HeaderStyle BackColor="#2096BD" Font-Bold="True" ForeColor="White" />
  <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />
  <RowStyle BackColor="#EEEEEE" ForeColor="Black" />
  <SelectedRowStyle BackColor="#008A8C" Font-Bold="True" ForeColor="White" />
  <SortedAscendingCellStyle BackColor="#F1F1F1" />
  <SortedAscendingHeaderStyle BackColor="#0000A9" />
  <SortedDescendingCellStyle BackColor="#CAC9C9" />
  <SortedDescendingHeaderStyle BackColor="#000065" />
</asp:GridView>
<br />
</td>
</tr>
</table>
</asp:Panel>
<br />
<br />
<div align="center" style="width: 100%">
  <br />
  <asp:Label ID="Label23" runat="server"
    Text="Press the New Visit panel in order to create a new visit history record."
    CssClass="label2"></asp:Label>
  <br />
  <asp:Label ID="Label24" runat="server"
    Text="Next, enter the ICD-10 code for the patient's disease and press the Submit button."
    CssClass="label2"></asp:Label>
  <br />
  <asp:Label ID="Label25" runat="server"
    Text="Then choose the drug search method (ICD-10,ATC ingredient, drug brand name)."
    CssClass="label2"></asp:Label>
  <br />
  <asp:Label ID="Label26" runat="server"
    Text="For the selected drug, drugstore quantity is shown and you can enter the desired quantity to
  submit or order."
    CssClass="label2"></asp:Label>
  <br />
  <br />
  <br />
  <br />
</div>
<asp:CollapsiblePanelExtender ID="CollapsiblePanelExtender1" runat="Server"
TargetControlID="Panel3"
  ExpandControlID="Panel2" CollapseControlID="Panel2" Collapsed="True"
ImageControlID="description_ToggleImage1"
  ExpandedImage="images/collapse.jpg" CollapsedImage="images/expand.jpg" />
<asp:Panel ID="Description_HeaderPanel" runat="server" Style="cursor: pointer;" CssClass="panel"
  Width="115px">
  <div class="heading">
    &nbsp;&nbsp;&nbsp;<asp:UpdatePanel ID="UpdatePanel1" runat="server">
      <ContentTemplate>
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<asp:ImageButton ID="Description_ToggleImage"
runat="server"
          AlternateText="collapse" ImageUrl="images/expand.jpg" />
        &nbsp;&nbsp;&nbsp;New Visit
      </ContentTemplate>
    </asp:UpdatePanel>
  </div>
</asp:Panel>
<asp:RoundedCornersExtender ID="Description_HeaderPanel_RoundedCornersExtender" runat="server"
  Corners="Top" Enabled="True" Radius="11" TargetControlID="Description_HeaderPanel">

```

```

</asp:RoundedCornersExtender>
<asp:Panel ID="Description_ContentPanel" runat="server" CssClass="style3" Height="386px">
  <table style="width: 100%;">
    <tr>
      <td align="center">
        <asp:Label ID="Label13" runat="server" CssClass="label" Text="ICD"></asp:Label>
      </td>
    </tr>
    <tr>
      <td align="center">
        <asp:TextBox ID="TextBox1" runat="server" Height="39px" TextMode="MultiLine"
Width="451px"
        autocomplete="off"></asp:TextBox>
        <asp:AutoCompleteExtender ID="TextBox1_AutoCompleteExtender" runat="server"
BehaviorID="AutoCompleteEx"
        CompletionInterval="1000" CompletionListCssClass="autocomplete_completionListElement"
        CompletionListHighlightedItemCssClass="autocomplete_highlightedListItem"
        CompletionListItemCssClass="autocomplete_listItem"
        EnableCaching="true" MinimumPrefixLength="1" ServiceMethod="GetICDList"
        ServicePath="WebService1.asmx"
        TargetControlID="TextBox1">
          <Animations>
            <OnShow>
              <Sequence>
                <!-- Make the completion list transparent and then show it -->
                <OpacityAction Opacity="0" />
                <HideAction Visible="true" />

                <!-- Cache the original size of the completion list the first time
the animation is played and then set it to zero -->
                <ScriptAction Script="
// Cache the size and setup the initial size
var behavior = $find('AutoCompleteEx');
if (!behavior._height) {
  var target = behavior.get_completionList();
  behavior._height = target.offsetHeight - 2;
  target.style.height = '0px';
}" />

                <!-- Expand from 0px to the appropriate size while fading in -->
                <Parallel Duration=".4">
                  <FadeIn />
                  <Length PropertyKey="height" StartValue="0"
EndValueScript="$find('AutoCompleteEx')._height" />
                </Parallel>
              </Sequence>
            </OnShow>
            <OnHide>
              <!-- Collapse down to 0px and fade out -->
              <Parallel Duration=".4">
                <FadeOut />
                <Length PropertyKey="height" StartValueScript="$find('AutoCompleteEx')._height"
EndValue="0" />
              </Parallel>
            </OnHide>
          </Animations>
        </asp:AutoCompleteExtender>
      </td>
    </tr>
    <tr>
      <td align="center">
        <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="Submit"
CssClass="button" />
      </td>
    </tr>
    <tr>
      <td align="center">

```

```

<asp:Panel ID="Panel1" runat="server" Height="277px" Visible="False">
  <table align="center" width="100%">
    <tr>
      <td align="right" width="50%">
        <asp:Label ID="Label14" runat="server" CssClass="label" Text="Drug search method
: "></asp:Label>
      </td>
      <td align="left">
        <asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
Height="21px"
        OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged"
        CssClass="dropdownlist">
          <asp:ListItem Selected="True">Choose</asp:ListItem>
          <asp:ListItem>Brand Name</asp:ListItem>
          <asp:ListItem>Ingredient (ATC)</asp:ListItem>
          <asp:ListItem>ICD (disease)</asp:ListItem>
        </asp:DropDownList>
      </td>
    </tr>
    <tr>
      <td align="right">
        <asp:Label ID="Label2" runat="server" Visible="False"
        CssClass="label"></asp:Label>
      </td>
      <td align="left" class="style1">
        <asp:TextBox ID="TextBox2" runat="server" Height="39px" TextMode="MultiLine"
        Visible="False"
        Width="276px" autocomplete="off"></asp:TextBox>
        <asp:AutoCompleteExtender ID="TextBox2_AutoCompleteExtender" runat="server"
        BehaviorID="AutoCompleteEx1"
        CompletionInterval="1000"
        CompletionListCssClass="autocomplete_completionListElement"
        CompletionListHighlightedItemCssClass="autocomplete_highlightedListItem"
        CompletionListItemCssClass="autocomplete_listItem"
        EnableCaching="true" Enabled="True" MinimumPrefixLength="1"
        ServiceMethod="GetDrugList"
        ServicePath="WebService1.aspx" TargetControlID="TextBox2">
          <Animations>
            <OnShow>
              <Sequence>
                <!-- Make the completion list transparent and then show it -->
                <OpacityAction Opacity="0" />
                <HideAction Visible="true" />
                <!-- Cache the original size of the completion list the first time
                the animation is played and then set it to zero -->
                <ScriptAction Script="
                // Cache the size and setup the initial size
                var behavior = $find('AutoCompleteEx1');
                if (!behavior._height) {
                  var target = behavior.get_completionList();
                  behavior._height = target.offsetHeight - 2;
                  target.style.height = '0px';
                }" />
                <!-- Expand from 0px to the appropriate size while fading in -->
                <Parallel Duration=".4">
                  <FadeIn />
                  <Length PropertyKey="height" StartValue="0"
                EndValueScript="$find('AutoCompleteEx1')._height" />
                </Parallel>
              </Sequence>
            </OnShow>
            <OnHide>
              <!-- Collapse down to 0px and fade out -->
              <Parallel Duration=".4">
                <FadeOut />

```

```

        <Length PropertyKey="height" StartValueScript="$find('AutoCompleteEx1')._height"
EndValue="0" />
    </Parallel>
</OnHide>
    </Animations>
</asp:AutoCompleteExtender>
<br />
<asp:TextBox ID="TextBox3" runat="server" Height="39px" TextMode="MultiLine"
Visible="False"
    Width="277px" autocomplete="off"></asp:TextBox>
<asp:AutoCompleteExtender ID="TextBox3_AutoCompleteExtender" runat="server"
BehaviorID="AutoCompleteEx2"
    CompletionInterval="1000"
CompletionListCssClass="autocomplete_completionListElement"
    CompletionListHighlightedItemCssClass="autocomplete_highlightedListItem"
CompletionListItemCssClass="autocomplete_listItem"
    EnableCaching="true" Enabled="True" MinimumPrefixLength="1"
ServiceMethod="GetIngredientList"
    ServicePath="WebService1.asmx" TargetControlID="TextBox3">
    <Animations>
<OnShow>
    <Sequence>
        <!-- Make the completion list transparent and then show it -->
        <OpacityAction Opacity="0" />
        <HideAction Visible="true" />

        <!-- Cache the original size of the completion list the first time
the animation is played and then set it to zero -->
        <ScriptAction Script="
// Cache the size and setup the initial size
var behavior = $find('AutoCompleteEx2');
if (!behavior._height) {
    var target = behavior.get_completionList();
    behavior._height = target.offsetHeight - 2;
    target.style.height = '0px';
}" />

        <!-- Expand from 0px to the appropriate size while fading in -->
        <Parallel Duration=".4">
            <FadeIn />
            <Length PropertyKey="height" StartValue="0"
EndValueScript="$find('AutoCompleteEx2')._height" />
        </Parallel>
    </Sequence>
</OnShow>
<OnHide>
        <!-- Collapse down to 0px and fade out -->
        <Parallel Duration=".4">
            <FadeOut />
            <Length PropertyKey="height" StartValueScript="$find('AutoCompleteEx2')._height"
EndValue="0" />
        </Parallel>
    </OnHide>
    </Animations>
</asp:AutoCompleteExtender>
</td>
</tr>
<tr>
    <td align="center" colspan="2">
        <asp:Button ID="Button3" runat="server" OnClick="Button3_Click" Text="Next"
Visible="False"
    CssClass="button" />
    </td>
</tr>
<tr>
    <td align="right">

```



```

using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1
{
    public partial class Visit_History : System.Web.UI.Page
    {
        string Patient_Id;
        Class1 ab = new Class1();
        string[] DrugId;
        string[] abc = { "=" };
        string[] split_store_quantity = { ": " };
        protected void Page_Load(object sender, EventArgs e)
        {
            Patient_Id = Request.QueryString["Patient_Id"];
            if (!Page.IsPostBack)
            {
                if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))
                    Response.Redirect("Login.aspx");
                List<PATIENT_Table> PatientList = ab.GetPatient_by_Patient_Id(int.Parse(Patient_Id));
                TextBox5.Text = PatientList[0].Patient_Name;
                TextBox6.Text = PatientList[0].ID_Number;
                TextBox7.Text = PatientList[0].Birth_Date;
                TextBox8.Text = PatientList[0].Gender;
                TextBox9.Text = PatientList[0].Phone_Number;
                DropDownList3.DataSource = ab.GetVisit_History_by_Patient_Id(int.Parse(Patient_Id));
                DropDownList3.DataBind();
                DropDownList3.Items.Insert(0, new ListItem("Choose", "Choose"));
            }
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            Label4.Text = ab.SetVisits_Table(int.Parse(Patient_Id), TextBox1.Text.Substring(0, 10)).ToString();
            Button2.Visible = false;
            Panel1.Visible = true;
            TextBox1.ReadOnly = true;
            Label12.Visible = false;
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string[] quantity = Label10.Text.Split(split_store_quantity, StringSplitOptions.RemoveEmptyEntries);

            if (DropDownList1.SelectedItem.Text.Equals("Brand Name"))
            {
                if (int.Parse(TextBox4.Text) <= int.Parse(quantity[1]))
                {
                    DrugId = TextBox2.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
                    ab.SetDrugPerVisit_Table(int.Parse(Label4.Text), int.Parse(DrugId[1]), int.Parse(TextBox4.Text));
                    ab.UpdateDrugstoreQuantityByDrugId(int.Parse(DrugId[1]), int.Parse(TextBox4.Text));
                    Label11.Visible = false;
                    Label12.Text = "Submit successful";
                    Label12.Visible = true;
                }
                else
                {
                    Label11.Visible = true;
                    DrugId = TextBox2.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
                }
            }
            else if (DropDownList1.SelectedItem.Text.Equals("Ingredient (ATC)"))
            {

```

```

if (int.Parse(TextBox4.Text) <= int.Parse(quantity[1]))
{
    DrugId = DropDownList2.SelectedItem.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
    ab.SetDrugPerVisit_Table(int.Parse(Label4.Text), int.Parse(DrugId[1]), int.Parse(TextBox4.Text));
    ab.UpdateDrugstoreQuantityByDrugId(int.Parse(DrugId[1]), int.Parse(TextBox4.Text));
    Label11.Visible = false;
    Label12.Text = "Submit successful";
    Label12.Visible = true;
}
else
{
    Label11.Visible = true;
    DrugId = DropDownList2.SelectedItem.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
}
}
else if (DropDownList1.SelectedItem.Text.Equals("ICD (disease)"))
{
    if (int.Parse(TextBox4.Text) <= int.Parse(quantity[1]))
    {
        DrugId = DropDownList2.SelectedItem.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
        ab.SetDrugPerVisit_Table(int.Parse(Label4.Text), int.Parse(DrugId[1]), int.Parse(TextBox4.Text));
        ab.UpdateDrugstoreQuantityByDrugId(int.Parse(DrugId[1]), int.Parse(TextBox4.Text));
        Label11.Visible = false;
        Label12.Text = "Submit successful";
        Label12.Visible = true;
    }
    else
    {
        Label11.Visible = true;
        DrugId = DropDownList2.SelectedItem.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
    }
}
}
Label10.Text = "Drug store quantity: " + ab.GetQuantity_by_Drug_Id(int.Parse(DrugId[1]));
DropDownList3.DataSource = ab.GetVisit_History_by_Patient_Id(int.Parse(Patient_Id));
DropDownList3.DataBind();
DropDownList3.Items.Insert(0, new ListItem("Choose", "Choose"));
Label12.Visible = false;
}

```

```

protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (DropDownList1.SelectedItem.Text.Equals("Brand Name"))
    {
        Label3.Visible = false;
        Button3.Visible = true;
        Button1.Visible = false;
        Button4.Visible = false;
        TextBox3.Visible = false;
        Label2.Visible = true;
        Label2.Text = "Search drug";
        TextBox2.Visible = true;
        DropDownList2.Visible = false;
    }
    else if (DropDownList1.SelectedItem.Text.Equals("Ingredient (ATC)"))
    {
        Label3.Visible = false;
        DropDownList2.Visible = false;
        Button1.Visible = false;
        Button4.Visible = false;
        TextBox2.Visible = false;
        Label2.Visible = true;
        Label2.Text = "Search ingredient";
        TextBox3.Visible = true;
        Button3.Visible = true;
    }
}

```

```

else if (DropDownList1.SelectedItem.Text.Equals("ICD (disease)"))
{
    Label3.Visible = true;
    Button3.Visible = false;
    TextBox2.Visible = false;
    Label2.Visible = false;
    TextBox3.Visible = false;
    Button1.Visible = false;
    Button4.Visible = false;
    DropDownList2.DataSource = ab.GetAllDrugsByICD_Id(TextBox1.Text.Substring(0, 10));
    DropDownList2.DataBind();
    DropDownList2.Items.Insert(0, new ListItem("Choose", "Choose"));
    DropDownList2.Visible = true;
}

Label10.Visible = false;
Label11.Visible = false;
TextBox4.Visible = false;
Label9.Visible = false;
Label12.Visible = false;
}

protected void Button3_Click(object sender, EventArgs e)
{
    if (DropDownList1.SelectedItem.Text.Equals("Ingredient (ATC)"))
    {
        DropDownList2.DataSource = ab.GetAllDrugsByIngredient(TextBox3.Text);
        DropDownList2.DataBind();
        DropDownList2.Items.Insert(0, new ListItem("Choose", "Choose"));
        DropDownList2.Visible = true;
        Button3.Visible = false;
        Button1.Visible = false;
        Button4.Visible = false;
        Label3.Visible = true;
        Label9.Visible = false;
        Label10.Visible = false;
        Label11.Visible = false;
        TextBox4.Visible = false;
    }

    else if (DropDownList1.SelectedItem.Text.Equals("Brand Name"))
    {
        DropDownList2.Visible = false;
        Button3.Visible = false;
        Button1.Visible = true;
        Button4.Visible = true;
        Label3.Visible = false;
        Label9.Visible = true;
        Label10.Visible = true;
        Label11.Visible = false;
        TextBox4.Visible = true;
        DrugId = TextBox2.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
        Label10.Text = "Drug store quantity: " + ab.GetQuantity_by_Drug_Id(int.Parse(DrugId[1]));
    }
    Label12.Visible = false;
}

protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
{
    Button1.Visible = true;
    Button4.Visible = true;
}

```



```

Label3.Visible = true;
Label9.Visible = true;
Label10.Visible = true;
Label11.Visible = false;
TextBox4.Visible = true;
DrugId = DropDownList2.SelectedItem.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
Label10.Text = "Drug store quantity: " + ab.GetQuantity_by_Drug_Id(int.Parse(DrugId[1]));
Label12.Visible = false;
}

protected void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
{
    string[] VisitId = DropDownList3.SelectedItem.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
    GridView1.DataSource = ab.GetDrugPerVisit_by_Visit_Id(int.Parse(VisitId[1]));
    GridView1.DataBind();
}

protected void Button4_Click(object sender, EventArgs e)
{
    if (DropDownList1.SelectedItem.Text.Equals("Ingredient (ATC)") ||
    DropDownList1.SelectedItem.Text.Equals("ICD (disease)"))
    {
        DrugId = DropDownList2.SelectedItem.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
        ab.SetDrugOrderByDrugId(int.Parse(DrugId[1]), int.Parse(TextBox4.Text));
    }
    else if (DropDownList1.SelectedItem.Text.Equals("Brand Name"))
    {
        DrugId = TextBox2.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
        ab.SetDrugOrderByDrugId(int.Parse(DrugId[1]), int.Parse(TextBox4.Text));
    }
    Label12.Text = "Order successful";
    Label12.Visible = true;
}

protected void Button100_Click(object sender, EventArgs e)
{
    if (Button100.Text.Equals("Edit Patient"))
    {
        TextBox5.CssClass = "defaulttextbox";
        TextBox6.CssClass = "defaulttextbox";
        TextBox7.CssClass = "defaulttextbox";
        TextBox8.CssClass = "defaulttextbox";
        TextBox9.CssClass = "defaulttextbox";
        TextBox5.ReadOnly = false;
        TextBox6.ReadOnly = false;
        TextBox7.ReadOnly = false;
        TextBox8.ReadOnly = false;
        TextBox9.ReadOnly = false;
        Button100.Text = "Update";
    }
    else
    {
        string Patient_Id1 = Request.QueryString["Patient_Id"];
        ab.UpdatePatient(int.Parse(Patient_Id1), TextBox6.Text, TextBox5.Text, TextBox8.Text,
        TextBox7.Text, TextBox9.Text);
        Button100.Text = "Edit Patient";
        TextBox5.CssClass = "textbox";
        TextBox6.CssClass = "textbox";
        TextBox7.CssClass = "textbox";
        TextBox8.CssClass = "textbox";
        TextBox9.CssClass = "textbox";
        TextBox5.ReadOnly = true;
        TextBox6.ReadOnly = true;
        TextBox7.ReadOnly = true;
        TextBox8.ReadOnly = true;
    }
}

```

```

        TextBox9.ReadOnly = true;
    }
}
}
}

```

## Warehouse.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="~/HomeMasterPage.Master"
CodeBehind="Warehouse.aspx.cs" Inherits="WebApplication1.Warehouse" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
    <title>Warehouse</title>
    <link href="Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
    <style type="text/css">

</style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
    <table style="width: 100%;">
        <tr>
            <td align="center">
                <asp:Label ID="Label1" runat="server" CssClass="label2"
                    Text="In the table below all unsatisfied orders are shown."></asp:Label>
                <br />
                <asp:Label ID="Label2" runat="server" CssClass="label2"
                    Text="Select one or more orders and press the commit button"></asp:Label>
                <br />
                <asp:Label ID="Label3" runat="server" CssClass="label2"
                    Text="In order to select all orders press the Check All button."></asp:Label>
                <br />
                <br />
            </td>
            <td align="center">
                <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Check All"
                    CssClass="button" />
            </td>
            <td align="center">
                <asp:GridView ID="GridView1" runat="server" BackColor="White" BorderColor="#999999"
                    BorderStyle="None" BorderWidth="1px" CellPadding="3" GridLines="Vertical">
                    <AlternatingRowStyle BackColor="#DCDCDC" />
                    <Columns>
                        <asp:TemplateField HeaderText="Check">
                            <EditItemTemplate>
                                <asp:CheckBox ID="CheckBox1" runat="server" />
                            </EditItemTemplate>
                            <ItemTemplate>
                                <asp:CheckBox ID="CheckBox1" runat="server" />
                            </ItemTemplate>
                        </asp:TemplateField>
                    </Columns>
                    <FooterStyle BackColor="#CCCCCC" ForeColor="Black" />
                    <HeaderStyle BackColor="#2096BD" Font-Bold="True" ForeColor="White" />
                    <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />
                    <RowStyle BackColor="#EEEEEE" ForeColor="Black" />
                    <SelectedRowStyle BackColor="#008A8C" Font-Bold="True" ForeColor="White" />
                    <SortedAscendingCellStyle BackColor="#F1F1F1" />
                    <SortedAscendingHeaderStyle BackColor="#0000A9" />
                    <SortedDescendingCellStyle BackColor="#CAC9C9" />
                    <SortedDescendingHeaderStyle BackColor="#000065" />
                </asp:GridView>
            </td>
        </tr>
    </table>

```

```

        </tr>
        <tr>
            <td align="center">
                <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="Commit"
                CssClass="button" />
            </td>
        </tr>
    </table>
</asp:Content>

```

### Warehouse.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1
{
    public partial class Warehouse : System.Web.UI.Page
    {
        Class1 ab = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {
                if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))
                    Response.Redirect("Login.aspx");
                GridView1.DataSource = ab.GetAllDrugOrders();
                GridView1.DataBind();
            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            for (int i = 0; i < GridView1.Rows.Count; i++)
            {
                GridViewRow row = GridView1.Rows[i];
                ((CheckBox)row.FindControl("CheckBox1")).Checked=true;
            }
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            for (int i = 0; i < GridView1.Rows.Count; i++)
            {
                GridViewRow row = GridView1.Rows[i];
                bool isChecked = ((CheckBox)row.FindControl("CheckBox1")).Checked;

                if (isChecked)
                {
                    ab.CommitDrugOrdersByDrug_Id(int.Parse(row.Cells[6].Text));
                }
            }
            GridView1.DataSource = ab.GetAllDrugOrders();
            GridView1.DataBind();
        }
    }
}

```

```
}  
}
```

### WebService1.asmx

```
using System;  
using System.Collections;  
using System.Collections.Generic;  
using System.Configuration;  
using System.Data;  
using System.Data.SqlClient;  
using System.Linq;  
using System.Web;  
using System.Web.Services;  
using System.Web.Services.Protocols;  
  
namespace WebApplication1  
{  
    /// <summary>  
    /// Summary description for WebService1  
    /// </summary>  
    [WebService(Namespace = "http://tempuri.org/")]  
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]  
    [System.ComponentModel.ToolboxItem(false)]  
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.  
    [System.Web.Script.Services.ScriptService]  
    public class WebService1 : System.Web.Services.WebService  
    {  
        [WebMethod]  
        public string[] GetCompletionList(string prefixText, int count)  
        {  
            DataClasses1DataContext db = new DataClasses1DataContext();  
            return (from m in db.ATC_Tables where m.ATC_Id.StartsWith(prefixText) select m.ATC_Id).ToArray();  
        }  
  
        [WebMethod]  
        public string[] GetICDList(string prefixText, int count)  
        {  
            DataClasses1DataContext db = new DataClasses1DataContext();  
            var query = from p in db.ICD_Tables  
                       where p.ICD_Id.StartsWith(prefixText)  
                       select p;  
  
            string[] items = new string[query.Count()];  
            //string items;  
  
            // return query.Single();  
            int i = 0;  
            foreach (var item in query)  
            {  
                items[i] = item.ICD_Id + item.Diseases;  
                i++;  
            }  
            return items;  
            //return (from m in db.ATC_Tables where m.ATC_Id.StartsWith(prefixText) select m.ATC_Id).ToArray();  
        }  
  
        [WebMethod]  
        public string[] GetDrugList(string prefixText, int count)  
        {  

```

```

DataClasses1DataContext db = new DataClasses1DataContext();
var query = from p in db DrugStore_Tables
            where p.DRUG_Table.DRUG_Name.StartsWith(prefixText)
            select p.DRUG_Table;

string[] items = new string[query.Count()];
//string items;

// return query.Single();
int i = 0;
foreach (var item in query)
{
    items[i] = item.DRUG_Name + " " + item.DRUG_Supplier + " " + item.DRUG_Dosage + " " +
item.DRUG_Package + " id=" + item.DRUG_Id;
    i++;
}
return items;
//return (from m in db.ATC_Tables where m.ATC_Id.StartsWith(prefixText) select m.ATC_Id).ToArray();
}

[WebMethod]
public string[] GetIngredientList(string prefixText, int count)
{
    DataClasses1DataContext db = new DataClasses1DataContext();

    return (from m in db.ATC_Tables where m.Ingredient.StartsWith(prefixText) select
m.Ingredient).Distinct().ToArray();
}

}
}

```

### DoctorMasterPage.Master

```

<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="DoctorMasterPage.master.cs"
    Inherits="WebApplication1.DoctorMasterPage" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
    <meta name="description" content="Your description goes here." />
    <meta name="keywords" content="your,keywords,goes,here" />
    <meta name="author" content="Your Name / Original design by Andreas Viklund - http://andreasviklund.com"
/>
    <link rel="stylesheet" href="andreas09.css" type="text/css" media="screen" />
    <style type="text/css">
        .style123
        {
            width: 100%;
            height: 124px;
        }
    </style>
</head>
<body style="width: 1024px; margin: 0 auto;">
    <form id="form1" runat="server">
    <div>
        <div id="container">

```



```
</html>
```

## Orders.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="~/DoctorMasterPage.Master"
    CodeBehind="Orders.aspx.cs" Inherits="WebApplication1.Orders" %>
```

```
<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
```

```
<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
```

```
    <title>Drug Order</title>
```

```
    <link href="Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
```

```
    <link href="Styles/StyleSheet1.css" rel="stylesheet" type="text/css" />
```

```
    <style type="text/css">
```

```
    </style>
```

```
</asp:Content>
```

```
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
```

```
    <span id="tabid" style="display: none">OrdersTab</span>
```

```
    <asp:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
```

```
    </asp:ToolkitScriptManager>
```

```
    <asp:UpdatePanel ID="UpdatePanel1" runat="server">
```

```
        <ContentTemplate>
```

```
            <table style="width: 100%;">
```

```
                <tr>
```

```
                    <td align="center">
```

```
                        <asp:Label ID="Label5" runat="server"
```

```
                            Text="Type the name of drug you would like to order and press the Next button. Autocomplete is
used when typing."
                            CssClass="label2"></asp:Label>
```

```
                        <br />
```

```
                        <asp:Label ID="Label6" runat="server"
```

```
                            Text="Next, the drugstore quantity is shown. You can enter the desired quantity and then press
the Order Drug button. "
```

```
                            CssClass="label2"></asp:Label>
```

```
                        <br />
```

```
                        <asp:Label ID="Label7" runat="server"
```

```
                            Text="If &quot;order successful&quot; appears, your order has been completed."
                            CssClass="label2"></asp:Label>
```

```
                        <br />
```

```
                        <br />
```

```
                    </td>
```

```
                </tr>
```

```
            <tr>
```

```
                <td align="center">
```

```
                    <asp:Label ID="Label4" runat="server" CssClass="label"
```

```
                        Text="Choose drug to order : "></asp:Label>
```

```
                    <asp:TextBox ID="TextBox1" runat="server" autocomplete="off" Height="22px"
```

```
                        Width="324px"></asp:TextBox>
```

```
                    <asp:AutoCompleteExtender ID="TextBox1_AutoCompleteExtender" runat="server"
```

```
                        BehaviorID="AutoCompleteEx" CompletionInterval="1000"
```

```
                        CompletionListCssClass="autocomplete_completionListElement"
```

```
                        CompletionListHighlightedItemCssClass="autocomplete_highlightedListItem"
```

```
                        CompletionListItemCssClass="autocomplete_listItem" EnableCaching="true"
```

```
                        MinimumPrefixLength="1" ServiceMethod="GetDrugList"
```

```
                        ServicePath="WebService1.asmx" TargetControlID="TextBox1">
```

```
                        <Animations>
```

```
                            <OnShow>
```

```
                                <Sequence>
```

```
                                    <%-- Make the completion list transparent and then show it --%>
```

```
                                    <OpacityAction Opacity="0" />
```

```
                                    <HideAction Visible="true" />
```

```
                                <%--Cache the original size of the completion list the first time
the animation is played and then set it to zero --%>
```

```

        <ScriptAction Script="
            // Cache the size and setup the initial size
            var behavior = $find('AutoCompleteEx');
            if (!behavior._height) {
                var target = behavior.get_completionList();
                behavior._height = target.offsetHeight - 2;
                target.style.height = '0px';
            }" />

        <!-- Expand from 0px to the appropriate size while fading in -->
        <Parallel Duration=".4">
            <FadeIn />
            <Length PropertyKey="height" StartValue="0"
EndValueScript="$find('AutoCompleteEx')._height" />
        </Parallel>
    </Sequence>
</OnShow>
<OnHide>
    <!-- Collapse down to 0px and fade out -->
    <Parallel Duration=".4">
        <FadeOut />
        <Length PropertyKey="height" StartValueScript="$find('AutoCompleteEx')._height"
EndValue="0" />
    </Parallel>
</OnHide>
</Animations>
</asp:AutoCompleteExtender>
</td>
</tr>
<tr>
    <td align="center">
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Next"
CssClass="button" />
    </td>
</tr>
<tr>
    <td align="center">
        <asp:Label ID="Label2" runat="server" Text="Enter quantity : " Visible="False"
CssClass="label"></asp:Label>
        &nbsp;&nbsp;&nbsp;<asp:TextBox ID="TextBox2" runat="server" Height="21px" Visible="False"
Width="41px"
        autocomplete="false"></asp:TextBox>
        &nbsp;&nbsp;&nbsp;<asp:Label ID="Label1" runat="server" Visible="False" CssClass="label"></asp:Label>
        <br />
    </td>
</tr>
<tr>
    <td align="center">
        <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="Order Drug"
Visible="False" CssClass="button" />
    </td>
</tr>
<tr>
    <td align="center">
        <asp:Label ID="Label3" runat="server" Text="Order successful" Visible="False"
CssClass="resullabel"></asp:Label>
    </td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</asp:Content>

```

### Orders.aspx.cs

```

using System;
using System.Collections.Generic;

```



```

using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1
{
    public partial class Orders : System.Web.UI.Page
    {
        string[] abc = { "=" };
        string[] DrugId;
        Class1 ab = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))
                Response.Redirect("Login.aspx");
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            Label1.Visible = true;
            Label2.Visible = true;
            TextBox2.Visible = true;
            DrugId = TextBox1.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
            Label1.Text="Drug store quantity: " + ab.GetQuantity_by_Drug_Id(int.Parse(DrugId[1]));
            Button2.Visible = true;
            Label3.Visible = false;
            TextBox2.Text = "";
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            DrugId = TextBox1.Text.Split(abc, StringSplitOptions.RemoveEmptyEntries);
            ab.SetDrugOrderByDrugId(int.Parse(DrugId[1]), int.Parse(TextBox2.Text));
            Label3.Visible = true;
            Label1.Visible = false;
            Button2.Visible = false;
            Label2.Visible = false;
            TextBox2.Visible = false;
        }
    }
}

```

#### Class1.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1
{
    public class Class1 : System.Web.Services.WebService
    {
        DataClasses1DataContext db = new DataClasses1DataContext();

        public int SetPatientAndGetPatient_Id(String ID_Number, String Patient_Name, String Gender, String
        Birth_Date, String Phone_Number)
        {
            PATIENT_Table patient = new PATIENT_Table();
            patient.ID_Number = ID_Number;

```

```

    patient.Patient_Name = Patient_Name;
    patient.Gender = Gender;
    patient.Birth_Date = Birth_Date;
    patient.Phone_Number = Phone_Number;
    db.PATIENT_Tables.InsertOnSubmit(patient);
    db.SubmitChanges();
    return patient.Patient_Id;
}

public System.Collections.IEnumerable GetAllPatients()
{
    var query = from p in db.PATIENT_Tables
                select p;
    return query;
}

public String[] GetAllPatientsName()
{
    //
    var query = from p in db.PATIENT_Tables
                orderby p.Patient_Name
                select p;

    string[] items = new string[query.Count()];
    //string items;

    // return query.Single();
    int i = 0;
    foreach (var item in query)
    {
        items[i] = item.Patient_Name + " " + item.Birth_Date;
        i++;
    }
    return items;
}

public void SetATC(String ATC_Id, String Ingredient)
{
    ATC_Table atc = new ATC_Table();
    atc.ATC_Id = ATC_Id;
    atc.Ingredient = Ingredient;
    db.ATC_Tables.InsertOnSubmit(atc);
    db.SubmitChanges();
}

public string GetPatient_id_by_Name_And_Birth_Date(String name_and_birth_date)
{
    String name = name_and_birth_date.Substring(0, name_and_birth_date.Length - 11);
    String birth_date = name_and_birth_date.Substring(name_and_birth_date.Length - 10, 10);
    var query = from p in db.PATIENT_Tables
                where p.Patient_Name.Equals(name) && p.Birth_Date.Equals(birth_date)
                select p.Patient_Id;
    return query.Single().ToString();
}

public int SetVisits_Table(int id, String ICD_id)
{
    VISITS_Table prog = new VISITS_Table();
    prog.Patient_Id = id;
}

```

```

prog.ICD_Id = ICD_id;
prog.Prescription_Date = System.DateTime.Today.ToShortDateString();
db.VISITS_Tables.InsertOnSubmit(prog);
db.SubmitChanges();
return prog.Visit_Id;
////////// to teleutaio soixei den ginotane
//int query = (from p in db.VISITS_Tables

//      select p.Visit_Id).Max();
//return query;
}

public void SetDrugPerVisit_Table(int Visit_Id, int Drug_Id, int Drug_Quantity)
{

    DrugsPerVisit_Table prog = new DrugsPerVisit_Table();
    prog.Visit_Id = Visit_Id;
    prog.Drug_Id = Drug_Id;
    prog.Drug_Quantity = Drug_Quantity;
    db.DrugsPerVisit_Tables.InsertOnSubmit(prog);
    db.SubmitChanges();
}

public String[] GetAllDrugsByIngredient(string Ingredient)
{

    var query = from p in db.ATC_DrugStore_Tables
                where p.ATC_Table.Ingredient.Equals(Ingredient)
                select p.DrugStore_Table.DRUG_Table;
    string[] items = new string[query.Count()];
    //string items;

    // return query.Single();
    int i = 0;
    foreach (var item in query)
    {

        items[i] = item.DRUG_Name + " " + item.DRUG_Supplier + " " + item.DRUG_Dosage + " " +
item.DRUG_Package + " id=" + item.DRUG_Id;
        i++;
    }
    return items;
}

public System.Collections.IEnumerable SearchAllDrugsByIngredient(string Ingredient)
{

    var query = from p in db.ATC_DRUG_Tables
                where p.ATC_Table.Ingredient.Equals(Ingredient)
                select new
                {
                    p.DRUG_Table.DRUG_Name,
                    p.DRUG_Table.DRUG_Supplier,
                    p.DRUG_Table.DRUG_Dosage,
                    p.DRUG_Table.DRUG_Package,
                    p.DRUG_Table.ROUTE_Table.ROUTE_Name
                };

    return query;
}

public System.Collections.IEnumerable SearchAllDrugsByICD_Id(string ICD_Id)
{

    var query = from p in db.ATC_DRUG_Tables

```

```

        where (from d in db.ATC_ICD_Tables
              where d.ICD_Table.ICD_Id.Equals(ICD_Id)
              select d.ATC_Table).Contains(p.ATC_Table)
select new
{
    p.DRUG_Table.DRUG_Name,
    p.DRUG_Table.DRUG_Supplier,
    p.DRUG_Table.DRUG_Dosage,
    p.DRUG_Table.DRUG_Package,
    p.DRUG_Table.ROUTE_Table.ROUTE_Name
};

return query;
}

public String[] GetAllDrugsByICD_Id(string ICD_Id)
{
    var query = from p in db.ATC_DrugStore_Tables
                where (from d in db.ATC_ICD_Tables
                      where d.ICD_Table.ICD_Id.Equals(ICD_Id)
                      select d.ATC_Table).Contains(p.ATC_Table)
                select p.DrugStore_Table.DRUG_Table;
    string[] items = new string[query.Count()];
    //string items;

    // return query.Single();
    int i = 0;
    foreach (var item in query)
    {
        items[i] = item.DRUG_Name + " " + item.DRUG_Supplier + " " + item.DRUG_Dosage + " " +
item.DRUG_Package + " id=" + item.DRUG_Id;
        i++;
    }
    return items;
}

public System.Collections.IEnumerable GetAllDrugsByCATEGORY_Name(string CATEGORY_Name)
{
    var query = from p in db.ATC_DRUG_Tables
                where (from d in db.CATEGORY_ATC_Tables
                      where d.CATEGORY_Name.Equals(CATEGORY_Name)
                      select d.ATC_Table).Contains(p.ATC_Table)
                select new
                {
                    p.DRUG_Table.DRUG_Name,
                    p.DRUG_Table.DRUG_Supplier,
                    p.DRUG_Table.DRUG_Dosage,
                    p.DRUG_Table.DRUG_Package,
                    p.DRUG_Table.ROUTE_Table.ROUTE_Name
                };

    return query;
}

public System.Collections.IEnumerable GetAllDrugCategories()
{
    var query = (from p in db.CATEGORY_ATC_Tables
                 select p.CATEGORY_Name).Distinct();
}

```

```

        return query;
    }

    public List<PATIENT_Table> GetPatient_by_Patient_Id(int Patient_Id)
    {
        var query = from p in db.PATIENT_Tables
                    where p.Patient_Id == Patient_Id
                    select p;
        return query.ToList();
    }

    public string[] GetVisit_History_by_Patient_Id(int Patient_Id)
    {
        var query = from p in db.VISITS_Tables
                    where p.Patient_Id == Patient_Id
                    select p;
        string[] items = new string[query.Count()];
        int i = 0;
        foreach (var item in query)
        {
            items[i] = item.Prescription_Date + " " + item.ICD_Id + " " + (db.ICD_Tables.Where(l => l.ICD_Id ==
item.ICD_Id).Select(l => l.Diseases)).Single() + " id=" + item.Visit_Id;
            i++;
        }
        return items;
    }

    public string GetQuantity_by_Drug_Id(int Drug_Id)
    {
        var query = from p in db.DrugStore_Tables
                    where p.Drug_Id == Drug_Id
                    select p.Quantity;

        return query.Single().ToString();
    }

    public System.Collections.IEnumerable GetDrugPerVisit_by_Visit_Id(int Visit_Id)
    {
        var query = from p in db.DrugsPerVisit_Tables
                    where p.Visit_Id == Visit_Id
                    select new
                    {
                        Drug_Name = db.DRUG_Tables.Where(l => l.DRUG_Id == p.Drug_Id).Select(r =>
r.DRUG_Name).Single(),
                        Drug_Supplier = db.DRUG_Tables.Where(l => l.DRUG_Id == p.Drug_Id).Select(r =>
r.DRUG_Supplier).Single(),
                        Drug_Dosage = db.DRUG_Tables.Where(l => l.DRUG_Id == p.Drug_Id).Select(r =>
r.DRUG_Dosage).Single(),
                        Drug_Package = db.DRUG_Tables.Where(l => l.DRUG_Id == p.Drug_Id).Select(r =>
r.DRUG_Package).Single(),
                        p.Drug_Quantity
                    };

        return query;
    }

    public void UpdateDrugstoreQuantityByDrugId(int Drug_Id, int Drug_Quantity)
    {
        DrugStore_Table query = (from p in db.DrugStore_Tables

```

```

        where p.Drug_Id == Drug_Id
        select p).Single();

query.Quantity -= Drug_Quantity;
db.SubmitChanges();
}

public void SetDrugOrderByDrugId(int Drug_Id, int Drug_Quantity)
{

    DrugOrder_Table query = (from p in db.DrugOrder_Tables
        where p.Drug_Id == Drug_Id
        select p).Single();

query.Quantity += Drug_Quantity;
db.SubmitChanges();
}

public TreeView j0(TreeView TreeView2)
{
    var query = (from p in db.TreeView_Tables
        where p.atc.StartsWith("/")
        select p.atc).AsEnumerable();

    int i = 0;

    int end_of_query = query.Count();
    while (i < end_of_query)
    {
        TreeNode treeNode0 = new TreeNode(query.ElementAt(i).Substring(1, query.ElementAt(i).Length - 1));
        TreeView2.Nodes.Add(treeNode0);
        i++;
    }

    return TreeView2;
}

public void j1(string nodetext, TreeNode treenode)
{
    if (nodetext.Length == 1)
    {
        var query = (from p in db.TreeView_Tables
            where p.atc.StartsWith("o" + nodetext)
            select p.atc).AsEnumerable();

        int i = 0;

        int end_of_query = query.Count();
        while (i < end_of_query)
        {
            TreeNode treeNode0 = new TreeNode(query.ElementAt(i).Substring(1, query.ElementAt(i).Length -
1));
            treenode.ChildNodes.Add(treeNode0);
            i++;
        }
    }
    else if (nodetext.Length == 3)
    {
        var query = (from p in db.TreeView_Tables
            where p.atc.StartsWith"+" + nodetext)
            select p.atc).AsEnumerable();

```

```

        int i = 0;

        int end_of_query = query.Count();
        while (i < end_of_query)
        {
            TreeNode treeNode0 = new TreeNode(query.ElementAt(i).Substring(1, query.ElementAt(i).Length -
1));
            treeNode.ChildNodes.Add(treeNode0);
            i++;
        }
    }

    else if (nodetext.Length == 4)
    {
        var query = (from p in db.TreeView_Tables
                    where p.atc.StartsWith("#" + nodetext)
                    select p.atc).AsEnumerable();

        int i = 0;

        int end_of_query = query.Count();
        while (i < end_of_query)
        {
            TreeNode treeNode0 = new TreeNode(query.ElementAt(i).Substring(1, query.ElementAt(i).Length -
1));
            treeNode.ChildNodes.Add(treeNode0);
            i++;
        }
    }

    else if (nodetext.Length == 5)
    {
        var query = (from p in db.TreeView_Tables
                    where p.atc.StartsWith("*" + nodetext)
                    select p.atc).AsEnumerable();

        int i = 0;

        int end_of_query = query.Count();
        while (i < end_of_query)
        {
            TreeNode treeNode0 = new TreeNode(query.ElementAt(i).Substring(1, query.ElementAt(i).Length -
1));
            treeNode.ChildNodes.Add(treeNode0);
            i++;
        }
    }
    //return TreeView2;
}

public System.Collections.IEnumerable GetDrugByATC(string ATC_Id)
{

    var query = from p in db.ATC_DRUG_Tables
                where p.ATC_Id.StartsWith(ATC_Id)
                select new
                {
                    p.DRUG_Table.DRUG_Name,
                    p.DRUG_Table.DRUG_Supplier,
                    p.DRUG_Table.DRUG_Dosage,
                    p.DRUG_Table.DRUG_Package,
                    p.DRUG_Table.ROUTE_Table.ROUTE_Name
                }
    }
}

```

```

        };

    return query;
}

public System.Collections.IEnumerable GetAllATC()
{

    var query = from p in db.ATC_Tables
                select p;

    return query;
}

public System.Collections.IEnumerable GetAllICD()
{

    var query = from p in db.ICD_Tables
                select p;
    return query;
}

public System.Collections.IEnumerable GetAllDrugs()
{

    var query = from p in db.ATC_DRUG_Tables
                select new
                {
                    p.DRUG_Table.DRUG_Name,
                    p.DRUG_Table.DRUG_Supplier,
                    p.DRUG_Table.DRUG_Dosage,
                    p.DRUG_Table.DRUG_Package,
                    p.DRUG_Table.ROUTE_Table.ROUTE_Name,
                    p.DRUG_Table.ROUTE_Table.ROUTE_Type
                };

    return query;
}

public System.Collections.IEnumerable GetAllRoute_Name()
{

    var query = (from p in db.ROUTE_Tables
                 select p.ROUTE_Name).Distinct();

    return query;
}

public System.Collections.IEnumerable GetAllRoute_Type()
{

    var query = (from p in db.ROUTE_Tables
                 select p.ROUTE_Type).Distinct();

    return query;
}

public System.Collections.IEnumerable GetAllDrugsByRoute_Name(string ROUTE_Name)

```



```

{
    var query = from p in db.DRUG_Tables
                where p.ROUTE_Table.ROUTE_Name.Equals(ROUTE_Name)
                select new
                {
                    p.DRUG_Name,
                    p.DRUG_Supplier,
                    p.DRUG_Dosage,
                    p.DRUG_Package,
                    p.ROUTE_Table.ROUTE_Type
                };

    return query;
}

public System.Collections.IEnumerable GetAllDrugsByRoute_Type(string ROUTE_Type)
{
    var query = from p in db.DRUG_Tables
                where p.ROUTE_Table.ROUTE_Type.Equals(ROUTE_Type)
                select new
                {
                    p.DRUG_Name,
                    p.DRUG_Supplier,
                    p.DRUG_Dosage,
                    p.DRUG_Package,
                    p.ROUTE_Table.ROUTE_Name
                };

    return query;
}

public System.Collections.IEnumerable GetAllDrugOrders()
{
    var query = from p in db.DrugOrder_Tables
                where p.Quantity != 0
                select new
                {
                    p.DrugStore_Table.DRUG_Table.DRUG_Name,
                    p.DrugStore_Table.DRUG_Table.DRUG_Supplier,
                    p.DrugStore_Table.DRUG_Table.DRUG_Dosage,
                    p.DrugStore_Table.DRUG_Table.DRUG_Package,
                    p.Quantity,
                    p.Drug_Id
                };

    return query;
}

public void CommitDrugOrdersByDrug_Id(int Drug_Id)
{
    var query = (from p in db.DrugOrder_Tables
                 where p.Drug_Id == Drug_Id
                 select p).Single();

    query.DrugStore_Table.Quantity += query.Quantity;
    query.DrugStore_Table.CONTAINER_Table.Quantity -= query.Quantity;
    query.Quantity = 0;
    db.SubmitChanges();
}

```

```

    }

    public void UpdatePatient(int Patient_Id, string ID_Number, string Patient_Name, string Gender, string
    Birth_Date, string Phone_Number)
    {

        var query = (from p in db.PATIENT_Tables
                    where p.Patient_Id == Patient_Id
                    select p).Single();

        query.ID_Number = ID_Number;
        query.Patient_Name = Patient_Name;
        query.Gender = Gender;
        query.Birth_Date = Birth_Date;
        query.Phone_Number = Phone_Number;
        db.SubmitChanges();

    }

    public string CheckLogin(String Username, String Password)
    {
        var query = from p in db.LOGIN_Tables
                    where p.Username.Equals(Username) && p.Password.Equals(Password)
                    select p.Username;
        if (query.Count() == 0)
            return "invalid";
        else
            return query.Single().ToString();
    }
}
}
}

```

### Category.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="~/DoctorMasterPage.Master"
CodeBehind="Category.aspx.cs" Inherits="WebApplication1.DrugSearch.Category" %>

<%@ Register assembly="AjaxControlToolkit" namespace="AjaxControlToolkit" tagprefix="asp" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
    <title>DrugBank Category</title>
    <link href="~/Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
    <style type="text/css">

        </style>
    </asp:Content>

<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
    <span id="tabid" style="display: none">DrugSearchTab</span>
    <asp:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
    </asp:ToolkitScriptManager>
    <asp:UpdatePanel ID="UpdatePanel12" runat="server">

        <ContentTemplate>
        <table style="width:100%;">
            <tr>
                <td align="center">
                    <asp:Label ID="Label3" runat="server" CssClass="label2"
                    Text="Choose category below to display drugs (based on DrugBank categorisation)."></asp:Label>
                <br />
            </td>
        </tr>
        </table>
    </ContentTemplate>
    </asp:UpdatePanel>
</asp:Content>

```

```

        <br />
    </td>
</tr>
<tr>
    <td align="center">
        <br />
        <asp:Label ID="Label1" runat="server" CssClass="label"
            Text="Choose Drug Category : "></asp:Label>
        <asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
            CssClass="dropdownlist" Font-Bold="False"
            onselectedindexchanged="DropDownList1_SelectedIndexChanged">
        </asp:DropDownList>
        <asp:ListSearchExtender ID="DropDownList1_ListSearchExtender" runat="server"
            Enabled="True" IsSorted="true" PromptCssClass="ListSearchExtenderPrompt"
            TargetControlID="DropDownList1">
        </asp:ListSearchExtender>
        <br />
        <br />
    </td>
</tr>
<tr>
    <td align="center">
        <asp:Label ID="Label2" runat="server" CssClass="resullabel" Text="Label"
            Visible="False"></asp:Label>
    </td>
</tr>
<tr>
    <td align="center">
        <asp:GridView ID="GridView1" runat="server" BackColor="White"
            BorderColor="#999999" BorderStyle="None" BorderWidth="1px" CellPadding="3"
            GridLines="Vertical">
            <AlternatingRowStyle BackColor="#DCDCDC" />
            <FooterStyle BackColor="#CCCCCC" ForeColor="Black" />
            <HeaderStyle BackColor="#2096BD" Font-Bold="True" ForeColor="White" />
            <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />
            <RowStyle BackColor="#EEEEEE" ForeColor="Black" />
            <SelectedRowStyle BackColor="#008A8C" Font-Bold="True" ForeColor="White" />
            <SortedAscendingCellStyle BackColor="#F1F1F1" />
            <SortedAscendingHeaderStyle BackColor="#0000A9" />
            <SortedDescendingCellStyle BackColor="#CAC9C9" />
            <SortedDescendingHeaderStyle BackColor="#000065" />
        </asp:GridView>
        <br />
    </td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</div>
</asp:Content>

```

### Category.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1.DrugSearch
{
    public partial class Category : System.Web.UI.Page
    {
        Class1 ab = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {

```

```

        if (!Page.IsPostBack)
        {
            if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))
                Response.Redirect("Login.aspx");
            DropDownList1.DataSource = ab.GetAllDrugCategories();
            DropDownList1.DataBind();
            DropDownList1.Items.Insert(0, new ListItem("Choose", "Choose"));
        }
    }

    protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
    {
        GridView1.DataSource = ab.GetAllDrugsByCATEGORY_Name(DropDownList1.SelectedItem.Text);
        GridView1.DataBind();
        Label2.Text = GridView1.Rows.Count + " Results";
        Label2.Visible = true;
    }
}
}

```

### DatabaseContents.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="~/DoctorMasterPage.Master"
CodeBehind="DatabaseContents.aspx.cs" Inherits="WebApplication1.DrugSearch.DatabaseContents" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
    <title>Database Contents</title>
    <link href="../../Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
    <style type="text/css">

        </style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
    <span id="tabid" style="display: none">DrugSearchTab</span>
    <asp:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
</asp:ToolkitScriptManager>
    <asp:UpdatePanel ID="UpdatePanel12" runat="server">
        <ContentTemplate>
            <table style="width: 100%;">
                <tr>
                    <td align="center">
                        <asp:Label ID="Label3" runat="server" CssClass="label2" Text="Choose below to display all ATC
codes, ICD-10 codes or drugs."></asp:Label>
                        <br />
                        <br />
                    </td>
                    <td align="center">
                        <asp:Label ID="Label1" runat="server" CssClass="label" Text="Display all : "></asp:Label>
                        <asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
CssClass="dropdownlist"
OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged">
                            <asp:ListItem>Choose</asp:ListItem>
                            <asp:ListItem>ICD Codes</asp:ListItem>
                            <asp:ListItem>ATC Codes</asp:ListItem>
                            <asp:ListItem>Drugs</asp:ListItem>
                        </asp:DropDownList>
                    </td>
                </tr>
                <tr>
                    <td align="center">
                        <asp:Label ID="Label2" runat="server" Text="Label" CssClass="resullabel"
Visible="False"></asp:Label>
                    </td>
                </tr>
            </table>
        </ContentTemplate>
    </asp:UpdatePanel>

```

```

</tr>
<tr>
  <td align="center">
    <asp:GridView ID="GridView1" runat="server" BackColor="White" BorderColor="#999999"
      BorderStyle="None" BorderWidth="1px" CellPadding="0" GridLines="Vertical">
      <AlternatingRowStyle BackColor="#DCDCDC" />
      <FooterStyle BackColor="#CCCCCC" ForeColor="Black" />
      <HeaderStyle BackColor="#2096BD" Font-Bold="True" ForeColor="White" />
      <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />
      <RowStyle BackColor="#EEEEEE" ForeColor="Black" />
      <SelectedRowStyle BackColor="#008A8C" Font-Bold="True" ForeColor="White" />
      <SortedAscendingCellStyle BackColor="#F1F1F1" />
      <SortedAscendingHeaderStyle BackColor="#0000A9" />
      <SortedDescendingCellStyle BackColor="#CAC9C9" />
      <SortedDescendingHeaderStyle BackColor="#000065" />
    </asp:GridView>
  </td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</asp:Content>

```

### DatabaseContents.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1.DrugSearch
{
    public partial class DatabaseContents : System.Web.UI.Page
    {
        Class1 ab = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))
                Response.Redirect("Login.aspx");
        }

        protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
        {
            if (DropDownList1.SelectedItem.Text.Equals("ICD Codes"))
            {
                GridView1.DataSource = ab.GetAllICD();
                GridView1.DataBind();
            }
            else if (DropDownList1.SelectedItem.Text.Equals("ATC Codes"))
            {
                GridView1.DataSource = ab.GetAllATC();
                GridView1.DataBind();
            }

            else if (DropDownList1.SelectedItem.Text.Equals("Drugs"))
            {
                GridView1.DataSource = ab.GetAllDrugs();
                GridView1.DataBind();
            }
            Label2.Text = GridView1.Rows.Count + " Results";
            Label2.Visible = true;
        }
    }
}

```

## ICD.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="~/DoctorMasterPage.Master"
CodeBehind="ICD.aspx.cs" Inherits="WebApplication1.DrugSearch.ICD" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
  <title>Drug Search by ICD-10 code</title>
  <link href="../../Styles/StyleSheet1.css" rel="stylesheet" type="text/css" />
  <link href="../../Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
  <style type="text/css">
    .style1
    {
      height: 29px;
    }
  </style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
  <span id="tabid" style="display: none">DrugSearchTab</span>
  <asp:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
  </asp:ToolkitScriptManager>
  <br />
  <asp:UpdatePanel ID="UpdatePanel12" runat="server">
    <ContentTemplate>
      <table style="width: 100%;">
        <tr>
          <td align="center" width="50%">
            <asp:Label ID="Label3" runat="server" CssClass="label2"
              Text="Search drugs by ICD code : "></asp:Label>
            <br />
            <br />
          </td>
        </tr>
        <tr>
          <td align="center" width="50%">
            <asp:TextBox ID="TextBox1" runat="server" autocomplete="off" Height="39px"
              TextMode="MultiLine" Width="451px"></asp:TextBox>
            <asp:AutoCompleteExtender ID="TextBox1_AutoCompleteExtender" runat="server"
              BehaviorID="AutoCompleteEx" CompletionInterval="1000"
              CompletionListCssClass="autocomplete_completionListElement"
              CompletionListHighlightedItemCssClass="autocomplete_highlightedListItem"
              CompletionListItemCssClass="autocomplete_listItem" EnableCaching="true"
              MinimumPrefixLength="1" ServiceMethod="GetICDList"
              ServicePath="../../WebService1.asmx" TargetControlID="TextBox1">
              <Animations>
                <OnShow>
                  <Sequence>
                    <%-- Make the completion list transparent and then show it --%>
                    <OpacityAction Opacity="100" />
                    <HideAction Visible="true" />

                    <%-- Cache the original size of the completion list the first time
                    the animation is played and then set it to zero --%>
                    <ScriptAction Script="
                      // Cache the size and setup the initial size
                      var behavior = $find('AutoCompleteEx');
                      if (!behavior._height) {
                        var target = behavior.get_completionList();
                        behavior._height = target.offsetHeight - 2;
                        target.style.height = '0px';
                      }" />

                    <%-- Expand from 0px to the appropriate size while fading in --%>
                    <Parallel Duration=".4">
                      <FadeIn />
                      <Length PropertyKey="height" StartValue="0"
                        EndValueScript="$find('AutoCompleteEx')._height" />
                  </Sequence>
                </OnShow>
              </Animations>
            </asp:AutoCompleteExtender>
          </td>
        </tr>
      </table>
    </ContentTemplate>
  </asp:UpdatePanel>
</asp:Content>
```

```

        </Parallel>
    </Sequence>
</OnShow>
<OnHide>
    <!-- Collapse down to 0px and fade out --%>
    <Parallel Duration=".4">
        <FadeOut />
        <Length PropertyKey="height" StartValueScript="$find('AutoCompleteEx')._height"
EndValue="0" />
    </Parallel>
</OnHide>
</Animations>
</asp:AutoCompleteExtender>
</td>
</tr>
<tr>
    <td align="center" class="style1">
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Search"
CssClass="button" />
    </td>
</tr>
<tr>
    <td align="center">
        <asp:Label ID="Label2" runat="server" Text="Label" CssClass="resullabel"
Visible="False"></asp:Label>
    </td>
</tr>
<tr>
    <td align="center">
        <asp:GridView ID="GridView1" runat="server" BackColor="White" BorderColor="#999999"
        BorderStyle="None" BorderWidth="1px" CellPadding="3" GridLines="Vertical">
            <AlternatingRowStyle BackColor="#DCDCDC" />
            <FooterStyle BackColor="#CCCCCC" ForeColor="Black" />
            <HeaderStyle BackColor="#2096BD" Font-Bold="True" ForeColor="White" />
            <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />
            <RowStyle BackColor="#EEEEEE" ForeColor="Black" />
            <SelectedRowStyle BackColor="#008A8C" Font-Bold="True" ForeColor="White" />
            <SortedAscendingCellStyle BackColor="#F1F1F1" />
            <SortedAscendingHeaderStyle BackColor="#0000A9" />
            <SortedDescendingCellStyle BackColor="#CAC9C9" />
            <SortedDescendingHeaderStyle BackColor="#000065" />
        </asp:GridView>
        <br />
    </td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</asp:Content>

```

### ICD.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1.DrugSearch
{
    public partial class ICD : System.Web.UI.Page
    {
        Class1 ab = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))

```

```

        Response.Redirect("Login.aspx");
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        if (TextBox1.Text.Length > 9)
        {
            GridView1.DataSource = ab.SearchAllDrugsByICD_Id(TextBox1.Text.Substring(0, 10));
            GridView1.DataBind();
            Label2.Text = GridView1.Rows.Count + " Results";
            Label2.Visible = true;
        }
    }
}
}
}

```

### Ingredient.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="~/DoctorMasterPage.Master"
CodeBehind="Ingredient.aspx.cs" Inherits="WebApplication1.DrugSearch.Ingredient" %>

```

```

<%@ Register assembly="AjaxControlToolkit" namespace="AjaxControlToolkit" tagprefix="asp" %>

```

```

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
    <title>Drug search by ATC ingredient</title>
    <link href="../../Styles/StyleSheet1.css" rel="stylesheet" type="text/css" />
    <link href="../../Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
    <style type="text/css">

    </style>
</asp:Content>

<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
    <span id="tabid" style="display: none">DrugSearchTab</span>
    <asp:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
    </asp:ToolkitScriptManager>
    <br />
    <asp:UpdatePanel ID="UpdatePanel12" runat="server">

    <ContentTemplate>
    <table width="100%" >
    <tr >
    <td align="center">
    <asp:Label ID="Label1" runat="server" CssClass="label2"
        Text="Search drugs by ATC ingredient : "></asp:Label>
    <br />
    <br />
    </td>
    </tr>
    <tr>
    <td align="center">
    <asp:TextBox ID="TextBox3" runat="server" autocomplete="off" Height="39px"
        TextMode="MultiLine" Width="277px"></asp:TextBox>
    <asp:AutoCompleteExtender ID="TextBox3_AutoCompleteExtender" runat="server"
        BehaviorID="AutoCompleteEx2" CompletionInterval="1000"
        CompletionListCssClass="autocomplete_completionListElement"
        CompletionListHighlightedItemCssClass="autocomplete_highlightedListItem"
        CompletionListItemCssClass="autocomplete_listItem" EnableCaching="true"
        Enabled="True" MinimumPrefixLength="1" ServiceMethod="GetIngredientList"
        ServicePath="../../WebService1.asmx" TargetControlID="TextBox3">
    <Animations>
    <OnShow>
    <Sequence>

```



```

<!-- Make the completion list transparent and then show it -->
<OpacityAction Opacity="0" />
<HideAction Visible="true" />

<!-- Cache the original size of the completion list the first time
the animation is played and then set it to zero -->
<ScriptAction Script="
// Cache the size and setup the initial size
var behavior = $find('AutoCompleteEx2');
if (!behavior._height) {
    var target = behavior.get_completionList();
    behavior._height = target.offsetHeight - 2;
    target.style.height = '0px';
}" />

<!-- Expand from 0px to the appropriate size while fading in -->
<Parallel Duration=".4">
    <FadeIn />
    <Length PropertyKey="height" StartValue="0"
EndValueScript="$find('AutoCompleteEx2')._height" />
</Parallel>
</Sequence>
</OnShow>
<OnHide>
<!-- Collapse down to 0px and fade out -->
<Parallel Duration=".4">
    <FadeOut />
    <Length PropertyKey="height" StartValueScript="$find('AutoCompleteEx2')._height"
EndValue="0" />
</Parallel>
</OnHide>
</Animations>
</asp:AutoCompleteExtender>
</td>
</tr>
<tr>
<td align="center" >
    <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Search"
    CssClass="button" />
</td>
</tr>
<tr>
<td align="center">
    <asp:Label ID="Label2" runat="server" Text="Label" CssClass="resullabel"
    Visible="False"></asp:Label>
</td>
</tr>
<tr>
<td align="center">
    <asp:GridView ID="GridView1" runat="server" BackColor="White"
    BorderColor="#999999" BorderStyle="None" BorderWidth="1px" CellPadding="3"
    GridLines="Vertical">
    <AlternatingRowStyle BackColor="#DCDCDC" />
    <FooterStyle BackColor="#CCCCCC" ForeColor="Black" />
    <HeaderStyle BackColor="#2096BD" Font-Bold="True" ForeColor="White" />
    <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />
    <RowStyle BackColor="#EEEEEE" ForeColor="Black" />
    <SelectedRowStyle BackColor="#008A8C" Font-Bold="True" ForeColor="White" />
    <SortedAscendingCellStyle BackColor="#F1F1F1" />
    <SortedAscendingHeaderStyle BackColor="#0000A9" />
    <SortedDescendingCellStyle BackColor="#CAC9C9" />
    <SortedDescendingHeaderStyle BackColor="#000065" />
    </asp:GridView>
    <br />
</td>
</tr>
</table>

```

```

        </ContentTemplate>
    </asp:UpdatePanel>
    <br />
</asp:Content>

```

### Ingredient.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1.DrugSearch
{
    public partial class Ingredient : System.Web.UI.Page
    {
        Class1 ab = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))
                Response.Redirect("Login.aspx");
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            GridView1.DataSource = ab.SearchAllDrugsByIngredient(TextBox3.Text);
            GridView1.DataBind();
            Label2.Text = GridView1.Rows.Count + " Results";
            Label2.Visible = true;
        }
    }
}

```

### RouteName.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="~/DoctorMasterPage.Master"
    CodeBehind="RouteName.aspx.cs" Inherits="WebApplication1.DrugSearch.RouteName" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
    <title>Drug search by package type</title>
    <link href="../../Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
    <style type="text/css">

        </style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
    <span id="tabid" style="display: none">DrugSearchTab</span>
    <asp:ToolkitScriptManager ID="Toolkitscriptmanager1" runat="server">
    </asp:ToolkitScriptManager>
    <asp:UpdatePanel ID="UpdatePanel12" runat="server">
        <ContentTemplate>
            <table style="width: 100%;">
                <tr>
                    <td align="center">
                        <asp:Label ID="Label1" runat="server" Text="Search drugs by package type : "
                        CssClass="label2"></asp:Label>
                        <br />
                        <br />
                    </td>
                </tr>
                <tr>
                    <td align="center">

```

```

        <br />
        <asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
        CssClass="dropdownlist"
        OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged">
        </asp:DropDownList>
        <asp:ListSearchExtender ID="DropDownList1_ListSearchExtender" runat="server"
        Enabled="True"
        IsSorted="true" PromptCssClass="ListSearchExtenderPrompt"
        TargetControlID="DropDownList1">
        </asp:ListSearchExtender>
    </td>
</tr>
<tr>
    <td align="center">
        <asp:Label ID="Label2" runat="server" Text="Label" CssClass="resullabel"
        Visible="False"></asp:Label>
    </td>
</tr>
<tr>
    <td align="center">
        <asp:GridView ID="GridView1" runat="server" BackColor="White" BorderColor="#999999"
        BorderStyle="None" BorderWidth="1px" CellPadding="3" GridLines="Vertical">
        <AlternatingRowStyle BackColor="#DCDCDC" />
        <FooterStyle BackColor="#CCCCCC" ForeColor="Black" />
        <HeaderStyle BackColor="#2096BD" Font-Bold="True" ForeColor="White" />
        <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />
        <RowStyle BackColor="#EEEEEE" ForeColor="Black" />
        <SelectedRowStyle BackColor="#008A8C" Font-Bold="True" ForeColor="White" />
        <SortedAscendingCellStyle BackColor="#F1F1F1" />
        <SortedAscendingHeaderStyle BackColor="#0000A9" />
        <SortedDescendingCellStyle BackColor="#CAC9C9" />
        <SortedDescendingHeaderStyle BackColor="#000065" />
        </asp:GridView>
        <br />
        <br />
    </td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</asp:Content>

```

### RouteName.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1.DrugSearch
{
    public partial class RouteName : System.Web.UI.Page
    {
        Class1 ab = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {
                if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))
                    Response.Redirect("Login.aspx");
                DropDownList1.DataSource = ab.GetAllRoute_Name();
                DropDownList1.DataBind();
                DropDownList1.Items.Insert(0, new ListItem("Choose", "Choose"));
            }
        }
    }
}

```

```

    }

    protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
    {
        GridView1.DataSource = ab.GetAllDrugsByRoute_Name(DropDownList1.SelectedItem.Text);
        GridView1.DataBind();
        Label2.Text = GridView1.Rows.Count + " Results";
        Label2.Visible = true;
    }
}
}
}

```

### RouteType.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="~/DoctorMasterPage.Master"
    CodeBehind="RouteType.aspx.cs" Inherits="WebApplication1.DrugSearch.RouteType" %>

<%@ Register Assembly="AjaxControlToolkit" Namespace="AjaxControlToolkit" TagPrefix="asp" %>
<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
    <title>Drug search by administration route</title>
    <link href="../../Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
    <style type="text/css">

        </style>
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
    <asp:ToolkitScriptManager ID="Toolkitscriptmanager1" runat="server">
</asp:ToolkitScriptManager>
    <span id="tabid" style="display: none">DrugSearchTab</span>
    <asp:UpdatePanel ID="UpdatePanel12" runat="server">
        <ContentTemplate>
            <table style="width: 100%;">
                <tr>
                    <td align="center">
                        <asp:Label ID="Label1" runat="server" Text="Search drugs by administration route : ">
                            <asp:Label ID="Label2" runat="server" Text="Label" Visible="False">
                                </asp:Label>
                            <br />
                            <br />
                        </td>
                    <td align="center">
                        <br />
                        <asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
                            CssClass="dropdownlist"
                            OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged">
                        </asp:DropDownList>
                        <asp:ListSearchExtender ID="DropDownList1_ListSearchExtender" runat="server"
                            Enabled="True"
                            IsSorted="true" PromptCssClass="ListSearchExtenderPrompt"
                            TargetControlID="DropDownList1">
                        </asp:ListSearchExtender>
                    </td>
                </tr>
                <tr>
                    <td align="center">
                        <asp:Label ID="Label2" runat="server" Text="Label" Visible="False">
                            </asp:Label>
                        </td>
                    <td align="center">
                        <asp:GridView ID="GridView1" runat="server" BackColor="White" BorderColor="#999999"
                            BorderStyle="None" BorderWidth="1px" CellPadding="3" GridLines="Vertical">
                            <AlternatingRowStyle BackColor="#DCDCDC" />
                            <FooterStyle BackColor="#CCCCCC" ForeColor="Black" />
                        </asp:GridView>
                    </td>
                </tr>
            </table>
        </ContentTemplate>
    </asp:UpdatePanel>

```

```

        <HeaderStyle BackColor="#2096BD" Font-Bold="True" ForeColor="White" />
        <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />
        <RowStyle BackColor="#EEEEEE" ForeColor="Black" />
        <SelectedRowStyle BackColor="#008A8C" Font-Bold="True" ForeColor="White" />
        <SortedAscendingCellStyle BackColor="#F1F1F1" />
        <SortedAscendingHeaderStyle BackColor="#0000A9" />
        <SortedDescendingCellStyle BackColor="#CAC9C9" />
        <SortedDescendingHeaderStyle BackColor="#000065" />
    </asp:GridView>
</td>
</tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</asp:Content>

```

### RouteType.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1.DrugSearch
{
    public partial class RouteType : System.Web.UI.Page
    {
        Class1 ab = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {
                if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))
                    Response.Redirect("Login.aspx");
                DropDownList1.DataSource = ab.GetAllRoute_Type();
                DropDownList1.DataBind();
                DropDownList1.Items.Insert(0, new ListItem("Choose", "Choose"));
            }
        }

        protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
        {
            GridView1.DataSource = ab.GetAllDrugsByRoute_Type(DropDownList1.SelectedItem.Text);
            GridView1.DataBind();
            Label2.Text = GridView1.Rows.Count + " Results";
            Label2.Visible = true;
        }
    }
}

```

### TreeView.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" MasterPageFile="~/DoctorMasterPage.Master"
CodeBehind="TreeView.aspx.cs" Inherits="WebApplication1.DrugSearch.TreeView" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="head">
    <title>Drug Treeview by ATC code</title>
    <link href="../../Styles/StyleSheet2.css" rel="stylesheet" type="text/css" />
    <style type="text/css">

        </style>
</asp:Content>

```

```

<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="ContentPlaceHolder1">
  <span id="tabid" style="display: none">DrugSearchTab</span>
  <asp:ScriptManager ID="ScriptManager1" runat="server">
  </asp:ScriptManager>
  <asp:UpdatePanel ID="UpdatePanel1" runat="server">

  <ContentTemplate>
  <table style="width:100%;">
  <tr>
  <td align="center">

    <asp:Label ID="Label2" runat="server" CssClass="label2"
      Text="The 5-level ATC treeview is shown below. "></asp:Label>
    <br />
    <asp:Label ID="Label3" runat="server" CssClass="label2"
      Text="Select one tree node and press the Search Drug button to"></asp:Label>
    <br />
    <asp:Label ID="Label4" runat="server"
      Text="display all drugs with the selected tree node ATC code."
      CssClass="label2"></asp:Label>
    <br />
    <br />

  </td>
  </tr>
  <tr>
  <td>
    <asp:TreeView ID="TreeView2" runat="server" ExpandDepth="0" Font-Italic="False"
      Font-Names="Arial" ForeColor="Red" ImageSet="Simple" NodeIndent="10"
      onselectednodechanged="TreeView2_SelectedNodeChanged">
      <HoverNodeStyle Font-Underline="True" ForeColor="#DD5555" />
      <NodeStyle Font-Names="Verdana" Font-Size="8pt" ForeColor="#5353FF"
        HorizontalPadding="5px" NodeSpacing="0px" VerticalPadding="0px" />
      <ParentNodeStyle Font-Names="Arial" ForeColor="#5377A9" />
      <SelectedNodeStyle Font-Bold="True" Font-Underline="True" ForeColor="#DD5555"
        HorizontalPadding="0px" VerticalPadding="0px" />
    </asp:TreeView>
  </td>
  </tr>
  <tr>
  <td align="center">
    <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
      Text="Search Drug" Visible="False" CssClass="button" />
  </td>
  </tr>
  <tr>
  <td align="center">
    <asp:Label ID="Label1" runat="server" CssClass="resullabel" Text="Label"
      Visible="False"></asp:Label>
  </td>
  </tr>
  <tr>
  <td align="center">
    <asp:GridView ID="GridView1" runat="server" CellPadding="3"
      GridLines="Vertical" BackColor="White" BorderColor="#0000CC"
      BorderStyle="None" BorderWidth="1px">
      <AlternatingRowStyle BackColor="#DCDCDC" />
      <FooterStyle BackColor="#CCCCCC" ForeColor="Black" />
      <HeaderStyle BackColor="#2096BD" Font-Bold="True" ForeColor="White" />
      <PagerStyle BackColor="#999999" ForeColor="Black" HorizontalAlign="Center" />
      <RowStyle BackColor="#EEEEEE" ForeColor="Black" />
      <SelectedRowStyle BackColor="#008A8C" Font-Bold="True" ForeColor="White" />
      <SortedAscendingCellStyle BackColor="#F1F1F1" />
      <SortedAscendingHeaderStyle BackColor="#0000A9" />
      <SortedDescendingCellStyle BackColor="#CAC9C9" />
      <SortedDescendingHeaderStyle BackColor="#000065" />
    </asp:GridView>
  </td>
  </tr>
  </table>
  </ContentTemplate>
  </asp:UpdatePanel>
</asp:Content>

```

```

        </td>
    </tr>
</table>
</ContentTemplate>
</asp:UpdatePanel>
</asp:Content>

```

### TreeView.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1.DrugSearch
{
    public partial class TreeView : System.Web.UI.Page
    {
        string[] split = { " " };
        Class1 ab = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {
                if (Session["Login"] == null || !(Session["Login"].Equals("doctor")))
                    Response.Redirect("Login.aspx");
                TreeView2 = ab.j0(TreeView2);
            }
        }

        protected void TreeView2_SelectedNodeChanged(object sender, EventArgs e)
        {
            Button1.Visible = true;
            string[] ee = TreeView2.SelectedNode.Split(split, StringSplitOptions.RemoveEmptyEntries);
            if (TreeView2.SelectedNode.ChildNodes.Count==0)
            {
                ab.j1(ee[0], TreeView2.SelectedNode);
                TreeView2.SelectedNode.Expand();
            }
            GridView1.Visible = false;
            Label1.Visible = false;
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string[] ee = TreeView2.SelectedNode.Split(split, StringSplitOptions.RemoveEmptyEntries);
            GridView1.DataSource = ab.GetDrugByATC(ee[0]);
            GridView1.DataBind();
            Label1.Text=GridView1.Rows.Count + " Results";
            Label1.Visible = true;
            GridView1.Visible = true;
        }
    }
}

```