

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Πληροφορική»

**Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Ανάπτυξη διαδικτυακής εφαρμογής .NET για Ηλεκτρονικούς Ιατρικούς Φακέλους</b>  <b>.NET web application for Electronic Health Records</b>
Όνοματεπώνυμο Φοιτητή	<b>Γεώργιος Κατσής</b>
Πατρώνυμο	<b>Δημήτρης</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ/ 13035</b>
Επιβλέπων	<b>Ευθύμιος Αλέπης, Επίκουρος Καθηγητής</b>

Ημερομηνία Παράδοσης **05/2016**

---

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

Ευθύμιος Αλέπης  
Επίκουρος Καθηγητής

(υπογραφή)

Μαρία Βίββου  
Καθηγητής

(υπογραφή)

Γεώργιος Τσιχριντζής  
Καθηγητής

## Περίληψη

Το θέμα της διπλωματικής εργασίας είναι η δημιουργία ενός ηλεκτρονικού ιατρικού φακέλου ασθενούς, όπου θα καταγράφονται οι αιματολογικές του εξετάσεις από τον χρήστη/γιατρό.

Στόχος της εργασίας αυτής είναι η δημιουργία μιας εύχρηστης βάσης δεδομένων, η οποία θα χρησιμοποιείται για την αποθήκευση των αναγκαίων στοιχείων των ασθενών ενός ιατρείου ή ακόμη και ενός νοσοκομείου, ώστε να μπορεί ο εκάστοτε χρήστης /ιατρός να τη χρησιμοποιεί είτε αποκτώντας πρόσβαση είτε στο πρόσφατο ιστορικό ασθένειάς τους, έτσι ώστε να ενημερώνεται πλήρως και ταυτόχρονα να μπορεί να εξαγάγει κάποια στατιστικά συμπεράσματα, είτε προσθέτοντας νέες ιατρικές εξετάσεις και ενημερώνοντας το φάκελο. Επίσης η βάση αυτή θα εξυπηρετεί και τον ασθενή, αφού θα του δίνει τη δυνατότητα να έχει πλήρη πρόσβαση και άρα συνεχή επίγνωση του ιστορικού του ιατρικού του φακέλου.

## Abstract

The subject of this thesis is the creation of an electronic medical file for patients, used by the user/doctor in order to register the patients' blood tests.

The purpose of this thesis is the creation of a convenient (easy to use) data base, which can be used for the storage of necessary data of patients of a medical office or even a hospital, so that the respective doctor/user can use it either to have access to the recent medical history of the patient and so be adequately informed and extract statistical conclusions, or to add data of new medical examinations and update the file. Furthermore, this data base will respectively serve the patient, as it will enable him to have full access and so be constantly aware of his own medical file' s history.

## Περιεχόμενα

1. Εισαγωγή.....	4
2. Έννοια Ιατρικού Φακέλου.....	6
3. Ηλεκτρονικός Ιατρικός Φάκελος.....	8
4. Παρόμοιες εφαρμογές στην Ελλάδα.....	15
5. Γενική εξέταση αίματος.....	18
6. Εγχειρίδιο χρήσης.....	24
7. Διάγραμμα ροής.....	31
8. .NET FRAMEWORK.....	33
9. Αντικειμενοστραφής Προγραμματισμός.....	38
10. Η γλώσσα προγραμματισμού C#.....	41
11. ASP.NET.....	43
12. Το πρότυπο MVC.....	45
13. Βάση δεδομένων σε περιβάλλον MySQL.....	49
14. APACHE SERVER.....	54
15. Το Visual Studio.....	56
16. Συμπεράσματα και μελλοντικές επεκτάσεις.....	58
17. Βιβλιογραφία.....	59

# 1. Εισαγωγή

Η αλματώδης πρόοδος της πληροφορικής και των τηλεπικοινωνιών αφήνει το στίγμα της σε πληθώρα τομέων της επιστήμης, ανάμεσα στους οποίους συγκαταλέγεται και η ιατρική. Καθημερινά, εισβάλλουν στη ζωή μας πολλές τεχνολογικές καινοτομίες, οι οποίες σε ένα μεγάλο ποσοστό αφορούν στην ιατρική με τη μορφή νέων θεραπευτικών προσεγγίσεων ασθενειών και την εξεύρεση σύγχρονων μεθόδων έγκαιρης διάγνωσης. Σε αυτές τις καινοτομίες συγκαταλέγονται τα ιατρικά πληροφοριακά συστήματα, βοηθώντας στην αποτελεσματική και λειτουργική αρχειοθέτηση των ιατρικού ιστορικού των ασθενών, με απώτερο σκοπό τη δημιουργία ενός συστήματος υγείας, στο οποίο ο πολίτης θα είναι ο επωφελούμενος.

Η Ευρωπαϊκή Ένωση προωθεί τη δημιουργία ενός συστήματος ηλεκτρονικών μητρώων υγείας με τη στήριξη της ανταλλαγής πληροφοριών και της τυποποίησης, καθώς και την ανάπτυξη δικτύων ανταλλαγής πληροφοριών μεταξύ φορέων περίθαλψης, ώστε να υπάρχει συντονισμός των δράσεων σε περίπτωση κινδύνου για τη δημόσια υγεία. Επιπρόσθετος στόχος είναι η παροχή υπηρεσιών υγείας σε απευθείας σύνδεση (online). Υποστηρίζονται υπηρεσίες που αφορούν σε: πληροφορίες υγιεινής ζωής, πρόληψη ασθενειών, ανάπτυξη συστημάτων τηλε-συμβουλευτικής, ηλεκτρονικής συνταγογράφησης, παραπομπής και επιστροφής των ιατρικών εξόδων.

Συνοπτικά, ο όρος “ηλεκτρονική υγεία” καλύπτει ένα ευρύ φάσμα εργαλείων βασισμένων στις τεχνολογίες πληροφοριών και επικοινωνιών που στοχεύουν στην καλύτερη πρόληψη, διάγνωση, θεραπεία, παρακολούθηση και διαχείριση της υγείας και του τρόπου ζωής. Δηλαδή, τη δημιουργία ηλεκτρονικού φακέλου του ασθενή, μια εξελισσόμενη ιδέα με πρωταρχικό σκοπό τη μακροπρόθεσμη συλλογή πληροφοριών υγείας για τους ασθενείς.

Το δικαίωμα του ασθενούς για διασφάλιση της εμπιστευτικότητας των προσωπικών του δεδομένων δεν αποδυναμώνεται, εξαιτίας της χρήσης του ηλεκτρονικού φακέλου υγείας. Αντίθετα, ο καθορισμός ηθικών και νομικών διαδικασιών και κριτηρίων, όσον αφορά στην ηλεκτρονική καταχώρηση, επεξεργασία και διακίνηση των ευαίσθητων προσωπικών δεδομένων των ασθενών σε πιθανούς χρήστες ή διαχειριστές δεδομένων υγείας καθίσταται απαραίτητος.

Τα τελευταία χρόνια, ο όγκος των πληροφοριών που σχετίζονται με την ιατρική περίθαλψη του ασθενή έχει αυξηθεί σημαντικά. Αυτό οφείλεται στην ενσωμάτωση μεγάλου αριθμού εργαστηριακών και παραϊατρικών εξετάσεων στα αρχεία των ασθενών. Επιπρόσθετα, τα καθήκοντα των ιατρών και νοσηλευτών αυξάνονται συνεχώς και η ανάγκη για διασφάλιση των ιατρικών αρχείων ασθενών γίνεται περισσότερο επιτακτική.

Οι κλασικοί ιατρικοί φάκελοι ασθενών που βασίζονται στην καταγραφή των δεδομένων του ασθενή σε χαρτί αδυνατούν να «συγκρατούν» το μεγάλο όγκο πληροφοριών με αποτέλεσμα το αρχείο να γίνεται ογκώδες, να χάνονται δεδομένα και γενικά να μην υπάρχει χρονική συσχέτιση των διαφόρων εξετάσεων με το πλήρες ιατρικό ιστορικό του ασθενή. Έτσι, προκύπτει έντονα η ανάγκη της δημιουργίας ενός ηλεκτρονικού φακέλου υγείας ασθενή, όπου θα περιλαμβάνει συγκεντρωμένες όλες τις

πληροφορίες της ιατρικής αναφοράς με αποτελεσματικό τρόπο και θα εκμεταλλεύεται τις δυνατότητες των νέων τεχνολογιών με σκοπό την παροχή ποιοτικότερης ιατροφαρμακευτικής περίθαλψης στον ασθενή.

Σκοπός της εφαρμογής είναι να διευκολύνει την πρόσβαση των χρηστών στην βάση δεδομένων, καθώς και την επεξεργασία της, μέσω μιας κατάλληλης και φιλικής διεπαφής όπου θα είναι προσιτή από κάθε ηλικία ή μορφωτικό επίπεδο. Επιπλέον η δυνατότητα εξαγωγής στατιστικών και η μελέτη τους από τον εκάστοτε γιατρό που χρησιμοποιεί την εφαρμογή, θα μπορούσε να οδηγήσει στην καλύτερη πρόληψη και καταπολέμηση ασθενειών που είναι επικίνδυνες για τον άνθρωπο.

Για την υλοποίηση της παραπάνω εφαρμογής χρησιμοποιείται το περιβάλλον του Visual Studio 2013 και η γλώσσα προγραμματισμού C# σε συνδυασμό με την MySQL για την δημιουργία της κατάλληλης βάσης.

## 2. Έννοια Ιατρικού Φακέλου

### Δεδομένα Υγείας

Τα δεδομένα υγείας είναι τα δεδομένα τα οποία συνδέονται με τη διάγνωση και τι άλλες διαδικασίες που ακολουθούνται κατά τη διάρκεια της περίθαλψης ενός ασθενή και γενικότερα είναι τα δεδομένα τα οποία αφορούν τη φυσική και ψυχική υγεία ενός ασθενή τόσο στο παρελθόν όσο στο παρόν και στο μέλλον. Ειδικότερα για ένα ασθενή, αυτά είναι τα προσωπικά στοιχεία, τα εργαστηριακά αποτελέσματα, οι ιατρικές εντολές, τα δεδομένα υγειονομικής φροντίδας, τα αρχεία εικόνων και τα διαχειριστικά δεδομένα. Τα στοιχεία του ιατρικού φακέλου εξαρτώνται άμεσα από την ειδικότητα του Ιατρού.

Ένας ιατρός κατά τη διαδικασία της εξέτασης και ακολούθως της διάγνωσης και περίθαλψης μιας ασθένειας, ανατρέχει σε πληροφορίες που βρίσκονται στον ιατρικό φάκελο ενός συγκεκριμένου ασθενή που μπορεί να είναι το προηγούμενο ιστορικό, διάφορες αλλεργίες, εργαστηριακές εξετάσεις κτλ. Επομένως, η αποδοτική διαχείριση των πληροφοριών που χαρακτηρίζουν ένα ασθενή, μπορούν να υποστηρίξουν και να βελτιώσουν την ποιότητα των υγειονομικών υπηρεσιών που προσφέρονται στον ασθενή.

Οι βασικές κατηγορίες δεδομένων υγείας , είναι οι πιο κάτω:

- **Δημογραφικά δεδομένα:** Περιλαμβάνουν τα δεδομένα τα οποία δεν είναι ιατρικής φύσης αλλά δεδομένα για ταυτοποίηση του ασθενή, έτσι ώστε να διακρίνεται ο ένας ασθενής από τον άλλον.
- **Κοινωνικοοικονομικά δεδομένα:** Περιλαμβάνουν τα δεδομένα τα οποία αποτελούν ένα υποσύνολο των προσωπικών δεδομένων και δίνουν ενδείξεις στους παροχείς ιατρικής περίθαλψης για τυχόν προβλήματα που θα συναντήσουν σχετικά με την υγεία του ασθενή καθώς και ενδείξεις για το σχεδιασμό της υγειονομικής φροντίδας του .
- **Οικονομικά δεδομένα:** Περιλαμβάνουν τα δεδομένα τα οποία αφορούν πληροφορίες τιμολόγησης των υπηρεσιών και τα δεδομένα αυτά μπορεί να σχετίζονται με ασφαλιστικές εταιρείες.
- **Κλινικά δεδομένα:** Τα δεδομένα αυτά προσδιορίζουν τη διάγνωση και τη θεραπεία του ασθενή. Δίνουν ένα πλήρες ιστορικό της πορείας της υγείας του ασθενή δια μέσου του χρόνου και παρουσιάζουν:

ο Κύριες ασθένειες

ο Θερμοκρασία και αιματική πίεση του ασθενή.

ο Εργαστηριακές εξετάσεις.

ο Ακτινολογικές εξετάσεις και άλλες απεικονιστικές εξετάσεις.

- ο Φαρμακευτικές αγωγές.
- ο Διαδικασίες εγχειρήσεων.

Πρόκειται για την συστηματοποιημένη συλλογή του ιστορικού και της κατάστασης υγείας ενός ασθενούς, ο οποίος δημιουργείται, διατηρείται και συντηρείται από έναν ιατρό ή μία μονάδα υγείας ή άλλον επαγγελματία φροντίδας υγείας. Σύμφωνα με την Ευρωπαϊκή Επιτροπή Προτυποποίησης, Ιατρικός Φάκελος είναι η αποθήκη όλων των πληροφοριών που αφορούν στο ιατρικό ιστορικό του ασθενούς, έτσι ώστε να αποτελεί τη βάση της διάγνωσης και της θεραπευτικής αντιμετώπισης του ασθενούς αλλά και τη βάση επιδημιολογικών ερευνών. Επιπλέον, παρέχει πληροφορίες διοικητικής, οικονομικής και στατιστικής φύσεως, καθώς και ποιοτικού ελέγχου. Οι ιατρικοί φακέλοι ταξινομούνται σε σχέση με:

**Το περιεχόμενο:** Φάκελος ενδο-νοσοκομειακών ασθενών, Φάκελος εξω-νοσοκομειακών ασθενών, Φάκελος Φροντίδας Υγείας.

**Τη δομή:** Φάκελος προσανατολισμένος στο πρόβλημα, Φάκελος προσανατολισμένος στο χρόνο, Φάκελος προσανατολισμένος στην εργασία, Φάκελος προσανατολισμένος στην αντιμετώπιση του ασθενή.

**Το σκοπό:** Νοσηλευτικός φάκελος, Ακτινολογικός φάκελος, Φαρμακευτικός φάκελος.

**Το μέσο που χρησιμοποιείται για την καταγραφή:** Χειρόγραφος φάκελος, Ηλεκτρονικός φάκελος, Φάκελος Πολυμέσων, Φάκελος ασθενή σε μικροφίλμ.

Πάντως ανεξάρτητα από την μορφή που έχει, κάθε ιατρικός φάκελος θα πρέπει να περιέχει όλα τα δεδομένα – πληροφορία που σχετίζεται με την κατάσταση υγείας του ασθενή. Η πληροφορία αυτή αναλυτικότερα αφορά το ιστορικό, τη κλινική εξέταση, τη διάγνωση, τα αποτελέσματα εργαστηριακών – παρακλινικών εξετάσεων, τις απεικονιστικές εξετάσεις, δηλαδή ακτινογραφίες, αξονικές τομογραφίες, μμαγνητικές, υπέρηχοι, τα ηλεκτροκαρδιογραφήματα και τις ενδοσκοπικές εξετάσεις, δηλαδή γαστροσκόπηση, κολonosκόπηση κτλ.

Συνήθως οι αντίστοιχες εξετάσεις συνοδεύουν τον φάκελο του ασθενούς υπό την μορφή με την οποία δημιουργούνται στα αντίστοιχα εργαστήρια, δηλαδή προτυπωμένα έντυπα για μικροβιολογικές – βιοχημικές εξετάσεις, ακτινογραφικά φιλμ, χαρτιά ηλεκτροκαρδιογραφήματων, συνοδευόμενα με χειρόγραφα δυσανάγνωστα ιστορικά με σύνθετες, αποδιοργανωμένες σημειώσεις και περιγραφές ελεύθερων κειμένων που περιλαμβάνουν συνώνυμα ή συντομεύσεις, που ανατρέπουν την σωστή οργάνωση. Αποτέλεσμα των μορφών αυτών είναι η παραγωγή ενός μεγάλου όγκου ιατρικού φακέλου, με μεγάλη πιθανότητα απώλειας δεδομένων, με μεγάλη δυσκολία ανάκτησης πληροφορίας, με ασύγχρονο συσχετισμό του ιστορικού με τις εξετάσεις και την κλινική εξέταση.



### 3. Ηλεκτρονικός Ιατρικός Φάκελος

Ο ορισμός της έννοιας αυτής όπως προκύπτει από το Ινστιτούτο Ιατρικής των Η.Π.Α. αναφέρει ότι ο ηλεκτρονικός ιατρικός φάκελος "είναι ένα σύστημα σχεδιασμένο με τέτοιο τρόπο ώστε να υποστηρίζει την απόλυτη διαθεσιμότητα και την ακρίβεια ιατρικών ή άλλων πληροφοριών με σκοπό τη παροχή ιατρικής περίθαλψης". Με αλλά λόγια περιλαμβάνει έγγραφα, αποθηκευμένα σε ηλεκτρονική μορφή, που αφορούν την κατάσταση του ασθενούς, όπως παραπεμπτικά και αποτελέσματα εξετάσεων, καρδιογραφήματα, υπερήχους, ακτινογραφίες, στοιχεία νοσηλείας κ.ο.κ. που είναι άμεσα διαθέσιμα στο ιατρικό και παραϊατρικό προσωπικό μέσω ενός ηλεκτρονικού συστήματος. Αποτελεί έτσι ένα μέσο επικοινωνίας μεταξύ του ιατρικού και παραϊατρικού προσωπικού που ασχολείται με την ιατρική περίθαλψη του ασθενή και τους δίνει την δυνατότητα να γνωρίζει το ιατρικό ιστορικό του ασθενούς με σκοπό την διευκόλυνση της διαδικασίας της διάγνωσης και την αποτελεσματικότερη θεραπεία του. Το preStandard ENV 13606 της Ευρωπαϊκής Επιτροπής Τυποποίησης (CEN) χαρακτηρίζει τον ηλεκτρονικό ιατρικό φάκελο ως "αποθήκη" όλων των πληροφοριών που αφορούν στο ιατρικό ιστορικό του ασθενούς. Επιπλέον υποστηρίζει πως αποτελεί τη βάση της διάγνωσης και της θεραπευτικής αντιμετώπισης του ασθενούς, όπως αναφέρθηκε και παραπάνω και τη βάση επιδημιολογικών ερευνών ενώ ταυτόχρονα παρέχει πληροφορίες διοικητικής, οικονομικής και στατιστικής φύσεως, καθώς και ποιοτικού ελέγχου. Σε έναν ηλεκτρονικό ιατρικό φάκελο εισάγονται πολλών τύπων δεδομένα σε δύο μορφές:

1. Σε μορφή ελεύθερου κειμένου και
2. Σε δομημένη μορφή.

Τα δεδομένα σε μορφή κειμένου καταγράφονται σε φυσική γλώσσα (narrative data) και μπορεί να είναι από κάποιον editor ή μέσω συστήματος αναγνώρισης φωνής, γεγονός που δίνει πλήρη ελευθερία έκφρασης. Αυτά προκύπτουν από τις ιατρικές σημειώσεις, τις σημειώσεις του νοσηλευτικού και διοικητικού προσωπικού και τα αποτελέσματα διαφόρων κλινικών και εργαστηριακών εξετάσεων. Αντίθετα, οι πληροφορίες σε δομημένη μορφή καταχωρούνται στη βάση δεδομένων του Η.Ι.Φ. σε κωδικοποιημένη μορφή, η οποία επιτυγχάνεται με βάση κάποια τεχνικά πρότυπα και δίνει πιο αξιόπιστα δεδομένα σε σχέση με την προηγούμενη μορφή. Ένα **πρότυπο** ορίζει γενικές δομές πληροφορίας και κοινά χαρακτηριστικά σε κάθε ηλεκτρονικό ιατρικό φάκελο, δηλαδή καθορίζει ένα λογικό μοντέλο, χωρίς να καθορίζει ακριβώς τι ιατρική πληροφορία θα περιέχει ή πως θα υλοποιηθεί.

Διεθνή πρότυπα για τον ιατρικό φάκελο είναι τα παρακάτω:

## 1. Το πρότυπο ISO/TC 215

Ο διεθνής οργανισμός τυποποίησης ISO (**International Standards Organization**) έχει ιδρύσει την Τεχνική Επιτροπή 215 (TC 215) με στόχο την προτυποποίηση στον τομέα της Πληροφορικής Υγείας. Η αντίστοιχη Ευρωπαϊκή Επιτροπή Τυποποίησης (European Standards Committee-CEN) έχει δημοσιεύσει ένα Pre-Standard για τη αρχιτεκτονική ηλεκτρονικού ιατρικού φακέλου (**Electronic HealthCare Record**) με την ονομασία EN 13606. Η επιδίωξή της είναι ένα πρότυπο ιατρικού φακέλου, όπου η κατάλληλη πληροφορία θα είναι διαθέσιμη όταν και όπου απαιτείται η υποστήριξη αποφάσεων.

Το παρόν πρότυπο είναι το μοναδικό πρότυπο ειδικά για ηλεκτρονικό ιατρικό φάκελο και αποτελεί τη βάση για συνεργασία με άλλα πρότυπα, όπως το HL7.

## 2. Το πρότυπο HL7

Η γλώσσα ή το πρότυπο **Health Level Seven** είναι ένα σύνολο από ανοιχτά πρότυπα, που επιτρέπει σε ετερογενή ιατρικά πληροφοριακά συστήματα να επικοινωνούν μεταξύ τους. Ονομάστηκε έτσι, γιατί κάνει τα πρωτόκολλα και τις δομές για την ανταλλαγή μηνυμάτων ιατρικού ενδιαφέροντος στο επίπεδο 7 (level seven) του γνωστού μοντέλου ISO. Το HL7 ενεκρίθη το έτος 1994 από τον Εθνικό Αμερικανικό Οργανισμό Προτύπων (American National Standards Institution-ANSI). Το πρότυπο αυτό έχει μεγάλη διάδοση τόσο στις ΗΠΑ, όσο και σε άλλες χώρες και η έκδοση 2.3 (1997) έχει υλοποιηθεί ευρύτατα. Τελευταία όμως, βρίσκεται σε εξέλιξη η έκδοση HL7 Version 3, η οποία πλεονεκτεί έναντι των προηγούμενων εκδόσεων διότι:

- Χρησιμοποιεί αντικειμενοστραφή μεθοδολογία
- Ορίζει ένα κατανοητό και κοινό για όλους μοντέλο πληροφορίας (Reference Information Model-RIM).
- Παρέχει μεθόδους για τη χρήση κωδικών και ιατρικών λεξικών για την ανταλλαγή μηνυμάτων με διάφορες εξωτερικές πηγές.
- Αξιολογεί με αξιόπιστο τρόπο την συμμόρφωση μιας συγκεκριμένης υλοποίησης με το πρότυπο HL7.
- Χρησιμοποιεί τα πρότυπα γλώσσας XML για τη σύνταξη των μηνυμάτων. Η XML (eXtensible Markup Language) είναι μια οικογένεια τεχνολογιών που στοχεύουν στην αναγνώριση και στο χειρισμό δομών σε ένα κλινικό έγγραφο, όπως οι διαδικαστικές σημειώσεις.

- Τέλος, θα είναι το ποιο οριστικό πρότυπο HL7, αφήνοντας λίγα περιθώρια για προαιρετικά χαρακτηριστικά.

Ένα σύγχρονο πληροφοριακό σύστημα νοσοκομείου (ή, γενικότερα, ένα πληροφοριακό σύστημα υγείας) χρησιμοποιεί ένα πρότυπο HL7 στον κεντρικό διακομιστή δεδομένων [ή στην ενδιάμεση βάση δεδομένων (Database Interface)] για να δέχεται, να οργανώνει, να κωδικοποιεί σε κατηγορίες και να αποστέλλει τις πληροφορίες στους ιατρικούς φακέλους των ασθενών, που βρίσκονται στη βάση δεδομένων για τους ασθενείς.

## Η χρήση του HL7 στην Ελλάδα

Ειδικότερα το HL7 είναι ένας κώδικας επικοινωνίας -π.χ. κατά το πρότυπο του τραπεζικού συστήματος "Δίας"-, το οποίο επιτρέπει την επίτευξη διαλειτουργικότητας, δηλαδή την ηλεκτρονική επικοινωνία ετερογενών πληροφοριακών συστημάτων (ανταλλάσσοντας δεδομένα μέσω τυποποιημένων μηνυμάτων). Η εφαρμογή του, ανοίγει το δρόμο για σημαντικές δράσεις όπως τη δημιουργία Φακέλου Υγείας των Πολιτών, Κάρτα Υγείας και Ασφάλισης, την ανάπτυξη Τηλεϊατρικής και κατ οίκον φροντίδας, τη δημιουργία Εθνικών Μητρώων όπως Νεοπλασιών Μεταμοσχεύσεων, Αίματος και πολλές ακόμα δράσεις που θα αλλάξουν το προφίλ της Υγείας και της Κοινωνικής Ασφάλισης στη χώρα μας. Έως σήμερα, σε 112 δημόσια νοσοκομεία της χώρας -σε σύνολο 132- έχουν εγκατασταθεί πληροφοριακά συστήματα τα οποία όμως -λόγω του διαφορετικού σχεδιασμού τους- δεν «επικοινωνούν». Ήδη έχει ολοκληρωθεί η εφαρμογή του HL7 σε 80 από τα 132 νοσοκομεία της επικράτειας και σύντομα θα ξεκινήσει η πιλοτική εφαρμογή της ηλεκτρονικής διασύνδεσης των παραπάνω μεταξύ τους. Για την εφαρμογή του Ηλεκτρονικού Φακέλου Υγείας σε εθνικό επίπεδο το χρονοδιάγραμμα είναι η πενταετία. Ειδικότερα σε Συνέντευξη τύπου με αφορμή το 9ο Διεθνές Συνέδριο Διαλειτουργικότητας HL7 2008 και το 3ο Πανελλήνιο Συνέδριο HL7- Hellas ο κ. Πατούλης, πρόεδρος του HL7-Hellas και δήμαρχος Αμαρουσίου τόνισε πόσο σημαντική είναι η δυνατότητα επικοινωνίας και ανταλλαγής ιατρικών πληροφοριών. Οι υγειονομικές υπηρεσίες προς τον πολίτη θα προσφέρονται πιο γρήγορα, με μεγαλύτερη ασφάλεια που θα προκύπτει από την αποφυγή λαθών και κυρίως θα υπάρχει ολοκληρωμένη γνώση του ιατρικού φακέλου από το επιστημονικό προσωπικό. "Η εφαρμογή του Ηλεκτρονικού Φακέλου Υγείας σε εθνικό επίπεδο θα αλλάξει τα δεδομένα καθώς αποτελεί βασικό εργαλείο για την ιατρική περίθαλψη η οποία έχει ως επίκεντρο τον ασθενή", σχολίασε ο πρόεδρος του HL7-Hellas και συμπλήρωσε ότι "βοηθά στην αύξηση της ποιότητας και της αποδοτικότητας, βελτιώνει την επικοινωνία μεταξύ διαφορετικών ειδικών που εργάζονται στον τομέα της Υγείας και μπορεί να συμβάλλει στην μείωση του κόστους με την αποφυγή για παράδειγμα "διπλών εξετάσεων". Παράλληλα ο κ. Πατούλης έφερε ως παράδειγμα της χρήσης του HL7 στα νοσοκομεία το γεγονός ότι θα υπάρχει η δυνατότητα, ένας αναλυτής στο εργαστήριο του νοσοκομείου να δέχεται απευθείας εντολές εξετάσεων

από τα κλινικά τμήματα και να επιστρέφει τις απαντήσεις των εξετάσεων που διενεργεί στα τμήματα που τις παρήγγειλαν αυτόματα.

**Όσον αφορά τα δεδομένα που περιέχει ένας ιατρικός φάκελος είναι τα εξής:**

- Δημογραφικά δεδομένα και στοιχεία ταυτοποίησης του ασθενή
- Ιστορικό, κλινικές εξετάσεις και δεδομένα του ασθενή
- Εργαστηριακά αποτελέσματα
- Απεικονιστικές εξετάσεις όπως, εικόνες, βίντεο και οποιαδήποτε άλλη μορφή εικόνας μπορεί να χρησιμοποιηθεί
- Σήματα όπως για παράδειγμα ηλεκτροκαρδιογραφήματα τα οποία βρίσκονται σε μορφή βιοσημάτων
- Αποφάσεις, διαγνώσεις
- Παραπεμπτικά και φαρμακευτική αγωγή
- Πληροφορίες που αφορούν στοιχεία χρεώσεις
- Εξουσιοδοτήσεις για την πρόσβαση στα δεδομένα

Ο Η.Ι.Φ. **πλεονεκτεί** έναντι των "κλασικών" φακέλων καθότι:

- Η αποθήκευση και ανάκληση των δεδομένων γίνεται γρήγορα και με ασφάλεια, χάρη στους Η/Υ.
- Επιπλέον, καθίσταται δυνατή η επεξεργασία των δεδομένων και η άμεση μεταφορά τους, με ηλεκτρονικά μέσα, σε οποιαδήποτε απόσταση.
- Το κόστος είναι μικρότερο, καθώς η μεταφορά είναι άμεση και δεν υπάρχει κόστος αποθήκευσης.
- Η ψηφιοποίηση των αρχείων καταργεί την αλλοίωση τους από τη χρήση και τον χρόνο και διευκολύνει τη συλλογή δεδομένων για την επιδημιολογία και τις κλινικές μελέτες.
- Μπορούν να χρησιμοποιηθούν ταυτόχρονα από πολλούς χρήστες.
- Δεν υπάρχει πρόβλημα αναγνωσιμότητας.

Όταν, για παράδειγμα, ο ιατρός βρίσκεται μπροστά σε μια άδεια κόλλα χαρτί, κυριολεκτικά μπορεί να γράψει οτιδήποτε. Αυτό που θα γράψει δεν είναι βέβαιο ότι θα είναι χρησιμοποιήσιμο από αυτούς που θα το διαβάσουν. Αυτό μπορεί να οφείλεται σε

διάφορους λόγους, που ξεκινούν από πιθανόν δυσδιάκριτο γραφικό χαρακτήρα και φτάνουν μέχρι την παράλειψη σημαντικών στοιχείων που πιθανόν δεν απασχολούν τον συγγραφέα αλλά αποτελούν καίρια στοιχεία που πρέπει να ξέρει ο αναγνώστης. Με λίγα λόγια, το περιεχόμενο του ιατρικού φακέλου εξαρτάται στον μέγιστο βαθμό από την ικανότητα του συγγραφέα του να καταγράψει και να αποδώσει σωστά τα στοιχεία που το αποτελούν. Βέβαια, αυτό δεν είναι ένα πρόβλημα που οφείλεται αποκλειστικά στην προσέγγιση του paper-based ιατρικού φακέλου, αλλά στις διαδικασίες και την προσέγγιση που ακολουθούνται στη συμπλήρωση του ιατρικού φακέλου. Πάντως, δεν είναι τυχαίο ότι αυτό το φαινόμενο συναντάται σχεδόν αποκλειστικά στην περίπτωση του paper-based ιατρικού φακέλου.

Όπως έχουν δείξει διάφορες μελέτες πάνω στο θέμα [Disk and Steen, 1991], οι ιατρικοί φάκελοι είναι απροσπέλαστοι σε ποσοστό 30% του χρόνου σε μεγάλους οργανισμούς (νοσοκομεία, κ.λπ.), ενώ συνήθως το περιεχόμενό τους είναι διασκορπισμένο σε διαφορετικά σημεία: γραφεία ιατρών, νοσοκομεία, διαγνωστικά κέντρα, κ.λπ.

Η πληθώρα ιατρικών φακέλων σε έναν οργανισμό μπορεί να είναι μοιραία όσο αφορά την χρησιμοποίησή τους από τους εργαζόμενους σε αυτόν, μια και το κόστος σε χρόνο και χρήμα για τη σωστή αποθήκευση και ταξινόμηση πιθανόν χιλιάδων φακέλων είναι τεράστιο.

Στην περίπτωση του paper-based ιατρικού φακέλου δεν μπορούμε να μιλάμε για την άμεση χρησιμοποίηση του περιεχομένου πληθώρας ιατρικών φακέλων για έρευνα, μια και κάτι τέτοιο απαιτεί ιδιαίτερη προσπάθεια (ανάγνωση των φακέλων, κωδικοποιημένη καταγραφή στοιχείων τους κ.λπ.) και το σημαντικότερο: **χρόνο**.

Έχει αποδειχθεί ότι το κλινικό προσωπικό κατ' εξακολούθηση αποτυγχάνει στην ανεύρεση πληροφοριών από ένα paper-based ιατρικό φάκελο κατά τη διάρκεια μιας συνεδρίας με ασθενή: στη μελέτη 168 περιπτώσεων αποδείχθηκε ότι αναζητήθηκαν και δεν βρέθηκαν πληροφορίες σε ποσοστό 81%. Στο 95% αυτών των περιπτώσεων ο ιατρικός φάκελος δεν ήταν διαθέσιμος κατά τη διάρκεια της συνεδρίας. Τα ποσοστά ανά κατηγορία μη διαθέσιμης πληροφορίας ήταν 36% για πληροφορίες που αφορούσαν εργαστηριακές εξετάσεις και πράξεις, 23% για φαρμακευτική και θεραπευτική αγωγή, 31% για ιατρικό ιστορικό και 10% για άλλες πληροφορίες. Όπως χαρακτηριστικά αναφέρεται, η μελέτη δεν εξετάζει τον αντίκτυπο που είχε η μη εύρεση της πληροφορίας στην εξέλιξη και το αποτέλεσμα της θεραπείας που τελικά ακολουθήθηκε για τους ασθενείς αυτούς.

Η πραγματικότητα πάντως είναι ότι, παρά τα μειονεκτήματα, υπάρχει τουλάχιστον ένα μέρος ιατρών που προτιμούν τον paper-based ιατρικό φάκελο στην καθημερινή τους κλινική ρουτίνα.

**Ο Η.Ι.Φ. μειονεκτεί έναντι των "κλασικών" ιατρικών φακέλων, καθότι:**

- Είναι απαραίτητη η ειδική εκπαίδευση του ιατρικού προσωπικού για τη χρήση τους.
- Είναι πιθανόν να διακοπεί η λειτουργία του Η/Υ με αποτέλεσμα να χαθούν δεδομένα.
- Δεν υπάρχει η δυνατότητα ελευθερίας έκφρασης.
- Το κόστος εγκατάστασης και συντήρησης των Η/Υ είναι αρκετά μεγάλο.
- Υπάρχει κίνδυνος παραβίασης ιατρικού απορρήτου.

Ο όρος **Ηλεκτρονικός Φάκελος Υγείας (ΗΦΥ)** χρησιμοποιείται διεθνώς περισσότερο από τον όρο ΗΙΦ. Στην Ελλάδα εξακολουθούν να χρησιμοποιούν τον όρο ΗΙΦ, μάλλον για παραδοσιακούς λόγους. Τα τελευταία χρόνια στην Ευρώπη ακούγεται όλο και περισσότερο ο όρος **Φάκελος Υγείας Του Πολίτη (ΦΥΠ)** [Citizen Health Record (CHR)]. Ο όρος αυτός είναι ο πιο αντιπροσωπευτικός από όλους τους προηγούμενους όρους και υπερκαλύπτει τη καταγραφή και συντήρηση του περιεχομένου του ιατρικού φακέλου αντιμετωπίζοντας επιτυχώς όλα τα προβλήματα που προκύπτουν από την ηλεκτρονική φύση του.

Συμπερασματικά, τα **χαρακτηριστικά** ενός σύγχρονου αυτοματοποιημένου ιατρικού φακέλου θα πρέπει να είναι:

- **Ασφάλεια**, την οποία πρέπει να διασφαλίζει η εμπιστευτικότητα των πληροφοριών και να πραγματοποιείται με πιστή καταγραφή των κλινικών ενεργειών του χρήστη. Ειδικότερα, περιέχει :
  - i. Έλεγχο πρόσβασης για να εκτελεί ο καθένας από τους εξουσιοδοτημένους χρήστες τις συγκεκριμένες λειτουργίες που του επιτρέπουν.
  - ii. Ηλεκτρονική υπογραφή, την οποία ελέγχει και τροποποιεί μόνο ο δημιουργός της.
  - iii. Ακεραιότητα δεδομένων, σύμφωνα με την οποία δεν μπορεί να διαγραφεί ή τροποποιηθεί καμία πληροφορία μετά την καταχώρησή της.
  - iv. Λογισμικό έλεγχο για την καταγραφή σε αρχείο του εξουσιοδοτημένου χρήστη που έκανε κάποια τροποποίηση, την ώρα και το είδος που έγινε αυτή.
- **Διασυνδεσιμότητα**, που εξασφαλίζει ένα κοινό περιβάλλον για όλους τους χρήστες. Αυτό επιτρέπει τη διανομή και την ανταλλαγή δεδομένων, καθώς και την αυτοματοποιημένη επεξεργασία τους σε διαφορετικά συστήματα.
- **Ευρύτητα-περιεκτικότητα**, ώστε να υπάρχει η δυνατότητα υποστήριξης πολλών τύπων δεδομένων σε μορφή ελευθέρου κειμένου και δομημένη μορφή (πχ. ακτινογραφίες, καρδιογραφήματα).
- **Μεταφερσιμότητα**, ώστε τα συστήματα των ιατρικών φακέλων να μπορούν να μεταφέρονται και ενσωματώνονται σε διάφορα ιδρύματα, ανεξάρτητα από το λογισμικό, την εθνική γλώσσα και το υλικό που χρησιμοποιείται.
- **Διαχρονική συμβατότητα**, ώστε να υπάρχει η δυνατότητα υποστήριξης του ιατρικού φακέλου για μεγάλα χρονικά διαστήματα.

- **Επεκτασιμότητα**, ώστε να μπορούν να ενταχθούν στον ιατρικό φάκελο νέες εφαρμογές.
- **Διαθεσιμότητα**, σύμφωνα με την οποία το σύστημα είναι διαθέσιμο 24 ώρες το εικοσιτετράωρο και όλες της ημέρες της εβδομάδας.
- **Χρήση προτύπων**, ώστε να καθορίζονται γενικές δομές πληροφορίας και κοινά χαρακτηριστικά σε κάθε αυτοματοποιημένο ιατρικό φάκελο.

## 4. Παρόμοιες εφαρμογές στην Ελλάδα

Πλέον ο Ηλεκτρονικός Ιατρικός Φάκελος εξυπηρετεί πολλά νοσοκομεία και γιατρούς παγκοσμίως αφού επιλύει πολλά και σημαντικά προβλήματα όπως είναι η δυσκολία στην αποτίμηση της συνολικής εικόνας του ασθενούς, οι χειροκίνητες διαδικασίες ενημέρωσης του ιατρικού φακέλου του ασθενούς, η διατήρηση πολλών και διαφορετικών φυσικών αποθηκευτικών χώρων με συνέπειες στο χρόνο και στο κόστος λειτουργίας του Νοσοκομείου, η φθορά ή απώλεια κρίσιμων και ευαίσθητων εγγράφων, η δυσχερής και χρονοβόρα οργάνωση, ταξινόμηση και αρχειοθέτηση εγγράφων ασθενών, οι περιορισμένες δυνατότητες αναζήτησης και ανάκτησης στο ιατρικό αρχείο και η δύσκολη πρόσβαση των στελεχών στο Ιατρικό Φάκελο. Πολλά από τα παραπάνω προβλήματα τα συναντάμε και στην Ελλάδα, γι' αυτό και έχει δημιουργηθεί η ανάγκη για ανάπτυξη παρόμοιων εφαρμογών που θα λύσουν τα χέρια πολλών γιατρών αλλά και ασθενών. Μία από τις πρώτες προσπάθειες που έγιναν για να ξεπεραστούν τα προβλήματα αυτά των νοσοκομείων μας, είναι ο *Ηλεκτρονικός Ιατρικός Φάκελος Ασθενών v 1.0 (EMR)*<sup>[1]</sup>. Πιο συγκεκριμένα η εφαρμογή αυτή αναπτύχθηκε από τον ιατρό Ευτύχιο Κωνσταντουδάκη και τους συνεργάτες του, με στόχο, να μπορεί να βοηθήσει την καθημερινή δουλειά του κάθε ιατρού στο ιατρείο του. Σκοπός του είναι να μειώσει όσο το δυνατό την γραφειοκρατία της ιατρικής πράξης, δίνοντας έτσι περισσότερο χρόνο στον ασθενή. Ακόμα με την συλλογή των στοιχείων μπορούν να βγούνε πάρα πολύ χρήσιμα στατιστικά στοιχεία, που σίγουρα μπορούν να φανούν πολύ χρήσιμα κυρίως στην πρόληψη. Η εφαρμογή ξεκίνησε να φτιάχεται και πήρε την μορφή beta τον Μάρτιο του 2006, όπου άρχισε να την δοκιμάζει καθημερινά στο αγροτικό ιατρείο που ήταν τότε. Τον Απρίλιο του 2006 είχε φτάσει στην τελική του μορφή, όπου άρχισε επίσημα πια να λειτουργεί κανονικά στο ιατρείο που είχε τότε. Αυτή ήταν η πρώτη έκδοση v1.0. Η εφαρμογή φτιάχτηκε με την βοήθεια της Microsoft Access 2003. Ο ΗΦΑ μπορεί να δουλέψει χωρίς πρόβλημα σε συστήματα με πλήρως εγκατεστημένο το Microsoft Office 2002/2003/2007 (Όχι δηλαδή τυπική εγκατάσταση) με την απαραίτητη ύπαρξη φυσικά της Access. Έχει φτιαχτεί με τέτοιο τρόπο ώστε να υπάρχει δυνατότητα αναβάθμισης χωρίς πρόβλημα, με την προσθήκη νέων δυνατοτήτων και βελτιώνοντας τις ήδη υπάρχουσες. Τον Ιούνιο του 2006 εγκαθίσταται σε όλα τα Π.Ι. που βρίσκονται υπό την ευθύνη του Κέντρου Υγείας Ιτέας. Σύνολο η εφαρμογή τρέχει πια σε 11 ιατρεία πια. Σε 5 από αυτά, η εφαρμογή τρέχει σε μορφή δικτύου, σε δύο υπολογιστές, με κοινή βάση δεδομένων για τους δύο χειριστές-ιατρούς του Περιφερειακού Ιατρείου, αφού τα ιατρεία αυτά είναι διαθέσια. Τον Σεπτέμβριο του 2006 εγκαθίσταται και στα 5 Περιφερειακά Ιατρεία που είναι υπό την ευθύνη του Κέντρου Υγείας Λιδορικού, οπότε και με αυτά το πρόγραμμα τρέχει πια σε ολόκληρο τον νομό Φωκίδας. Το πρόγραμμα αυτή την στιγμή λειτουργεί/τρέχει σε όλα τα Π.Ι. περιοχή ευθύνης του Κ.Υ. Ιτέας από τον Ιούνιο του 2006, με απόλυτη επιτυχία. Από τις αρχές Σεπτεμβρη του 2006, τρέχει και σε όλα τα Π.Ι. που είναι υπό την ευθύνη του Κ.Υ. Λιδορικού. Το πρόγραμμα έχει δοθεί από τους κατασκευαστές του προς χρήση στο Κ.Υ. Ιτέας και το Γενικό Νοσοκομείο Αμφισσας εντελώς δωρεάν, με σκοπό να εξοπλιστούν όλα τα Π.Ι. της περιοχής ευθύνης του Κ.Υ. Ιτέας αρχικά και στην συνέχεια τα Π.Ι. στην περιοχή ευθύνης του Κ.Υ. Λιδορικού. Πλέον μετά την αναβάθμιση της εφαρμογής που έγινε, άρχισε να δίνεται με την εμπορική του πια μορφή.



Η Κρήτη έχει να παρουσιάσει σημαντικές πρωτιές σε αυτόν τον τομέα τόσο στη χρήση του ηλεκτρονικού φακέλου ασθενή όσο και στην χρήση κωδικοποιήσεων στην πρωτοβάθμια φροντίδα υγείας. Ο δανεισμός από την Σουηδία του αγγλόφωνου προγράμματος Eginodoit και η χρήση του ως ηλεκτρονικού φακέλου ασθενή (βασισμένο στην κωδικοποίηση ICHPPC-2 Defined, πρόγονο του ICPC) στο κέντρο υγείας Σπηλίου από το 1988 αποτέλεσε την πρώτη καινοτόμα προσπάθεια. Η χρήση του Fakelos (σε MS Access) με την χρήση του ICPC-1 και του ICD-9 αρχικά στο κέντρο υγείας Ανωγείων και αργότερα σε πάνω από 30 ιατρεία πρωτοβάθμιας σε όλη την Ελλάδα αποτέλεσε ένα σημαντικό βήμα στην χρήση ηλεκτρονικού ιατρικού φακέλου. Η χρηματοδότηση από το Β' ΚΠΣ και η γιγαντιαία προσπάθεια του Ιδρύματος Τεχνολογίας και Έρευνας Κρήτης ένωσε όλες τις μονάδες υγείας σε ένα δίκτυο, το HygeiaNet και άλλαξε προς το καλύτερο σημαντικές πτυχές της καθημερινής ζωής στην υγεία στην Κρήτη. Η πιλοτική εφαρμογή ενός ηλεκτρονικού φακέλου ασθενή με δυνατότητες "on the job training" με βάση την προηγούμενη εμπειρία βασισμένο στο ICPC-2 και στο ICD-10 αποτελεί την παρούσα κατάσταση.

Ένα άλλο παράδειγμα Ηλεκτρονικού Ιατρικού Φακέλου είναι αυτός που είναι εγκατεστημένος στην 1η Πανεπιστημιακή Παιδιατρική Κλινική του Ιπποκρατείου Νοσοκομείου Θεσσαλονίκης<sup>[2]</sup>, όπου η εφαρμογή επιτρέπει στην κλινική να οργανώνει τις ιατρικές πληροφορίες των ασθενών της και να δημιουργεί τα ενημερωτικά σημειώματα με αυτοματοποιημένο τρόπο. Επιπλέον έχει την δυνατότητα να καταγράφει την πορεία νόσου κατά την εισαγωγή του ασθενή, να προβάλλει στατιστικά και συγκεντρωτικών αναφορών καθώς και να προσαρμόζεται στην ηλικία ή στο φύλο του ασθενή.

Το 2009 στο νοσοκομείο «Γ. Παπαγεωργίου» εισάγονται σύγχρονα ηλεκτρονικά συστήματα. Ο Ηλεκτρονικός Ιατρικός Φάκελος του Ασθενούς, που εφαρμόζεται για πρώτη φορά ολοκληρωμένα σε νοσηλευτικό ίδρυμα της χώρας, ανοίγει νέο κεφάλαιο στη λειτουργία του βορειοελλαδικού νοσοκομείου. Το πληροφορικό σύστημα άρχισε να αναπτύσσεται σταδιακά πριν από μία περίπου τετραετία. Ολοκληρώθηκε πρόσφατα και στο εξής ξεκινά η προσπάθεια εφαρμογής του από το σύνολο των κλινικών και των εργαστηρίων του νοσοκομείου.

Σε όλα τα τμήματα καταγράφεται ηλεκτρονικά κάθε ιατρική πράξη, από τα εξωτερικά ιατρεία ως τις κλινικές και τα εργαστήρια. Επιπλέον, υπάρχει διασύνδεση με τα απεικονιστικά εργαστήρια. Η ιατρική εικόνα από το ακτινολογικό εργαστήριο, το ενδοσκοπικό και το εργαστήριο πυρηνικής ιατρικής θα είναι πλέον διαθέσιμη σε κάθε ιατρό μέσα από τον Ηλεκτρονικό Υπολογιστή του, συνοδευόμενη από το πόρισμα και το ενημερωτικό σημείωμα του ασθενούς (διάγνωση, χρόνος νοσηλείας, θεραπεία, θέραπων γιατρός, πρακτικά χειρουργείου, ακτινογραφίες, κ.λ.π.). Το νέο μοντέλο απλοποιεί τη διαδικασία διαχείρισης του ιατρικού φακέλου, διευκολύνοντας τόσο τη δουλειά του ιατρού όσο και του ασθενούς. «Αναζητώντας τρόπο για την εξοικονόμηση πόρων, βασικά ανθρώπινου δυναμικού, δηλαδή χρόνου, προτείναμε την εφαρμογή χρήσης του ηλεκτρονικού φακέλου και ταυτόχρονη μείωση χρήσης του κλασικού ιατρικού φακέλου», εξήγησε ο διευθυντής πληροφορικής ΓΝΠ κ. Γεώργιος Ιωακειμίδης. Με τον Ηλεκτρονικό Φάκελο επιτυγχάνεται ταχύτερη και άμεση ενημέρωση του ιατρού. Μειώνεται ο χρόνος διάγνωσης και των επαναληπτικών εξετάσεων. Περιορίζονται οι επαναληπτικές εξετάσεις, ενώ μειώνεται ο χρόνος που αφιέρωνε το προσωπικό για αναζήτηση φακέλου. Η πλήρης εφαρμογή του

Ηλεκτρονικού Φακέλου προς το παρόν ποικίλει από κλινική σε κλινική, όπως δηλώθηκε στην ομιλία του κ. Γεωργίου Ιωακειμίδη. Σημαντικά βήματα προς την κατεύθυνση αυτή έχουν κάνει οι δύο ορθοπαιδικές κλινικές, η χειρουργική, η οφθαλμολογική, ενώ στο ακτινολογικό τέθηκε σε εφαρμογή η ηλεκτρονική διαχείριση εικόνας.

Πέρα από τα νοσοκομεία και τις κλινικές, οι ιατρικοί φάκελοι άρχισαν να εφαρμόζονται και από δήμους για την καλύτερη εξυπηρέτηση των δημοτών τους. Παράδειγμα αυτής της τακτικής αποτελεί ο δήμος Πυλαίας-Χορτιάτη, όπου σε συνεργασία με το Εθνικό Διαδημοτικό Δίκτυο Υγιών Πόλεων Προαγωγής Υγείας, προσφέρει δωρεάν στους κατοίκους, την υπηρεσία του Ηλεκτρονικού Ιατρικού Φακέλου<sup>[3]</sup>. Έτσι οι δημότες έχουν την δυνατότητα, μέσω αυτής της υπηρεσίας, να έχουν 24ωρη πρόσβαση στο ιατρικό ιστορικό τους, να καταχωρούν ιατρικά δεδομένα, ραντεβού, καθώς και να ενημερώνονται πάνω σε ιατρικά θέματα.

Τέλος, μία από τις πιο σύγχρονες και πιο πρόσφατες εφαρμογές είναι αυτή που είναι εγκατεστημένη στο Τζάνειο Νοσοκομείο και συγκεκριμένα στην Α' Χειρουργική Κλινική<sup>[4]</sup>. Η εφαρμογή αποτελείται από ένα ηλεκτρονικό σύστημα αρχειοθέτησης των ασθενών (Ramma), όπου οι γιατροί έχουν την δυνατότητα να εισάγουν δεδομένα και πληροφορίες σχετικά με τον ασθενή. Όπως και στους προηγούμενους ιατρικούς φακέλους, έτσι και σε αυτήν την περίπτωση η δυνατότητα πρόσβασης των γιατρών είναι 24ωρη κάτι που εξυπηρετεί στο να λαμβάνονται γρηγορότερα οι θεραπευτικές αποφάσεις. Αυτό όμως που ενισχύει την χρησιμότητα του ιατρικού φακέλου και της εύκολης πρόσβασής του, είναι δυνατότητα που έχουν οι γιατροί να επικοινωνούν και να ανταλλάσσουν συμβουλές με άλλα νοσηλευτικά ιδρύματα, πάνω σε ιατρικά θέματα των ασθενών τους.

## 5. Γενική Εξέταση Αίματος

Η Γενική Εξέταση Αίματος είναι μία από τις συχνότερα ζητούμενες και διενεργούμενες εργαστηριακές εξετάσεις αίματος. Σε αυτήν μελετώνται τόσο ποσοτικά όσο και μορφολογικά όλα τα έμμορφα συστατικά (κύτταρα) του αίματος, τα οποία αιωρούνται μέσα σε υγρό περιβάλλον που ονομάζεται πλάσμα. Τα στοιχεία αυτά παράγονται και ωριμάζουν αρχικά στον μυελό των οστών και στη συνέχεια υπό φυσιολογικές συνθήκες απελευθερώνονται στο αίμα ανάλογα με τις ανάγκες. Η ποσοτική μελέτη αφορά τον ολικό αριθμό ή την εκατοστιαία αναλογία των αιμοσφαιρίων, δηλαδή των κυττάρων του αίματος (Ερυθρών, Λευκών, Αιμοπεταλίων), ενώ μορφολογικά αναζητούνται μεταβολές ή αλλοιώσεις αναφορικά με το μέγεθος, το σχήμα, το είδος καθώς και άλλων φυσικών χαρακτηριστικών των αιμοσφαιρίων.

Είναι ίσως η πιο σημαντική εξέταση αίματος, αφού τα ευρήματα της δίνουν πολύτιμες διαγνωστικές πληροφορίες για την κατάσταση και την εικόνα της υγείας του ανθρώπινου οργανισμού.



Στη Γενική Εξέταση Αίματος προσδιορίζονται η Αιμοσφαιρίνη, ο Αιματοκρίτης, ο αριθμός των Ερυθροκυττάρων, των Λευκοκυττάρων (με τον Λευκοκυτταρικό τους τύπο), των Αιμοπεταλίων καθώς επίσης και σειρά άλλων αιματολογικών παραμέτρων με ιδιαίτερη σημασία ο καθένας τους.

Με την εισαγωγή σύγχρονων αιματολογικών αναλυτών υψηλής τεχνολογίας και την αυτοματοποίηση των μετρήσεων, οι προσδιορισμοί αυτοί έγιναν όχι μόνο ταχύτεροι αλλά και ακριβέστεροι.

Ο προσδιορισμός όλων αυτών των αιματολογικών παραμέτρων είναι πάρα πολύ σημαντικός και χρήσιμος για τη διάγνωση πολλών νοσημάτων όπως, (μικροβιακών και ιογενών λοιμώξεων), αιμοσφαιρινοπαθειών και αναιμιών (κληρονομικών, αιμολυτικών, σιδηροπενικών), αιμορραγιών ή θρομβώσεων και λευχαιμιών. Εξίσου σημαντικός και χρήσιμος είναι επίσης ο προσδιορισμός των αιματολογικών παραμέτρων και για την παρακολούθηση της πορείας πολλών παθήσεων και νοσημάτων καθώς και για την αποτελεσματικότητα της ακολουθούμενης θεραπείας.

Η διαταραχή οποιουδήποτε τμήματος του ανθρώπινου οργανισμού μπορεί να προκαλέσει αλλαγές στη σύσταση του αίματος. Τα διάφορα συστατικά που αποτελούν το αίμα αντικατοπτρίζουν τις περισσότερες από τις λειτουργίες του οργανισμού. Κάτω από φυσιολογικές συνθήκες τα συστατικά αυτά βρίσκονται σε σταθερά και φυσιολογικά επίπεδα. Οι όποιες αλλαγές όμως στη φυσιολογική λειτουργία του οργανισμού, συχνά συνοδεύονται και από αλλαγές τόσο στα ποσοτικά όσο και στα μορφολογικά χαρακτηριστικά των συστατικών του αίματος, επιτρέποντας έτσι τη συγκέντρωση χρήσιμων στοιχείων απαραίτητων για τη σωστή και έγκαιρη διάγνωση.

Στον παρακάτω πίνακα απεικονίζονται οι φυσιολογικές τιμές των εξετάσεων αλλά και το τι σημαίνουν:

<b>TEST</b>	<b>ΕΞΕΤΑΣΗ</b>	<b>ΜΟΝΑΔΑ ΜΕΤΡΗΣΗΣ</b>	<b>ΦΥΣΙΟΛΟΓΙΚΕΣ ΤΙΜΕΣ</b> A: Άνδρες Γ:Γυναίκες
<b>Hb (Hemoglobin):</b> Κύριο συστατικό των ερυθρών αιμοσφαιρίων. Δεσμεύει και μεταφέρει οξυγόνο από τους πνεύμονες στους ιστούς. Συμβάλλει στη μεταφορά διοξειδίου του άνθρακα από τους ιστούς στους πνεύμονες και στη ρύθμιση της οξεοβασικής ισορροπίας του αίματος. Καθοριστική μέτρηση για αξιολόγηση διαφόρων μορφών αναιμίας.	Αιμοσφαιρίνη	g/dl	A: 13.5-17.5 Γ: 12.0-16.0
<b>Hct (Hematocrit):</b> Καθορίζει την εκατοστιαία αναλογία του όγκου των ερυθρών αιμοσφαιρίων στον συνολικό όγκο του αίματος. Χρήσιμος δείκτης για παρουσία αναιμίας, απώλειας αίματος και αφυδάτωσης.	Αιματοκρίτης	%	A: 41-53 Γ: 36-46
<b>RBC (Red Blood Cells):</b> Παράγονται στον μυελό των οστών και έχουν μέσο όρο ζωής 120 μέρες. Μεταφέρουν, μέσω της αιμοσφαιρίνης που περιέχουν, οξυγόνο	Αριθμός Ερυθρών αιμοσφαιρίων	10 <sup>12</sup> /L	A: 4.50-6.50 Γ: 3.80-5.80

από τους πνεύμονες στους ιστούς. Καθοριστική μέτρηση για παρουσία αναιμίας ή πολυκυτταραιμίας.			
<b>MCV (Mean Cell Volume):</b> Σημαντικός δείκτης του όγκου των ερυθρών αιμοσφαιρίων και ταξινόμησης αναιμιών. Αυξάνεται στις μεγαλοβλαστικές αναιμίες και στα μυελοδυσπλαστικά σύνδρομα. Ελαττώνεται στις σιδηροπενικές αναιμίες, στις θαλασσαιμίες και σε αναιμίες λόγω χρόνιων παθήσεων.	Μέσος όγκος Ερυθρών αιμοσφαιρίων	fl	78-98
<b>MCH (Mean Cell Haemoglobin):</b> Δείκτης της περιεκτικότητας αιμοσφαιρίνης στα ερυθρά αιμοσφαίρια. Σημαντικός δείκτης για τη ταξινόμηση αναιμιών. Εκφράζει τη μέση περιεκτικότητα αιμοσφαιρίνης που περιέχεται σε κάθε ερυθρό αιμοσφαίριο. Αυξάνεται και ελαττώνεται στις ίδιες καταστάσεις όπως το MCV.	Μέση περιεκτικότητα Αιμοσφαιρίνης	pg	26-34
<b>MCHC (Mean Cell Haemoglobin Concentration):</b> Δείκτης της μέσης πυκνότητας αιμοσφαιρίνης στο μέσο ερυθρό αιμοσφαίριο. Ιδιαίτερα χρήσιμος δείκτης για τη ταξινόμηση	Μέση πυκνότητα Αιμοσφαιρίνης	g/dl	31.5-37.5

αναιμιών. Αυξάνεται στην αφυδάτωση και στην κληρονομική σφαιροκυττάρωση, ενώ ελαττώνεται στην υπερυδάτωση, στις σιδηροπενικές αναιμίες και στις θαλασσαιμίες.			
<b>RDW-CV (Red Distribution Width-Coefficient Variation):</b> Χρήσιμος δείκτης διερεύνησης αιματολογικών διαταραχών. Αποτελεί δείκτη της απόκλισης ή μεταβολής στο μέγεθος των ερυθρών αιμοσφαιρίων. Ανιχνεύει ανωμαλίες των ερυθρών αιμοσφαιρίων οι οποίες σχετίζονται με ανισοκυττάρωση.	Εύρος κατανομής μεγέθους Ερυθρών αιμοσφαιρίων (με συντελεστή μεταβλητότητας)	%	11-15
<b>RDW-SD (Red Distribution Width-Standard Deviation):</b> Χρήσιμος δείκτης διερεύνησης αιματολογικών διαταραχών. Αποτελεί δείκτη της απόκλισης ή μεταβολής στο μέγεθος των ερυθρών αιμοσφαιρίων. Ανιχνεύει ανωμαλίες των ερυθρών αιμοσφαιρίων οι οποίες σχετίζονται με ανισοκυττάρωση.	Εύρος κατανομής μεγέθους Ερυθρών αιμοσφαιρίων (με σταθερή απόκλιση)	fl	37-47
<b>PLT (Platelets):</b> Τα αιμοπετάλια ή θρομβοκύτταρα είναι μικρά απύρρηνα κύτταρα απαραίτητα στη λειτουργία της αιμόστασης, στον μηχανισμό της πήξης του αίματος, στον σχηματισμό θρόμβων και στη διαδικασία επούλωσης πληγών.	Αριθμός Αιμοπεταλίων	$10^9/L$	150-400
<b>MPV (Mean Platelet Volume):</b> Σημαντικός δείκτης του όγκου των	Μέσος όγκος Αιμοπεταλίων	fl	8-12

αιμοπεταλίων και αξιολόγησης αιματολογικών και αιμορραγικών διαταραχών.			
<b>PDW (Platelet Distribution Width):</b> Δείκτης μεγέθους αιμοπεταλίων. Χρήσιμος δείκτης διερεύνησης αιμορραγικών διαταραχών.	Εύρος κατανομής μεγέθους Αιμοπεταλίων	%	12-28
<b>PCT (Plateletcrit):</b> Δείκτης εκατοστιαίας αναλογίας αιμοπεταλίων ανά μονάδα όγκου αίματος. Χρήσιμος δείκτης διερεύνησης αιμορραγικών διαταραχών.	Αιμοπεταλιοκρίτης	%	0.190-0.290
<b>WBC (White Blood Cells):</b> Κύτταρα απαραίτητα για την άμυνα και την επιβίωση του οργανισμού. Καταπολεμούν τις λοιμώξεις και προστατεύουν τον οργανισμό από κάθε βλαβερή ουσία. Σημαντική εξέταση για την ύπαρξη και τη βαρύτητα μιας νοσηρής κατάστασης στον οργανισμό.	Αριθμός Λευκών αιμοσφαιρίων	10 <sup>9</sup> /L	4.0-10.8
<b>Differential count:</b> Τα λευκά αιμοσφαίρια διακρίνονται σε 3 κύριους τύπους. Πολυμορφοπύρρηνα (Ουδετερόφιλα, Ηωσινόφιλα, Βασεόφιλα), Λεμφοκύτταρα και Μονοκύτταρα. Κάθε τύπος παίζει τον δικό του ρόλο στη λειτουργία του ανοσοποιητικού συστήματος.	Λευκοκυτταρικός τύπος (Διαφορική μέτρηση)	%	-
<b>Neutrophils:</b> Παρέχουν προστασία στον οργανισμό	Ουδετερόφιλα (Πολυμορφοπύρρηνα)	%	40-75

<p>έναντι βακτηριακών λοιμώξεων. Μεταναστεύουν στην περιοχή του τραυματισμού ή της μόλυνσης και καταπολεμούν τη μόλυνση με φαγοκυττάρωση. Αύξηση παρατηρείται σε οξείες λοιμώξεις και φλεγμονές. Ελαττώνονται με φαρμακολογική αγωγή και σε αυτοάνοσα νοσήματα.</p>			
<p><b>Lymphocytes:</b> Θεωρούνται τα βασικά κύτταρα του ανοσοποιητικού συστήματος. Διαδραματίζουν σημαντικό ρόλο κυρίως ενάντια στις ιογενείς λοιμώξεις. Αυξημένα λεμφοκύτταρα (λεμφοκυττάρωση) έχουμε σε οξείες ή χρόνιες ιογενείς λοιμώξεις (ιογενή νοσήματα, ιλαρά, ανεμοβλογιά, ηπατίτιδα, βρουκέλλωση, σύφιλη, λοιμώδη μονοπυρήνωση κλπ.) και σε λεμφοκυτταρικές λευχαιμίες και λεμφώματα. Ελαττώνονται (λεμφοπενία) σε κληρονομικές ανοσολογικές διαταραχές, στη φυματίωση, στο AIDS, στη νόσο Hodgkin, απλαστική αναιμία και μετά από χορήγηση κορτιζόνης, ανοσοκατασταλτικών φαρμάκων και χημειοθεραπείας</p>	<p>Λεμφοκύτταρα</p>	<p>%</p>	<p>20-45</p>
<p><b>Monocytes:</b> Αντιμετωπίζουν βαριές λοιμώξεις με φαγοκυττάρωση. Αυξάνονται κατά τη διάρκεια χρόνιων λοιμώξεων (τύφο, φυματίωση), σε αυτοάνοσα</p>	<p>Μονοκύτταρα</p>	<p>%</p>	<p>2-10</p>




νοσήματα, στη μυελομονοκυτταρική λευχαιμία και σε άλλες κακοήθειες (Hodgkin).			
---	--	--	--

## 6. Εγχειρίδιο Χρήσης

Ανοίγοντας την εφαρμογή, μας ζητείται να συνδεθούμε στο σύστημα [1]. Εδώ έχουμε δύο επιλογές: είτε να συνδεθούμε σαν γιατρός, είτε σαν ασθενής.

Εικόνα 1. Σύνδεση

Συνδεδεμένοι σαν γιατρός, η εφαρμογή μας μεταφέρει στον κατάλογο με τους ασθενείς μας [2],[3]. Εδώ τώρα όπως βλέπουμε έχουμε πολλές επιλογές. Στα αριστερά του πίνακα έχουμε τις επιλογές να επεξεργαστούμε τα στοιχεία ενός ασθενή ή να διαγράψουμε κάποιον. Στα δεξιά του πίνακα έχουμε την επιλογή της εισαγωγής των αποτελεσμάτων κάποιας εξέτασης στον ασθενή και την επιλογή για να δούμε το ιστορικό του για όποια εξέταση θέλουμε.



Κατάλογος Ασθενών

Γραφείο γιατρού	ΑΜΚΑ	Όνομα	Επίθετο	Όνομα Πατέρα	Όνομα Μητέρας	Τηλέφωνο 1	Τηλέφωνο 2	Διεύθυνση	Νούμερο	Περιοχή	Πόλη	ΤΚ	Ηλικία	
<p>Ώρες επίσκεψη: Δευτέρα-Παρασκευή 8-12μ, 15-21μ</p> <p>Καλώς ήλθετε Γυναίκε</p> <p>Αποσύνδεση</p>	<a href="#">Επεξεργασία Διαγραφή</a>	22222222	Παναγιώτης	Παπάς	Γώργος	Πόπη	2107845128	694255897	Αιόλου	69	Καμίνια	Αθήνα	15561	21
	<a href="#">Επεξεργασία Διαγραφή</a>	11111111	Παύλος	Κορινθίου	Κώστας	Πηνελόπη	2105698742	698563214	Ξανθίου	15	Χολαργός	Αθήνα	15561	35
	<a href="#">Επεξεργασία Διαγραφή</a>	1122336655	Γιώργος	Αφεντουλίδης	Μάριος	Αναστασία	2106512397	697452123	Παπαφλέσσα	23	Κηφισιά	Αθήνα	69854	36
	<a href="#">Επεξεργασία Διαγραφή</a>	1199887744	Πάτροκλος	Πετράκης	Χρήστος	Κατερίνα	2109854756	2147483647	Ψηλορίτη	85	Πατήσια	Αθήνα	65874	32
	<a href="#">Επεξεργασία Διαγραφή</a>	7744112255	Κώστας	Μπούκας	Δημήτρης	Άννα	2102389745		Αιόλου	74	Καμίνια	Αθήνα	22222	27
	<a href="#">Επεξεργασία Διαγραφή</a>	4444444444	Παύλος	Λαλάκης	Κώστας	Βάσω	2106598745	2147483647	Εξαρχείων	56	Εξάρχεια	Αθήνα	56897	45

Εικόνα 2 Κατάλογος Ασθενών α



Κατάλογος Ασθενών

ΑΜΚΑ	Όνομα	Επίθετο	Όνομα Πατέρα	Όνομα Μητέρας	Τηλέφωνο 1	Τηλέφωνο 2	Διεύθυνση	Νούμερο	Περιοχή	Πόλη	ΤΚ	Ηλικία	Επάγγελμα	Προσθήκη Εξέτασης	Ιστορικά
<a href="#">πθ</a> 22222222	Παναγιώτης	Παπάς	Γώργος	Πόπη	2107845128	694255897	Αιόλου	69	Καμίνια	Αθήνα	15561	21	Δημόσιος Υπάλληλος	<a href="#">Προσθήκη Εξέτασης</a>	<a href="#">Ιστορικό</a>
<a href="#">πθ</a> 11111111	Παύλος	Κορινθίου	Κώστας	Πηνελόπη	2105698742	698563214	Ξανθίου	15	Χολαργός	Αθήνα	15561	35	Σερβιτόρος	<a href="#">Προσθήκη Εξέτασης</a>	<a href="#">Ιστορικό</a>
<a href="#">πθ</a> 1122336655	Γιώργος	Αφεντουλίδης	Μάριος	Αναστασία	2106512397	697452123	Παπαφλέσσα	23	Κηφισιά	Αθήνα	69854	36	Γιατρός	<a href="#">Προσθήκη Εξέτασης</a>	<a href="#">Ιστορικό</a>
<a href="#">πθ</a> 1199887744	Πάτροκλος	Πετράκης	Χρήστος	Κατερίνα	2109854756	2147483647	Ψηλορίτη	85	Πατήσια	Αθήνα	65874	32	Ανεργος	<a href="#">Προσθήκη Εξέτασης</a>	<a href="#">Ιστορικό</a>
<a href="#">πθ</a> 7744112255	Κώστας	Μπούκας	Δημήτρης	Άννα	2102389745		Αιόλου	74	Καμίνια	Αθήνα	22222	27	Προγραμματιστής	<a href="#">Προσθήκη Εξέτασης</a>	<a href="#">Ιστορικό</a>
<a href="#">πθ</a> 4444444444	Παύλος	Λαλάκης	Κώστας	Βάσω	2106598745	2147483647	Εξαρχείων	56	Εξάρχεια	Αθήνα	56897	45	Φοιτητής	<a href="#">Προσθήκη Εξέτασης</a>	<a href="#">Ιστορικό</a>

Εικόνα 3 Κατάλογος ασθενών β

Έστω ότι θέλουμε να προσθέσουμε κάποια αποτελέσματα μιας εξέτασης στον ασθενή μας. Αυτόματα η εφαρμογή μας ζητάει να επιλέξουμε την εξέταση στην οποία θα καταχωρίσει τα αποτελέσματα [4].

# ehealth record

Γραφείο γιατρού

Ώρες επίσκεψης:  
Δευτέρα-Παρασκευή  
8-12πμ, 15-21μμ

**Καλώς ήλθατε Γιατρέ**

Αποσύνδεση

Παρακαλώ επιλέξτε εξέταση:

Εξέταση

Πίσω

Εικόνα 4 Επιλογή Εξέτασης

Αφού επιλέξουμε την εξέταση περνάμε στην εισαγωγή των αποτελεσμάτων <sup>[5]</sup> και αφού συμπληρώσουμε όλα τα πεδία που μας ζητάει πατάμε “Καταχώρηση” για να γίνει η αποθήκευση. Τέλος αν όλα πάνε καλά μας εμφανίζει ένα μήνυμα επιβεβαίωσης <sup>[6]</sup> ότι όλα λειτούργησαν σωστά και γυρνάμε πάλι στην σελίδα με τους ασθενείς μας.

Γενική Αίματος

AMKA Ασθενή 22222222

Αριθμός Λευκών αιμοσφαιρίων(WBC)  10<sup>9</sup>/L

Αριθμός Ερυθρών αιμοσφαιρίων(RBC)  10<sup>12</sup>/L

Αιμοσφαιρίνη(Hb)  g/dl

Αιματοκρίτης(HCT)  %

Μέσος όγκος Ερυθρών αιμοσφαιρίων(MCV)  fl

Μέση περιεκτικότητα Αιμοσφαιρίνης(MCH)  pg

Αριθμός Αιμοπεταλίων(PLT)  10<sup>9</sup>/L

Μέσος όγκος Αιμοπεταλίων(MPV)  fl

Ημερομηνία Εξέτασης

Ιούλιος		Ιούλιος 2015					Αύγουστος	
Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ	Κυρ		
29	30	1	2	3	4	5		
6	7	8	9	10	11	12		
13	14	15	16	17	18	19		
20	21	22	23	24	25	26		
27	28	29	30	31	1	2		
3	4	5	6	7	8	9		

Καταχώριση

Πίσω

Browse... No file selected.

Αποθήκευση

Εικόνα 5 Εισαγωγή Αποτελεσμάτων



Εικόνα 6 Μήνυμα Επιβεβαίωσης

Επιπλέον, στην προηγούμενη λειτουργία, έχουμε την δυνατότητα να εισάγουμε αποτελέσματα εξετάσεων, μέσω ενός XML αρχείου, το οποίο αποθηκεύεται στην αντίστοιχη εξέταση του ασθενή στην βάση δεδομένων μας. Και σε αυτήν την περίπτωση εάν η καταχώρηση γίνει με επιτυχία θα μας βγάλει το ανάλογο μήνυμα.

Εάν τώρα επιλέξουμε να δούμε το ιστορικό του ασθενή, θα πρέπει να επιλέξουμε την εξέταση που μας ενδιαφέρει [7] και εν συνεχεία μας παρουσιάζεται ένας πίνακας με όλα τα στοιχεία και αποτελέσματα του ασθενή που μας ενδιαφέρουν. Εδώ έχουμε την επιλογή επίσης να αποθηκεύσουμε τα αποτελέσματα αυτά στον υπολογιστή με το πάτημα του κουμπιού “Εξαγωγή σε XML”, όπου μας αποθηκεύει τον αντίστοιχο πίνακα στον σκληρό δίσκο του υπολογιστή μας.



Εικόνα 7 Εμφάνιση Εξετάσεων

Ο γιατρός επίσης έχει την δυνατότητα να εγγράψει έναν καινούργιο ασθενή-πελάτη του πατώντας το κουμπί “Προσθήκη Ασθενή”. Αυτόματα μεταφερόμαστε στην

φόρμα συμπλήρωσης των απαραίτητων στοιχείων για την εγγραφή του ασθενή<sup>[8]</sup>. Αφού ολοκληρωθεί επιτυχώς η εγγραφή των στοιχείων, ο ασθενής έχει πλέον καταχωρηθεί στην λίστα του γιατρού.

**Γραφείο γιατρού**  
 Ώρες επίσκεψης:  
 Δευτέρα-Παρασκευή  
 8-12πμ, 15-21μμ

*Καλώς ήλθατε Γιατρέ*

Αποσύνδεση

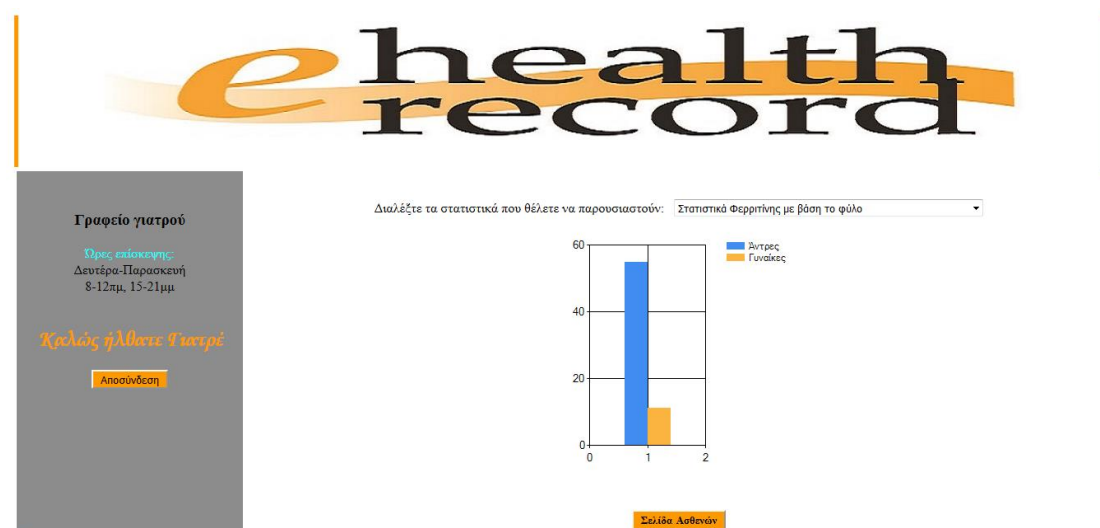
ΑΜΚΑ  
 Όνομα  
 Επίθετο  
 Όνομα Πατέρα  
 Όνομα Μητέρας  
 Τηλέφωνο1  
 Τηλέφωνο2  
 Οδός  
 Αριθμός  
 Περιοχή  
 Πόλη  
 ΤΚ  
 Ηλικία  
 Φύλο  
 Επάγγελμα

Εγγραφή

Σπίσω

Εικόνα 8 Εγγραφή Ασθενή

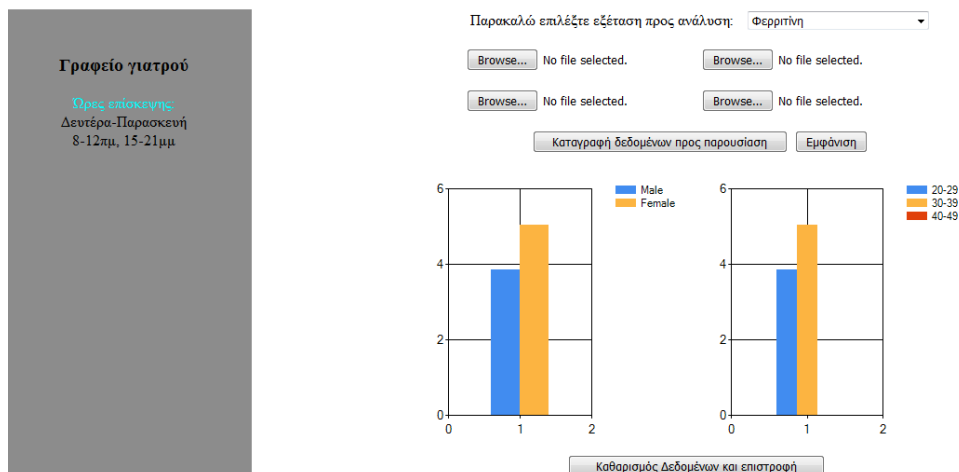
Ο γιατρός έχει την δυνατότητα, επιπλέον, να έχει μπροστά του τα στατιστικά στοιχεία των ασθενών του με βάση δύο παράγοντες: το φύλο και την ηλικία. Έτσι αφού πατήσουμε το κουμπί “Στατιστικά” στην αρχική σελίδα, μπορούμε να επιλέξουμε όποια στατιστικά θέλουμε από όποια εξέταση μας ενδιαφέρει<sup>[9]</sup>.



Εικόνα 9 Στατιστικά Στοιχεία

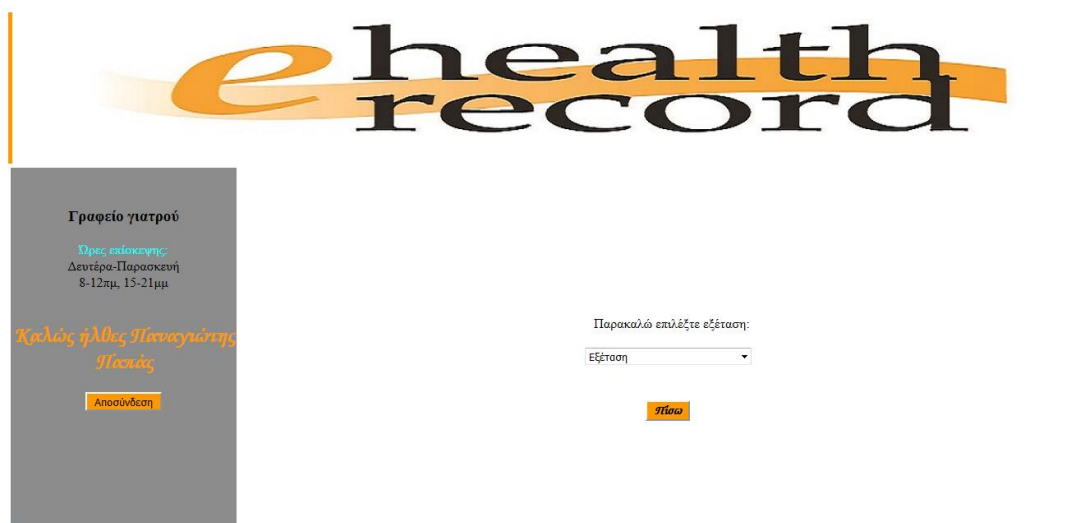
Τέλος, ο γιατρός πατώντας την επιλογή “Άμεση παρατήρηση στατιστικών μέσω εισαγωγής εξετάσεων” έχει την δυνατότητα να εξάγει άμεσα στατιστικά στοιχεία, μέσω εξετάσεων που δεν βρίσκονται στην βάση του αλλά από εξετάσεις

που βρίσκονται στον υπολογιστή του σε μορφή XML. Έτσι λοιπόν, αφού επιλέξει την εξέταση που θέλει, στην συνέχεια επιλέγει τα αρχεία απ' όπου θα εξάγει τα στατιστικά και αφού πατήσει την επιλογή “Καταγραφή δεδομένων προς παρουσίαση”, επιλέγει “εμφάνιση” για να του παρουσιαστούν οι αντίστοιχοι πίνακες με τα στατιστικά<sup>[10]</sup>.



Εικόνα 10 Στατιστικά μέσω δεδομένων από τον υπολογιστή

Τώρα, εάν από την αρχική σελίδα δεν συνδεθούμε σαν τον γιατρό αλλά σαν ασθενής, έχουμε την δυνατότητα να δούμε όλο το ιστορικό μας, για όποια εξέταση θέλουμε και φυσικά να την αποθηκεύσουμε στον υπολογιστή μας για προσωπική χρήση<sup>[10][11]</sup>.



Εικόνα 11 Επιλογή Εξέτασης

# ehealth record

**Γραφείο γιατρού**  
Ώρες επίσκεψης:  
Δευτέρα-Παρασκευή  
8-12πμ, 15-21μμ

Παρακαλώ επιλέξτε εξέταση:

Γενική Αίματος

amka	wbc	rbc	hb	hct	mcv	mnh	plt	mpv	date
222222222	1	1	2	344	87	6	8	9	15/7/2015
222222222	5	5	5	5	5	5	5	5	25/12/2015
222222222	5	8	9	7	6	6	3	5	29/9/2015
222222222	456	5	6	7	6	433	5	5	30/9/2015
222222222	5	5	5	5	5	5	5	5	5/5/2015

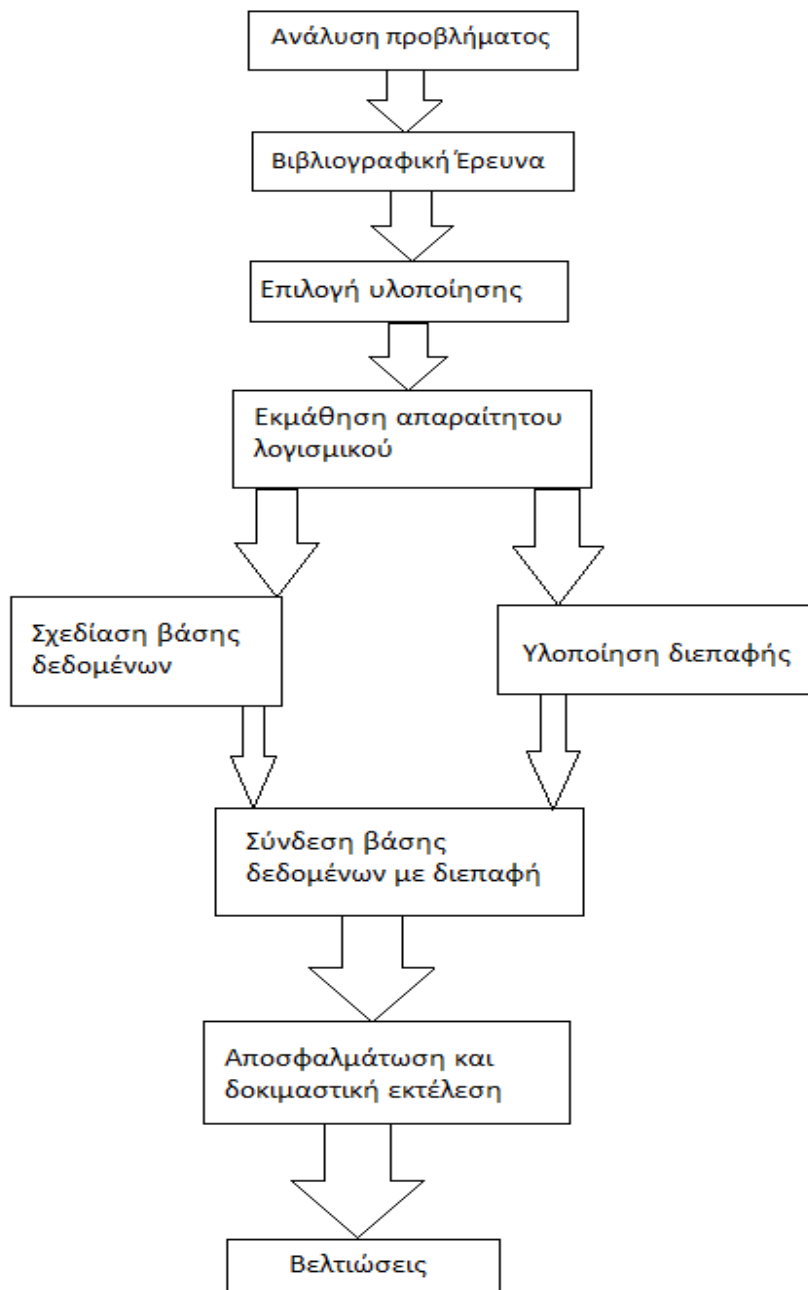
[Εξαγωγή σε ΧΜΛ](#)

[Πίσω](#)

Εικόνα 12 Εμφάνιση Ιστορικού Εξέτασης

## 7. Διάγραμμα Ροής

Το διάγραμμα ροής αντιστοιχεί σε μια απεικόνιση των βημάτων που ακολουθήθηκαν για την επίτευξη των στόχων της εφαρμογής μας. Το διάγραμμα χωρίζεται σε επίπεδα, καθένα από τα οποία αποτελεί μια ανεξάρτητη διεργασία. Η πορεία εξέλιξης της εργασίας ταυτίζεται με την σειριακή ανάγνωση του διαγράμματος ροής, ενώ οι περιπτώσεις στις οποίες ένα επίπεδο του διαγράμματος περιλαμβάνει περισσότερα πεδία ερμηνεύονται ως η παράλληλη υλοποίηση των επιμέρους διεργασιών.



Εικόνα 13 Διάγραμμα ροής



3. **Ανάλυση του προβλήματος:** Στο πρώτο βήμα του διαγράμματος ροής προσπαθήσαμε να αντιληφθούμε και να καταγράψουμε τις απαιτήσεις του δοθέντος προβλήματος.
  4. **Βιβλιογραφική έρευνα:** Στο βήμα αυτό έγινε η διερεύνηση της τρέχουσας χρησιμοποιούμενης τεχνολογίας και των σημερινών συνθηκών στους χώρους που μας απασχολούν ανατρέχοντας σε επιστημονικά συγγράμματα και μέσα από χρήση του διαδικτύου.
  5. **Επιλογή υλοποίησης:** Στο βήμα αυτό επιλέξαμε την υλοποίηση που κρίναμε ως βέλτιστη λύση του δοθέντος προβλήματος.
- **Εκμάθηση απαραίτητου λογισμικού:** Στο βήμα αυτό εξοικειωθήκαμε με το περιβάλλον του Visual Studio και της MYSQL. προγράμματα σε γλώσσα C#, μικρά σε έκταση, και μέσα από την αποσφαλμάτωση και εκτέλεσή τους κατανοήσαμε τον τρόπο λειτουργίας του λογισμικού και της δυνατότητες που μας παρέχει ως προς τη διασύνδεση με μια βάση δεδομένων.
  - **Σχεδίαση βάσης δεδομένων:** Σε αυτό το στάδιο δημιουργήσαμε τους απαραίτητους πίνακες για την αποθήκευση των ασθενών καθώς και τους πίνακες των εξετάσεων που περιλαμβάνονται στον Ηλεκτρονικό Ιατρικό Φάκελο.
  - **Υλοποίηση διεπαφής:** Στο βήμα αυτό προσπαθήσαμε να κατασκευάσουμε ένα απλό και φιλικό interface τόσο για τον γιατρό, όσο και για τον ασθενή.
  - **Σύνδεση βάσης δεδομένων με διεπαφή:** Στο στάδιο αυτό γράψαμε τον κώδικα που συνδέει την βάση δεδομένων μας με την διεπαφή.
  - **Αποσφαλμάτωση και δοκιμαστική εκτέλεση:** Στο βήμα αυτό έγινε ο έλεγχος ορθής λειτουργίας του προγράμματος μέσα από έναν αριθμό εκτέλεσης του σε επίπεδο debugger. Καταγράψαμε το σύνολο σφαλμάτων και προειδοποιήσεων που επέστρεψε το πρόγραμμα και έγιναν οι απαραίτητες διορθώσεις στον κώδικα μας.
  - **Βελτιώσεις:** Στο τελευταίο αυτό στάδιο του διαγράμματος, μετά από μια σειρά δοκιμών και ελέγχων που έγιναν, πραγματοποιήθηκαν οι τελικές βελτιώσεις για την σωστή και ομαλή λειτουργία της εφαρμογής μας.

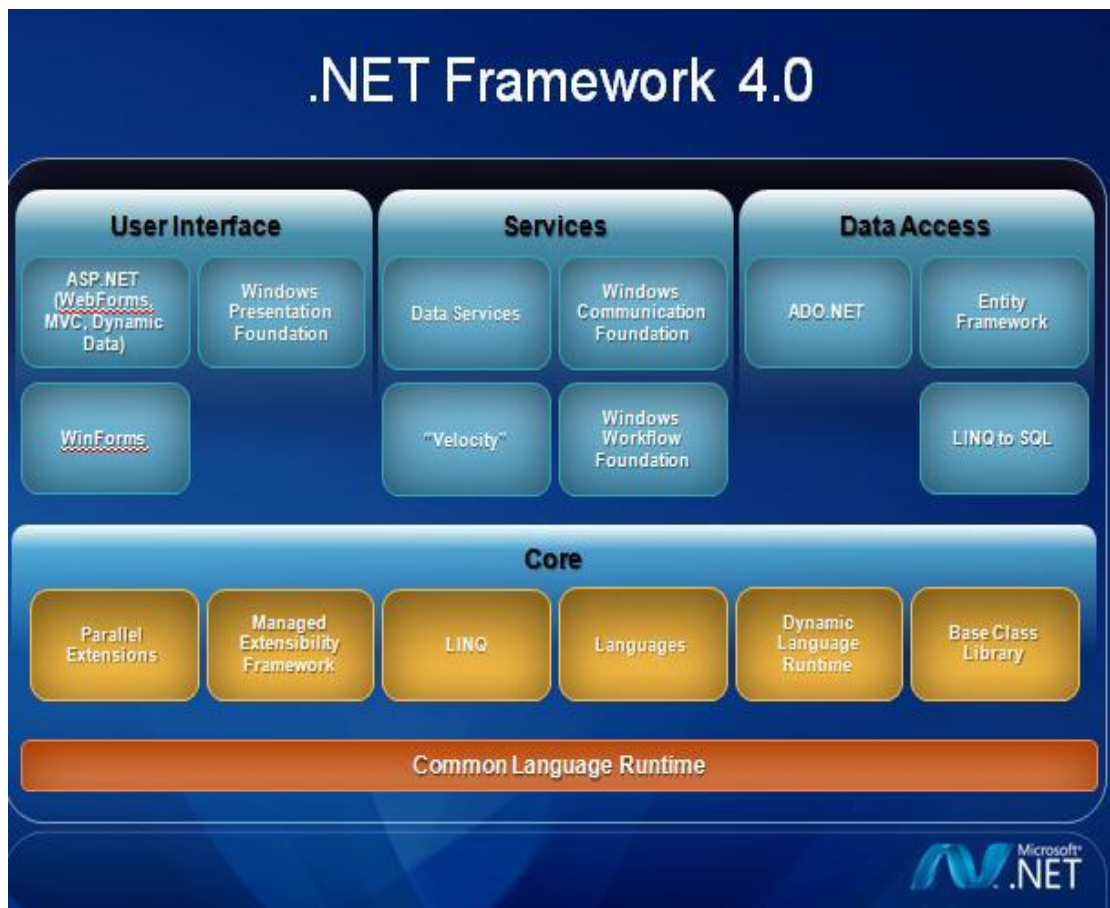
## 8. .NET FRAMEWORK

Το .NET Framework [5] είναι μια πλατφόρμα ανάπτυξης λογισμικού, για την δημιουργία εφαρμογών που προορίζονται για τα λειτουργικά συστήματα Windows, Windows phone, Windows Server και Windows Azure. Αποτελείται από την Common Language Runtime (CLR) , που ουσιαστικά είναι ένα εικονικό περιβάλλον εκτέλεσης των προγραμμάτων .NET, και την βιβλιοθήκη κλάσεων .NET Framework class library. Γενικά, το .NET Framework παρέχει ένα ευέλικτο αλλά ταυτόχρονα ελεγχόμενο περιβάλλον εκτέλεσης εφαρμογών, ευκολία στην ανάπτυξη εφαρμογών και project, και ταυτόχρονη χρήση πολλών γλωσσών προγραμματισμού, οι οποίες είναι οι .NET Languages.

Πιο συγκεκριμένα οι βασικές ψηφίδες του .NET Framework είναι οι ακόλουθες:

- .NET Framework Base Class Library. Περιλαμβάνει πολλές βιβλιοθήκες κλάσεων με σκοπό να παρέχουν έτοιμο κώδικα στους προγραμματιστές, για λειτουργίες που θεωρείται ότι είναι συχνά χρησιμοποιούμενες, και μπορούν να μοντελοποιηθούν αποτελεσματικά. Παραδείγματα τέτοιων λειτουργιών είναι η εγγραφή και ανάγνωση σε αρχεία ή στην οθόνη, η επικοινωνία με βάσεις δεδομένων, η δημιουργία γραφικών στοιχείων κλπ.
- .NET Languages. Είναι ένα σύνολο γλωσσών προγραμματισμού που μπορούν να συνδυαστούν, και στο ίδιο Project, και η επικοινωνία μεταξύ τους. Αυτές οι γλώσσες είναι οι C#, Visual Basic, JS script, F#, J#, C++. Γενικά οι γλώσσες αυτές ακολουθούν τις αρχές του αντικειμενοστραφούς προγραμματισμού.
- ASP.NET. Το ASP.NET είναι ένα server-side Web application Framework που έχει σχεδιαστεί με σκοπό να παρέχει την ικανότητα στους Web Developers να δημιουργούν δυναμικές ιστοσελίδες. Είναι ένα σχετικά καινούριο Framework, που εμφανίστηκε το 2002. Είναι δομημένο στο Common Language Runtime (CLR), και επιτρέπει στους προγραμματιστές να χρησιμοποιήσουν την .NET γλώσσα προγραμματισμού της αρεσκείας τους. Το ASP.NET υποστηρίζει τρία διαφορετικά μοντέλα ανάπτυξης, που είναι τα Web Pages, Web Forms, και το MVC που είναι και αυτό που χρησιμοποιήθηκε για την ανάπτυξη του παρόντος πληροφοριακού συστήματος.
- Common Language Runtime (CLR). Το εικονικό περιβάλλον (virtual machine), που ελέγχει και εκτελεί τις .NET εφαρμογές.
- Garbage Collection. Το εικονικό περιβάλλον CLR είναι υπεύθυνο για την αποδέσμευση των μη χρησιμοποιούμενων πόρων που δεσμεύουν οι .NET εφαρμογές.

- Common Type System (CTS), και Common Language Specification (CLS). Είναι ένα σύνολο από κανόνες οι οποίοι έχουν σκοπό να καταστήσουν δυνατή την επικοινωνία μεταξύ των διαφορετικών .NET Languages. Με αυτόν τον τρόπο είναι δυνατή ή εύκολη μετατροπή κώδικα από μια γλώσσα σε μία άλλη, αφού ουσιαστικά οι τύποι των δεδομένων (τα αντικείμενα) υπακούουν σε κοινούς κανόνες, και έχουν το ίδιο interface προς τον προγραμματιστή.



Πριν επεκταθούμε σε μια πιο αναλυτική παρουσίαση της δομής της πλατφόρμας .NET και των τεχνολογιών που την περιβάλλουν και χρησιμοποιήθηκαν στην ανάπτυξη του παρόντος πληροφοριακού συστήματος είναι σκόπιμο να παρουσιάσουμε τα σημαντικότερα χαρακτηριστικά και πλεονεκτήματά της:

- **Υποστήριξη πολλαπλών γλωσσών.** Στο .NET, η διαλειτουργικότητα μεταξύ των γλωσσών επιτυγχάνεται μέσω της διαγλωσσικής κληρονομικότητας. Μαζί με ένα ενοποιημένο σύστημα τύπων, η συνένωση κώδικα γραμμένου σε διαφορετικές γλώσσες είναι πλέον εύκολη υπόθεση. Αυτό διευκολύνει και άλλα προγραμματιστικά πρότυπα όπως το συναρτησιακό της F#.

- **Ανάπτυξη βασισμένη σε components.** Η δημιουργία ανεξάρτητων κομματιών λογισμικού που περιέχουν βιβλιοθήκες κλάσεων και διαμοιράσιμα τμήματα λειτουργικότητας μιας εφαρμογής έχει γίνει πολύ πιο εύκολη από ότι ήταν παλιότερα. Η διαμοιράσιμη μονάδα κώδικα στο .NET βασίζεται στην έννοια του assembly, το οποίο μεταφέρει πληροφορίες που απαιτούνται για τη χρήση της όπως η έκδοση της.
- **Μεταφερισιμότητα.** Η Μεταφερισιμότητα (portability) επιτυγχάνεται μέσω της εικονικής μηχανής στο CLR. Η εικονική μηχανή αυτή είναι η προδιαγραφή μιας αφηρημένης εικονικής μηχανής για την οποία κώδικας γραμμένος σε οποιαδήποτε από τις .NET γλώσσες μεταγλωττίζεται σε μια ενδιάμεση γλώσσα γνωστή ως Common Intermediate Language (CIL) και περιέχεται σε ένα assembly. Ένα assembly είναι είτε ένα εκτελέσιμο αρχείο (exe) είτε ένα dll. Η μεταγλώττιση σε αυτήν την ενδιάμεση γλώσσα είναι που εξασφαλίζει και την ανεξαρτησία από την πλατφόρμα, η οποία ωστόσο μένει μόνο στη θεωρία, αφού η μοναδική πλήρης υλοποίηση του .NET είναι διαθέσιμη μόνο για την πλατφόρμα των Windows. Όμως είναι διαθέσιμη μια μερική υλοποίηση ανοιχτού κώδικα με το όνομα Mono.
- **Απλή εγκατάσταση εφαρμογών.** Αντίθετα με τα components που βασίζονται στο COM, δεν χρειάζεται καμία καταχώρηση στο registry του συστήματος για την εγκατάσταση των assemblies καθώς μια απλή αντιγραφή αρκεί για τη χρήση τους. Επιπρόσθετα τα πιθανά προβλήματα της χρήσης των Dynamic Link Libraries (DLLs) απαλείφθηκαν με την υποστήριξη της ύπαρξης πολλαπλών εκδόσεων του ίδιου component. Το ίδιο το .NET Framework είναι ένα καλό παράδειγμα αυτής της δυνατότητας, αφού είναι δυνατόν να υπάρχουν πολλαπλές εκδόσεις του framework εγκατεστημένες στο ίδιο μηχάνημα.
- **Επικοινωνία με υπάρχων κώδικα.** Παρόλο που η πλατφόρμα .NET προορίζεται να είναι ένα υποκατάστατο των παλαιότερων τεχνολογιών, υποστηρίζει τη χρήση κομματιών λογισμικού που έχουν αναπτυχθεί σε παλιότερες τεχνολογίες όπως της COM και παρέχει πρόσβαση σε λειτουργίες χαμηλού επιπέδου του λειτουργικού συστήματος μέσω του μηχανισμού P/Invoke.
- **Αξιοπιστία.** Το .NET είναι σχεδιασμένο για τη δημιουργία λογισμικού υψηλής αξιοπιστίας (robustness). Πραγματοποιεί εκτεταμένους ελέγχους κατά τη μεταγλώττιση αλλά και κατά την εκτέλεση των προγραμμάτων. Τα χαρακτηριστικά του .NET από μόνα τους ωθούν τους προγραμματιστές στην απόκτηση αξιόπιστων προγραμματιστικών συνηθειών. Το μοντέλο διαχείρισης της μνήμης είναι απλό: η δέσμευση μνήμης γίνεται με μια λέξη (new), δεν υπάρχουν δείκτες προς θέσεις μνήμης σαν τύποι δεδομένων, ούτε αριθμητική δεικτών ενώ η αποδέσμευση μνήμης γίνεται αυτόματα μέσω του μηχανισμού garbage collection. Αυτό το απλό μοντέλο διαχείρισης μνήμης απαλείφει ολόκληρες κατηγορίες λαθών που απασχολούν τους προγραμματιστές σε C και

C++. Η ανάπτυξη προγραμμάτων γίνεται με την πεποίθηση ότι αρκετά λάθη θα εντοπιστούν από το σύστημα πολύ πριν την ολοκλήρωσή της.

- **Αυτόματη διαχείριση μνήμης.** Άλλο ένα σημαντικό χαρακτηριστικό του .NET Framework είναι η δυνατότητα αυτόματης διαχείρισης της μνήμης μέσω του Garbage Collector. Ιστορικά η διαχείριση της μνήμης ήταν από τις δυσκολότερες εργασίες που είχε να κάνει ένας προγραμματιστής. Οι παλαιότερες προσεγγίσεις στη διαχείριση μνήμης ήταν είτε χαμηλού επιπέδου (όπως η malloc/free στη C και η new/delete στη C++), προκαλώντας διάφορα bugs, είτε αρκετά πολύπλοκη (όπως στην περίπτωση του COM). Ένας από τους στόχους του .NET και του CLR πιο συγκεκριμένα ήταν να απαλείψει από τους προγραμματιστές την ανάγκη διαχείρισης της μνήμης. Το CLR με το μηχανισμό Garbage Collector διαχειρίζεται τη μνήμη ερευνώντας πότε μπορεί να ελευθερωθεί με ασφάλεια μνήμη που δεν χρειάζεται άλλο σε ένα πρόγραμμα. Η μνήμη δεσμεύεται με την αρχικοποίηση των διαφόρων τύπων (αντικειμένων) από ένα διαθέσιμο τμήμα της μνήμης που διαχειρίζεται το CLR γνωστό ως heap (σωρός). Όσο υπάρχει κάποια άμεση ή έμμεση (μέσω του γράφου των αντικειμένων) αναφορά προς ένα αντικείμενο, τότε αυτό θεωρείται πως χρησιμοποιείται. Στην αντίθετη περίπτωση, όταν δεν μπορεί να αναφερθεί από κάποιο άλλο τμήμα του κώδικα τότε είναι διαθέσιμο για καταστροφή. Ο garbage collector τρέχει ανά τακτά χρονικά διαστήματα σε ένα ξεχωριστό thread, εντοπίζει ποια αντικείμενα είναι έτοιμα προς καταστροφή και αποδεσμεύει τη μνήμη που αυτά χρησιμοποιούσαν.
- **Ασφάλεια.** Το .NET είναι σχεδιασμένο να λειτουργεί σε κατανεμημένα περιβάλλοντα, η ασφάλεια (security) των οποίων είναι υψίστης σημασίας. Με μέτρα που είναι ενσωματωμένα στις γλώσσες αλλά και κατά την εκτέλεση των προγραμμάτων, το .NET επιτρέπει τη δημιουργία προγραμμάτων που δεν μπορούν να προσβληθούν από έξω. Σε δικτυακά περιβάλλοντα οι εφαρμογές .NET είναι ασφαλείς από εισβολές μη εξουσιοδοτημένου κώδικα που επιχειρεί να δράσει στο παρασκήνιο και να εγκαταστήσει κακόβουλο λογισμικό ή να εισβάλει σε συστήματα αρχείων. Το περιβάλλον εκτέλεσης του .NET έχει ενσωματωμένο έναν μηχανισμό ασφάλειας με το όνομα Code Access Security (CAS) που απομονώνει τον κώδικα ανάλογα με την προέλευσή του, τον δημιουργό του ή άλλες πολιτικές. Το μοντέλο ασφάλειας συμβαδίζει με τους μηχανισμούς ασφάλειας που παρέχει το λειτουργικό σύστημα όπως οι Access Control Lists και τα Windows security tokens.
- **Απόδοση.** Η ενδιάμεση γλώσσα στην οποία μεταγλωττίζονται τα προγράμματα στο .NET, μεταγλωττίζεται πάντα με τη μέθοδο Just-in-Time (JIT). Ο μεταγλωττιστής JIT μεταγλωττίζει κάθε τμήμα του κώδικα στην ενδιάμεση γλώσσα όταν αυτό καλείται (just in time). Όταν έχει ήδη μεταγλωττιστεί κάποιο τμήμα κώδικα, το αποτέλεσμα αποθηκεύεται μέχρι η εφαρμογή να τερματίσει έτσι ώστε να μην χρειάζεται να ξαναμεταγλωττιστεί όταν ζητηθεί το συγκεκριμένο τμήμα. Η διαδικασία αυτή είναι αποδοτικότερη σε σύγκριση με την απευθείας μεταγλώττιση ολόκληρου του κώδικα, γιατί υπάρχει περίπτωση μεγάλα τμήματα κώδικα να μην εκτελεστούν ποτέ στην

πραγματικότητα. Με τον μεταγλωττιστή JIT τέτοιος κώδικας δεν θα μεταγλωττίζονταν ποτέ.

- **Υποστήριξη με επαγγελματικά εργαλεία.** Το .NET Framework συνοδεύεται από ένα σύνολο εργαλείων για την υποστήριξη της ανάπτυξης λογισμικού σε αυτό, όπως το σύστημα σχεδιασμού, τα frameworks για unit testing, την πλατφόρμα για build MSBuild και το σύστημα debugging, μέσα από το Visual Studio 2013. Το Visual Studio είναι ένα από τα καλύτερα Integrated Development Environments (IDEs). Χρησιμοποιείται για την ανάπτυξη όλων των ειδών εφαρμογών που μπορούν να αναπτυχθούν στο .NET: από console εφαρμογές και εφαρμογές με γραφικές διεπαφές (GUIs) μέχρι web εφαρμογές και web services σε managed αλλά και εγγενή κώδικα. Οι Premium και Ultimate εκδόσεις του Visual Studio στοχεύουν στην ομαδική ανάπτυξη μαζί με τον Team Foundation Server (TFS) που χρησιμοποιείται για την παρακολούθηση ενός έργου .NET, τη διαχείριση του κώδικα, τη διαχείριση των αναφορών αλλά και το διαμοιρασμό κώδικα, τεκμηριώσεων ή άλλων εγγράφων.

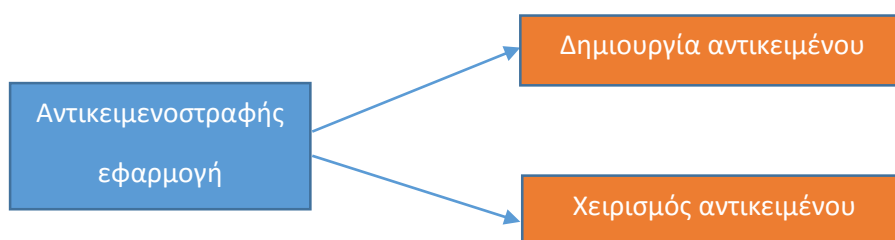
## 9. ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Η ανάπτυξη του κώδικα της εργασίας, είναι εξολοκλήρου βασισμένη στην έννοια του αντικειμενοστραφούς προγραμματισμού. Ο αντικειμενοστραφής προγραμματισμός γεννήθηκε και άρχισε να αναπτύσσεται όταν πλέον ήταν φανερό ότι οι παραδοσιακές προσεγγίσεις στον προγραμματισμό δεν μπορούσαν να ανταποκριθούν στις νέες απαιτήσεις ανάπτυξης προγραμμάτων.

Καθώς τα προγράμματα μεγάλωναν, γίνονταν υπερβολικά πολύπλοκα και έτσι διαπιστώθηκε ότι υπήρχαν αδυναμίες με τη χρήση διαδικαστικών γλωσσών προγραμματισμού. Η κυριότερη αιτία είναι ότι οι διαδικαστικές γλώσσες ( πχ PASCAL, C, κλπ), δίνουν έμφαση στις ενέργειες που πρέπει να εκτελέσει ένα πρόγραμμα. Η αντικειμενοστραφής σχεδίαση εκλαμβάνει σαν πρωτεύοντα δομικά στοιχεία ενός προγράμματος τα δεδομένα, από τα οποία δημιουργούνται, με κατάλληλη μορφοποίηση τα **αντικείμενα** (objects). Αυτή η σχεδίαση αποδείχθηκε ότι επιφέρει καλύτερα αποτελέσματα, αφού τα προγράμματα που δημιουργούνται είναι περισσότερο ευέλικτα, επαναχρησιμοποιήσιμα και φιλικά. Οι συναρτήσεις ενός αντικειμένου που στην C# ονομάζονται συναρτήσεις-μέλη (member function), αποτελούν συνήθως το μόνο τρόπο για να προσπελάσουμε τα δεδομένα ενός αντικειμένου. Τα δεδομένα είναι κρυμμένα και έτσι είναι ασφαλή από κάποια τυχαία αλλοίωσή τους. Τα δεδομένα και οι συναρτήσεις τους θεωρούνται ενθυλακωμένα (encapsulated) σε μία ανεξάρτητη οντότητα.

Τα αντικείμενα είναι βασικά στοιχεία στον αντικειμενοστραφή προγραμματισμό τα οποία έχουν χαρακτηριστικά αλλά και συμπεριφορές.

- Με τον όρο **χαρακτηριστικό** εννοούμε οτιδήποτε καθορίζει την φυσική υπόσταση του αντικειμένου. Τα χαρακτηριστικά είναι τα δεδομένα-μέλη της κλάσης τα οποία τα κρύβουμε με ειδικές τεχνικές, για να είναι ασφαλή από τυχόν αλλοίωσή τους.
- Με τον όρο **συμπεριφορά** το πώς συμπεριφέρεται ένα αντικείμενο από μόνο του ή όταν μία ενέργεια επιδράσει επάνω του. Οι συμπεριφορές είναι οι συναρτήσεις-μέλη της κλάσης οι οποίες χειρίζονται με κατάλληλο τρόπο τα δεδομένα-μέλη της κλάσης. Οι συναρτήσεις-μέλη λέγονται αλλιώς και μέθοδοι μιας κλάσης.



Η **κλάση** είναι ένα σύνολο από παρόμοια αντικείμενα.

- Η κλάση είναι το πρότυπο, δηλαδή καθορίζει τα δεδομένα (χαρακτηριστικά) και τις συναρτήσεις (συμπεριφορές) που θα έχουν τα αντικείμενα της κλάσης.
- Άρα όλα τα αντικείμενα είναι πιστά αντίγραφα της κλάσης στην οποία ανήκουν.
- Η κλάση κατά τον ορισμό της δεν δημιουργεί αντικείμενα. Τα αντικείμενα δημιουργούνται από άλλους παράγοντες μέσα στο πρόγραμμα (άλλες συναρτήσεις, δομές κλπ).

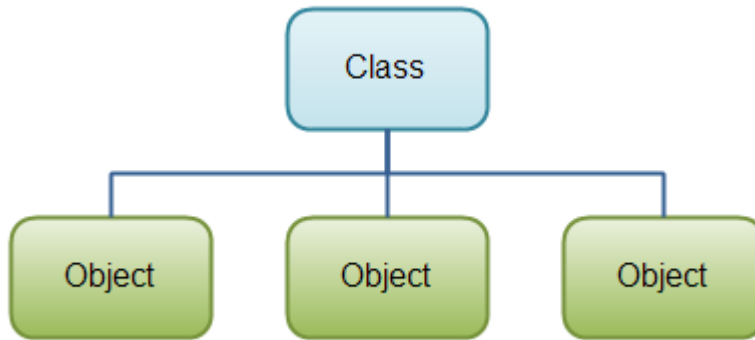
Η έννοια της κλάσης στον αντικειμενοστραφή προγραμματισμό είναι άρρηκτα συνδεδεμένη και με την έννοια της **κληρονομικότητας**. Κληρονομικότητα ονομάζεται η διαδικασία που επιτρέπει μία κλάση να κληρονομεί τις ιδιότητες και την συμπεριφορά μιας άλλης κλάσης.

- Η κλάση που κληρονομείται λέγεται βασική κλάση (base class) ενώ,
- Η κλάση που κληρονομεί λέγεται παραγόμενη κλάση (derived class).

Για παράδειγμα η κλάση οχήματα χωρίζεται σε επιβατικά, φορτηγά, λεωφορεία κλπ.

Η παραγόμενη κλάση κληρονομεί χαρακτηριστικά και συμπεριφορές από την βασική κλάση αλλά μπορεί να έχει και δικά της. Αν γυρίσουμε στο προηγούμενο παράδειγμα όλα τα οχήματα (βασική κλάση) έχουν κινητήρα και τροχούς (χαρακτηριστικά), ξεκινούν και σταματούν (συμπεριφορά), αλλά τα φορτηγά έχουν καρότσα (χαρακτηριστικό) η οποία ανατρέπεται (συμπεριφορά) και τα λεωφορεία έχουν πολλά καθίσματα (χαρακτηριστικό).





Δύο βασικά χαρακτηριστικά στον αντικειμενοστραφή προγραμματισμό είναι η **επαναχρησιμοποίηση** και ο **πολυμορφισμός**. Η επαναχρησιμοποίηση είναι η δυνατότητα χρησιμοποίησης μιας κλάσης όταν οριστεί από άλλα προγράμματα. Είναι κάτι δηλαδή σαν τις βιβλιοθήκες συναρτήσεων της γλώσσας προγραμματισμού όπου μπορούμε να τις χρησιμοποιήσουμε σε πολλά προγράμματα. Επίσης η κληρονομικότητα είναι μια μορφή επαναχρησιμοποίησης αφού μπορούμε να πάρουμε μια κλάση και χωρίς να την τροποποιήσουμε να παράξουμε μία άλλη κλάση προσθέτοντας επιπλέον χαρακτηριστικά και μεθόδους. Η νέα κλάση κληρονομεί χαρακτηριστικά και μεθόδους από την βασική αλλά χρησιμοποιεί και τα δικά της.

Πολυμορφισμός είναι η δυνατότητα που δίνεται:

1. Σε ένα αντικείμενο να αναφέρεται σε διαφορετικές κλάσεις (από αυτή από την οποία δημιουργήθηκε) και να επιδρά διαφορετικά σε άλλα αντικείμενα.
2. Σε μία συνάρτηση να υπάρχει πολλές φορές μέσα σε μία κλάση με το ίδιο όνομα και τον ίδιο σκοπό, αλλά με διαφορετικά αποτελέσματα. Για παράδειγμα η συνάρτηση πάτησε() λειτουργεί διαφορετικά στο φρένο, διαφορετικά στο γκάζι και διαφορετικά στον συμπλέκτη του αυτοκινήτου.
3. Αλλά και σε έναν τελεστή να λειτουργεί διαφορετικά μέσα σε ένα πρόγραμμα.

## 10. Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C#

Στα τέλη της δεκαετίας του '70 το μέγεθος των έργων που υλοποιούσαν οι προγραμματιστές με τη C έφτασε στα όρια του. Έτσι εμφανίστηκε ένας νέος τρόπος προγραμματισμού, ο αντικειμενοστραφής προγραμματισμός (Object Oriented Programming - OOP) και μια νέα γλώσσα, η αντικειμενοστραφής γλώσσα C++. Στη συνέχεια, λόγω της μεγάλης διάδοσης του internet οι προγραμματιστές επιθυμούσαν να μεταφέρουν τον κώδικα τους σε διαφορετικά περιβάλλοντα. Αυτό δε μπορούσε να γίνει με τη C++, οπότε δημιουργήθηκε η αντικειμενοστραφής γλώσσα προγραμματισμού Java, η οποία έλυνε δύο βασικά προβλήματα :

1. **Μεταφερσιμότητα:** μετάφραση του πηγαίου κώδικα στην ενδιάμεση γλώσσα bytecode και στη συνέχεια εκτέλεση αυτού του κώδικα στην εικονική μηχανή της Java (Java Virtual Machine - JVM).

2. ο κώδικας μπορούσε να εκτελεστεί σε οποιοδήποτε περιβάλλον διαθέτε εικονική μηχανή Java.

3. **ασφάλεια:** επειδή ο JVM τρέχει το κώδικα bytecode, έχει τον πλήρη έλεγχο του προγράμματος έτσι ώστε να μπορεί να αποτρέψει οποιαδήποτε κακόβουλη ενέργεια ενός Java προγράμματος.

Αν και η Java έλυσε 2 βασικά προβλήματα της λείπουν 2 βασικά χαρακτηριστικά, τα οποία συναντάμε στη C# και είναι τα εξής:

1. **Διαγλωσσική διαλειτουργικότητα:** για η δημιουργία μεγάλων κατανεμημένων συστημάτων λογισμικού χρειάζεται ο κώδικας που παράγεται από μια γλώσσα να συνεργάζεται με κώδικα που παράγει μια δεύτερη γλώσσα.

2. **Πλήρης ενσωμάτωση με τα Windows:** η Java, αν και μπορεί να εκτελεστεί κάτω από τα Windows, δεν συνδέεται στενά με αυτά, σε αντίθεση με τη C#.

Η C# είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού, συνεπώς είναι υψηλού επιπέδου. Αυτό σημαίνει πως είναι φιλική προς τον προγραμματιστή, αφού αποκρύπτει τις λεπτομέρειες του υλικού του υπολογιστικού συστήματος, και παρέχει έναν ενιαίο τρόπο επικοινωνίας προς αυτό, μέσω των έτοιμων κλάσεων και βιβλιοθηκών που προσφέρει. Αποτελεί μια συνεχώς εξελισσόμενη γλώσσα, και σε κάθε νέα της έκδοση προστίθενται νέα χαρακτηριστικά και συντακτικό, με στόχο να κάνει τα πράγματα ευκολότερα για τον προγραμματιστή. Ένα σημαντικό σημείο της γλώσσας προγραμματισμού C# είναι ότι ο κώδικάς της είναι “διαχειρίσιμος” (managed), επομένως απαιτεί την ύπαρξη της βιβλιοθήκης .NET και του CLR. Έτσι ενώ εκτελείται μία εφαρμογή, το CLR ελέγχει και διαχειρίζεται τη μνήμη, την εκτέλεση της συλλογής των απορριμμάτων (garbage collection), χειρίζεται τις εξαιρέσεις και παρέχει πολλές υπηρεσίες και που για τις οποίες οι προγραμματιστές δεν χρειάζεται να γράψουν κώδικα. Επίσης ο compiler παράγει Intermediate Language (IL) αντί για γλώσσα μηχανής καθώς το CLR “καταλαβαίνει” IL. Όταν το CLR δει τον κώδικα τότε ο μεταγλωττιστής JIT μεταγλωττίζει τμήμα-τμήμα τον κώδικα και τον εκτελεί.

Με την γλώσσα προγραμματισμού C# έχουμε την δυνατότητα να γράψουμε desktop εφαρμογές μέσω του Windows Presentation Foundation (WPF) και Console applications. Ακόμη, μέσω του ASP.NET μπορεί να ενσωματωθεί κώδικας σε C# ανάμεσα σε τυπικό κώδικα HTML με ξεκάθαρο και αποτελεσματικό τρόπο, βοηθώντας

την κατασκευή δυναμικών ιστοσελίδων. Μέσω των ADO.NET Entity Framework και LINQ, έχουμε πρόσβαση σε δεδομένα.

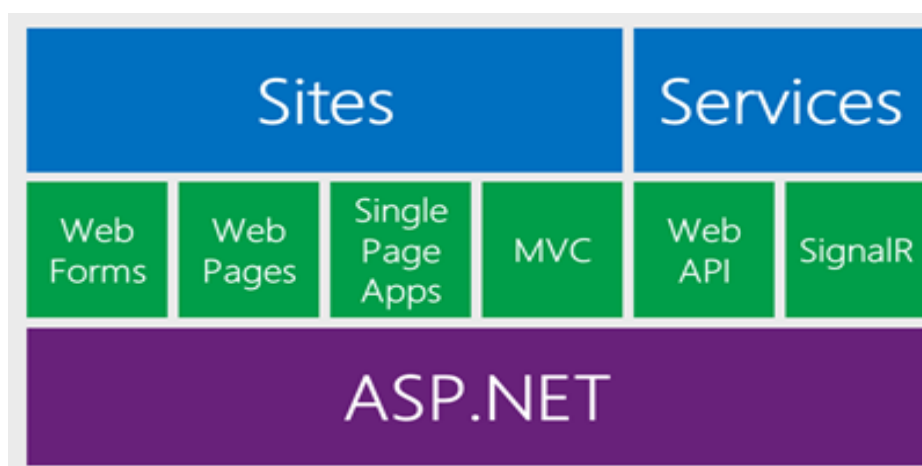
Εν συντομία, τα βασικά χαρακτηριστικά της είναι:

- **Απλότητα (Simplicity).** Η C# είναι μια απλή γλώσσα που μπορεί να χρησιμοποιηθεί χωρίς εντατική εκμάθηση, ενώ ταυτόχρονα είναι εναρμονισμένη με σύγχρονες προγραμματιστικές πρακτικές. Οι θεμελιώδεις αρχές της γλώσσας μπορούν να κατανοηθούν γρήγορα κάτι που σημαίνει ότι οι προγραμματιστές θα είναι παραγωγικοί σε σύντομο χρονικό διάστημα. Η C# έχει σχεδιαστεί έτσι ώστε να μειώνεται η πιθανότητα πρόκλησης λαθών από την πολυπλοκότητα του κώδικα, αφού τη μειώνει σε μεγάλο βαθμό με το απλουστευμένο συντακτικό της και την οργάνωση κώδικά της.
- **Αντικειμενοστρέφεια (Object – Orientation).** Η C# από τα θεμέλια της σχεδιάστηκε να είναι αντικειμενοστραφής. Ο αντικειμενοστραφής προγραμματισμός επικράτησε σαν προγραμματιστικό πρότυπο την προηγούμενη δεκαετία και παραμένει στις πρώτες προτιμήσεις των προγραμματιστών. Οι ανάγκες για κατανεμημένα συστήματα πελάτη - εξυπηρετητή συμπίπτουν με την ενθυλάκωση και την ανταλλαγή μηνυμάτων που είναι βασικά χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού. Κατά πολλούς ειδικούς στις γλώσσες προγραμματισμού, η επιτυχής λειτουργία των προγραμματιστικών συστημάτων σε δικτυακά περιβάλλοντα αυξανόμενης πολυπλοκότητας βασίζεται στην Αντικειμενοστρέφεια. Η C# παρέχει μια ξεκάθαρη και αποδοτική αντικειμενοστραφή πλατφόρμα παρέχοντας στους προγραμματιστές μια συλλογή βιβλιοθηκών δοκιμασμένων αντικειμένων που παρέχουν λειτουργικότητα που ποικίλει από απλούς τύπους δεδομένων, σε διεπαφές εισόδου/εξόδου ή δικτυακές και εργαλεία για τη δημιουργία παραθυρικών εφαρμογών. Αυτές οι βιβλιοθήκες μπορούν να προσαρμοστούν στις ανάγκες του προγραμματιστή. Επιπρόσθετα η C# υποστηρίζει και τον προγραμματισμό βασισμένο σε components (component – based programming) ο οποίος επιτρέπει τον προσδιορισμό αυτόνομων μονάδων λειτουργικότητας (components) που είναι απομονωμένα και τεκμηριωμένα, παρουσιάζοντας ένα μοντέλο με ιδιότητες, μεθόδους, events και μεταδεδομένα για το component. Η C# υποστηρίζει αυτά τα χαρακτηριστικά άμεσα κάνοντας έτσι τη διαδικασία δημιουργίας και χρήσης των components πολύ εύκολη.
- **Οικειότητα (Familiarity).** Στο ξεκίνημα της υλοποίησής της, οι δημιουργοί της τεχνολογίας C# απέρριψαν την ολοκληρωτική χρήση της C++ σαν γλώσσα υλοποίησης. Στη νέα γλώσσα ωστόσο, κράτησαν αρκετά χαρακτηριστικά της C++ αλλά και της Java και αφαίρεσαν την άχρηστη πολυπλοκότητα και των δύο. Έτσι έχοντας κρατήσει αρκετά από τα αντικειμενοστραφή χαρακτηριστικά και τη γενική φιλοσοφία της C++ αλλά και τη γενική ευκολία της Java βελτιώνοντας ορισμένα σημεία της, είναι σχετικά εύκολη τη «μετακόμιση» στη C# δεδομένου ότι η C η C++ αλλά και η Java είναι ευρέως γνωστές και χρησιμοποιούνται συχνά.

## 11. ASP.NET

Η εφαρμογή μας έχει αναπτυχθεί με την βοήθεια της ASP.NET. Asp.net είναι ένα framework της Microsoft το οποίο μας επιτρέπει να φτιάχνουμε δυναμικές ιστοσελίδες με την γλώσσα προγραμματισμού ASP. Είναι καθαρά και μόνο για web development κι όχι για ανάπτυξη εφαρμογών για υπολογιστές desktop. Το framework της ASP.NET μας δίνει τη δυνατότητα να φτιάξουμε διάφορα δυναμικά πράγματα για μία ιστοσελίδα, εύκολα και γρήγορα, ύστερα από διάφορα εργαλεία που μας παρέχει. Οι web εφαρμογές που αναπτύσσουμε φιλοξενούνται σε server που θα πρέπει να έχει Microsoft Windows και τον IIS που θα είναι υπεύθυνος να εκτελεί τα scripts μας. Ως βάσει έχει τη γλώσσα προγραμματισμού ASP και φυσικά τρέχει server side που σημαίνει ότι εκτελείται στον server και ο επισκέπτης της ιστοσελίδας βλέπει μόνο το αποτέλεσμα που παρήγαγε.

Asp είναι μία γλώσσα προγραμματισμού της Microsoft που ανταγωνίζεται την PHP αλλά χάνει συνεχώς έδαφος. Βγαίνει από τα αρχικά των λέξεων Active Server Pages και την εκτελεί μία εφαρμογή που βρίσκεται στον server που ονομάζεται IIS. Η asp δεν χρησιμοποιείται για desktop εφαρμογές (για προγράμματα στον υπολογιστή) αλλά για web εφαρμογές (ιστοσελίδες και online προγράμματα). Το μειονέκτημα είναι ότι για να προγραμματίσουμε σε αυτήν, θα πρέπει να φιλοξενούμε τα scripts μας σε server που έχει Microsoft Windows που είναι πιο ακριβός σε σχέση με έναν Linux server. Είναι και αυτή όπως και η PHP server side language σε αντίθεση πχ με την Javascript που είναι client side. Συνεργάζεται με την MSSql και μπορούμε να σχεδιάσουμε μία δυναμική ιστοσελίδα ώστε ο χρήστης κάνοντας μερικά click κάπου, να αλλάζουν πολλά πράγματα δυναμικά. Τα τελευταία χρόνια έχει κερδίσει έδαφος το framework της Microsoft με την ονομασία ASP.NET και πλέον χρησιμοποιούμε αυτό για να κάνουμε οτιδήποτε ευκολότερα και γρηγορότερα.



Το αρχείο που περιέχει μια ASP.NET σελίδα έχει την κατάληξη .aspx. Αυτά τα αρχεία για να “τρέξουν” απαιτούν την ύπαρξη ενός ειδικού λογισμικού στον server ο

οποίος ονομάζεται web-server. Ο web-server που χρησιμοποιείται για να εκτελεστούν τα aspx αρχεία σε περιβάλλον Windows είναι ο Internet Information Server (IIS). Στον web-server υπάρχει ένα μικρό κομμάτι λογισμικού το οποίο αναλαμβάνει να εκτελέσει τον κώδικα σε asp.net. Ένα aspx αρχείο μπορεί να περιέχει κώδικα σε HTML, CSS, Javascript, ASP αλλά και κώδικα στην γλώσσα .NET την οποία έχουμε επιλέξει για να προγραμματίσουμε την εφαρμογή στο κομμάτι που αφορά τον server. Γενικά μπορούμε να χωρίσουμε το αρχείο αυτό σε 2 τμήματα. Το πρώτο είναι αυτό το οποίο περιέχει τον κώδικα τον οποίο θα εκτελεστεί τοπικά στον περιηγητή. Το δεύτερο μέρος είναι ο κώδικας ο οποίος θα εκτελεστεί στον server. Συνοπτικά η λειτουργία της ASP.NET τεχνολογίας είναι η εξής:

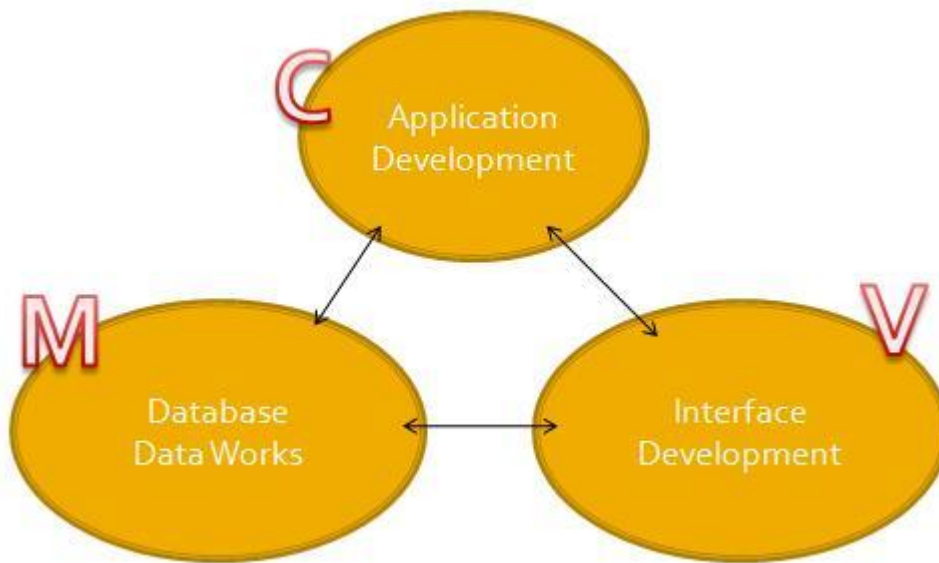
- Όταν ο περιηγητής ζητήσει ένα αρχείο HTML, ο server στέλνει το αρχείο αυτό.
- Όταν ο περιηγητής ζητήσει ένα ASP.NET αρχείο, ο IIS περνά την αίτηση στη μηχανή του server που επεξεργάζεται το ASP.NET κομμάτι.
- Αυτή η ASP.NET μηχανή διαβάζει το αρχείο, γραμμή προς γραμμή, και εκτελεί τα scripts που περιέχονται μέσα στο αρχείο.
- Τέλος, το ASP.NET αρχείο επιστρέφεται στον περιηγητή ως HTML αρχείο.

## 12. ΤΟ ΠΡΟΤΥΠΟ MVC

Στον αντικειμενοστραφή προγραμματισμό είναι σχεδόν απαραίτητο ο κώδικας που γράφουμε να είναι καλά σχεδιασμένος και οργανωμένος. Έτσι και η εργασία μας έχει αναπτυχθεί και σχεδιαστεί πάνω σε μία από τις πιο γνωστές μεθόδους σχεδίασης κώδικα, την μέθοδο MVC. Η μεθοδολογία σχεδίασης MVC, ουσιαστικά ξεχωρίζει τις διάφορες πτυχές ενός τμήματος του λογισμικού. Ο διαχωρισμός αυτός δίνει μεγάλη έμφαση στον επαναχρησιμοποιήσιμο κώδικα καθώς και σε μία πιο δομημένη αρχιτεκτονική της εφαρμογής. Αυτό έχει ως συνέπεια την γρηγορότερη ανάπτυξη εφαρμογών που χρησιμοποιούν διεπαφή με χρήστη (user interface).

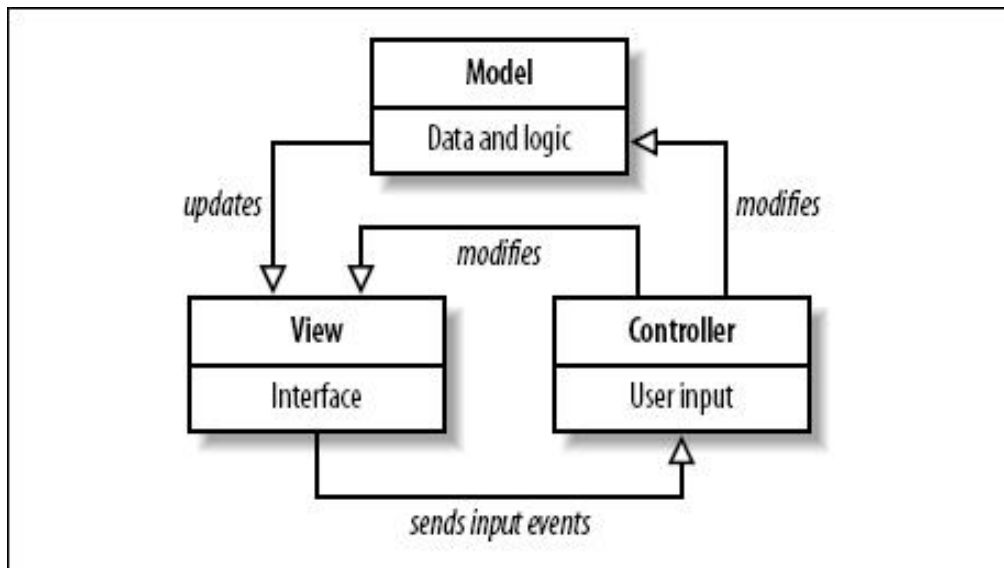
Ο όρος MVC προέρχεται από τα αρχικά των λέξεων Model – View – Controller, και προωθεί τρία δομικά στοιχεία για την ανάπτυξη της μεθοδολογίας αυτής:

1. Το πρώτο είναι το **Model**, στο οποίο τοποθετούμε τις λειτουργίες της εφαρμογής που σχετίζονται με την πρόσβαση στη βάση δεδομένων. Οι λειτουργίες αυτές είναι με τη μορφή function(μεθόδων στον προγραμματισμό). Είναι κάποιες συναρτήσεις με τις οποίες εκτελούμε διάφορες λειτουργίες διαχείρισης των δεδομένων που λαμβάνουμε από τη βάση. Για παράδειγμα αν θέλουμε σε μία σελίδα να εμφανίσουμε κάποια βιβλία από τη βάση δεδομένων, το πρώτο βήμα είναι ότι στο model των βιβλίων υπάρχει κάποια function για παράδειγμα «getAllBooks()» η οποία περιέχει κώδικα που μιλάει με τη βάση και τραβάει τα δεδομένα που θέλουμε.
2. Έπειτα υπάρχει το **View**, μέσα στο οποίο υπάρχει το HTML της σελίδας της εφαρμογής μας. Είναι αυτό που βλέπουμε. Τις περισσότερες φορές ένα View “μιλάει” με ένα controller και αφού ο controller κάνει τις διάφορες επεξεργασίες των δεδομένων στέλνει στη View συγκεκριμένα δεδομένα να εμφανίσει.
3. Τέλος υπάρχει ο **Controller**, ο οποίος είναι ο μεσίτης μεταξύ Model και της View. Ελέγχει το πώς «τρέχει» η εφαρμογή. “Μιλάει” με το Model, παίρνει τα δεδομένα που ζητά και εν συνεχεία και αφού τα επεξεργαστεί, τα στέλνει πίσω στη View για απεικόνιση.



*Εικόνα 2 Αρχιτεκτονική Model-View-Controller*

Για παράδειγμα, ένας χρήστης συμπληρώνει μία φόρμα σε ένα site και πατάει το κουμπί submit. Στη συνέχεια ο controller έχοντας λάβει το input του χρήστη “μιλάει” με το model χρησιμοποιώντας το input του χρήστη σαν μεταβλητή και ζητάει δεδομένα από το model τα οποία όταν τα λαμβάνει τα προσαρμόζει ανάλογα με αυτό που ζήτησε ο χρήστης. Στη συνέχεια ο controller με τα νέα δεδομένα πλέον, αλλάζει το view. Έτσι εάν σε ένα application βιβλιοπωλείου ο χρήστης συμπληρώσει μία search form για βιβλία με συγκεκριμένη κατηγορία ο αντίστοιχος controller θα “μιλήσει” με κάποια function του model που θα ζητάει όλα τα βιβλία και θα δέχεται ως παράμετρο την κατηγορία. Το model θα βρίσκει τα βιβλία αυτά και μέσω της χρήσης της function στο controller, αυτός θα τα εμφανίζει στην οθόνη.



Εικόνα 3 Παράδειγμα MVC

## Θετικά Αρχιτεκτονικής MVC

- Διαχωρισμός Προβλημάτων (Separation of Concerns).** Αυτό είναι και το πιο βασικό πλεονέκτημα του MVC. Ουσιαστικά δημιουργείται μία εφαρμογή η οποία έχει τρία επίπεδα, το επίπεδο των Models, το επίπεδο των Controllers και το επίπεδο των Views -που θα αναλυθούν παρακάτω- και το κάθε επίπεδο επιτελεί ξεχωριστό έργο και ταυτόχρονα συνεργάζεται με τα άλλα επίπεδα. Μία σωστή MVC εφαρμογή είναι εκείνη που τα τρία επίπεδα είναι ξεκάθαρα καθορισμένα και δεν συμπλέκονται. Για παράδειγμα είναι λάθος στο επίπεδο των View να υπάρχει κώδικας που “μιλάει” με την βάση δεδομένων και “τραβάει” δεδομένα.
- Επεκτασιμότητα.** Το δεύτερο πλεονέκτημα της MVC αρχιτεκτονικής είναι πολύ εξίσου σημαντικό. “Επεκτασιμότητα” είναι η δυνατότητα που διαθέτει μία εφαρμογή, κατά την οποία μπορούμε μελλοντικά να προσθέσουμε λειτουργίες σε αυτή ή να αλλάξουμε κάποιες από τις ήδη υπάρχουσες λειτουργίες και να έχουμε άλλα αποτελέσματα. Για να το δούμε εντελώς απλά και κατανοητά, η πλατφόρμα WordPress είναι επεκτάσιμη με τη χρήση των διάφορων plugins διότι προσθέτουμε στις ήδη υπάρχουσες λειτουργίες και άλλες λειτουργίες. Τα προγράμματα που είναι φτιαγμένα με MVC αρχιτεκτονική έχουν βασικό χαρακτηριστικό ότι είναι επεκτάσιμα. Είναι ίσος το σημαντικότερο για μία εφαρμογή να μπορεί να επεκτείνεται εύκολα και χωρίς να δημιουργεί η εκάστοτε αλλαγή προβλήματα σε άλλες συνιστώσες από την δεδομένη που αλλάζει ή επεκτείνεται.
- Ελεγχιμότητα (Testability).** Αυτό είναι ένα πολύ κρίσιμο χαρακτηριστικό. Οι MVC εφαρμογές έχουν την δυνατότητα να είναι ελέγξιμες και με τον τρόπο



αυτό συντηρούνται πιο εύκολα. Ας δώσουμε ένα απλό παράδειγμα. Έστω ότι έχουμε μία εφαρμογή η οποία διαθέτει μία λειτουργία Login - σύνδεσης, δηλαδή ζητά από τον χρήστη να πληκτρολογήσει κάποια στοιχεία σε μία φόρμα και εν συνέχεια αν ταυτοποιηθεί, τον εισάγει μέσα στο σύστημα. Αυτή τη λειτουργία την ελέγχει κάποιος Login Controller ο οποίος περιέχει κώδικα που διαχειρίζεται τα δεδομένα αυτά που εισήχθησαν από τον χρήστη. Αυτός ο Controller θεωρείται μία “μονάδα” ή αλλιώς Unit. Στα MVC Frameworks μπορούμε με πολλή ευκολία να γράψουμε έναν απλό κώδικα - tests με τον οποίο τεστάρουμε αυτόν τον Controller αλλά και κάθε μία από τις λειτουργίες του. Παίρνουμε τα αποτελέσματα και βλέπουμε αν η συγκεκριμένη μονάδα της εφαρμογής μας λειτουργεί σωστά. Σε online συστήματα που πολλές φορές απευθύνονται σε χιλιάδες χρήστες είναι πολύ κρίσιμο να γνωρίζουμε πριν την διανομή του κώδικά μας εάν είναι σωστά υλοποιημένα όλα τα υποσυστήματα μας. Στις διαδικτυακές εφαρμογές ο έλεγχος είναι μέγιστης σημασίας.

- **“Καθαρά” URLs.** Τα περισσότερα MVC Frameworks για web applications – διαδικτυακές εφαρμογές δίνουν τη δυνατότητα να έχουμε “καθαρά” urls. Καθαρά URLs σημαίνει ότι οι διευθύνσεις που θα εμφανίζονται στον Browser του επισκέπτη καθώς αυτός θα περιηγείται στην εφαρμογής μας, θα βγάζουν νόημα και θα είναι γραμμένα σε φυσική γλώσσα. Το γεγονός αυτό θα εξηγήσουμε σε επόμενο κεφάλαιο γιατί είναι ωφέλιμο. Στο σημείο αυτό θα προχωρήσουμε σε μία ανάλυση του MVC Framework που χρησιμοποιήθηκε για τον σκοπό αυτής της εργασίας. Αυτό είναι το CakePHP ένα από τα πιο διαδεδομένα PHP Frameworks και ταυτόχρονα απλό και εύχρηστο.

## 13. ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΠΕΡΙΒΑΛΛΟΝ MySQL

### Εισαγωγή

Μια Βάση Δεδομένων (Database) είναι ένας οργανωμένος τρόπος αποθήκευσης πληροφοριών και πρόσβασης σε αυτές. Μια βάση δεδομένων είναι κάτι παραπάνω από μια απλή συλλογή αποθηκευμένων στοιχείων. Ένας άλλος ορισμός είναι ότι μια βάση δεδομένων είναι ένα ολοκληρωμένο σύστημα που αποτελείται από δεδομένα και από το κατάλληλο λογισμικό, τα οποία χρησιμοποιώντας το υλικό (hardware) βοηθούν στην ενημέρωση και πληροφόρηση των χρηστών. Ένα πρόγραμμα που διαχειρίζεται βάσεις δεδομένων αποκαλείται Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS, Database Management System) και με την βοήθειά του μπορούμε να αποθηκεύσουμε, προσθέσουμε, τροποποιήσουμε, εμφανίσουμε ή και διαγράψουμε τα αποθηκευμένα δεδομένα. Τα δεδομένα που υπάρχουν στις βάσεις δεδομένων πρέπει να είναι:

- Ολοκληρωμένα (Integrated), δηλαδή τα δεδομένα πρέπει να είναι αποθηκευμένα σε ομοιόμορφα οργανωμένα σύνολα αρχείων όπου δεν πρέπει να υπάρχει επανάληψη ή πλεονασμός (redundancy) των ίδιων στοιχείων.
- Καταμεριζόμενα (Shared), δηλαδή να μπορούν περισσότεροι του ενός χρήστες να βλέπουν και να μοιράζονται τα ίδια δεδομένα την ίδια χρονική στιγμή.

Το σχεσιακό μοντέλο περιγράφει τη Βάση Δεδομένων και οργανώνει τις εγγραφές με βάση τις σχέσεις. Γι' αυτό το λόγο μια Βάση δεδομένων σχεδιασμένη με βάση το σχεσιακό μοντέλο, μπορεί εύκολα να υλοποιηθεί με ένα μοντέλο Οντοτήτων - Συσχετίσεων. Στις σχεσιακές Βάσεις Δεδομένων, οι εγγραφές οργανώνονται σε πίνακες. Οι πίνακες σε μια σχεσιακή Βάση Δεδομένων, αποτελούνται από μια ή περισσότερες στήλες που αντιστοιχούν σε τιμές πεδίων (ή στα χαρακτηριστικά για τα μοντέλα Οντοτήτων - Συσχετίσεων) και από γραμμές που αντιστοιχούν σε εγγραφές για αυτά τα πεδία.

Το Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS, Database Management System) είναι ένα σύνολο από “προγράμματα” που επιτρέπουν τον χειρισμό των δεδομένων μιας ή περισσότερων βάσεων δεδομένων που ανήκουν στο ίδιο σύστημα. Το DBMS περιέχει κάποια εργαλεία γενικής χρήσης για να μπορούμε να δημιουργούμε και να χειριζόμαστε τα δεδομένα. Στα νεότερα DBMS μπορούμε να έχουμε άμεση πληροφόρηση χωρίς να απαιτείται η παρουσία ενός προγραμματιστή. Τα δεδομένα ενός DBMS μπορούν να χρησιμοποιηθούν σε κάθε μορφής ερώτημα (query) ώστε να αντλήσουμε τις πληροφορίες θέλουμε.

# Η γλώσσα MySQL

Η MySQL είναι ίσως η δημοφιλέστερη ανοικτή εφαρμογή διαχείρισης βάσεων δεδομένων (Database). Η MySQL αναπτύσσεται, διανέμεται, και υποστηρίζεται από την εταιρεία MySQL AB (<http://www.mysql.com>), η οποία είναι μια εμπορική επιχείρηση ανάπτυξης ανοικτού λογισμικού. Ανοικτό λογισμικό, σημαίνει ότι είναι δυνατό για οποιονδήποτε να χρησιμοποιήσει και να τροποποιήσει το λογισμικό, χωρίς καμία οικονομική επιβάρυνση. Το λογισμικό MySQL υπόκειται στους περιορισμούς της άδειας ευρέος κοινού (General Public Licence– GPL), ενώ σχετική άδεια απαιτείται για τη χρήση του σε εμπορικές εφαρμογές, οι όροι της οποίας περιγράφονται στη διεύθυνση <http://www.mysql.com/company/legal/licensing/>.

Η SQL (Structured Query Language – Δομημένη Γλώσσα Ερωτήσεων) είναι το πρότυπο γλώσσας μέσω της οποίας χειριζόμαστε ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ). Η SQL είναι μια πλήρης γλώσσα βάσεων δεδομένων που διαθέτει εντολές για ορισμό δεδομένων, ερωτήσεις και ενημερώσεις. Επομένως μπορούμε να διακρίνουμε δύο υποσύνολα εντολών στην SQL, τις εντολές που αφορούν τον ορισμό και τη διαχείριση της δομής της βάσης (γλώσσα ορισμού δεδομένων) και τις εντολές που αφορούν την διαχείριση των ίδιων των δεδομένων (γλώσσα χειρισμού δεδομένων). Είναι ενδιαφέρον να σημειώσουμε πως τα περισσότερα ΣΔΒΔ έχουν κάνει προσθήκες πάνω στην standard SQL, δηλαδή επιπρόσθετες εντολές στο πρότυπο σύνολο εντολών της SQL που υποστηρίζονται μόνο από το συγκεκριμένο ΣΔΒΔ. Έτσι μπορούμε να αξιοποιήσουμε πλήρως όλες τις δυνατότητες που μας παρέχει το ΣΔΒΔ και να εξειδικεύσουμε τη σχεδίαση της βάσης ανάλογα με τις λεπτομέρειες υλοποίησης του εκάστοτε ΣΔΒΔ.

## Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)

Εφόσον τα δεδομένα σε μια (σχεσιακή) βάση δεδομένων αποθηκεύονται σε πίνακες, η ΓΟΔ αναλαμβάνει την δημιουργία και διαχείριση των πινάκων. Αυτό περιλαμβάνει τον προσδιορισμό του τύπου δεδομένων και των ιδιοτήτων των πεδίων του πίνακα, αλλά και επιπρόσθετων ιδιοτήτων πάνω στον ίδιο τον πίνακα (π.χ. αν οι χαρακτήρες θα είναι στα ελληνικά ή τα αγγλικά, κ.ά.). Επίσης διαχειρίζεται τις ίδιες τις βάσεις μέσα στο ΣΔΒΔ, ορίζει χρήστες βάσης και τα δικαιώματά τους πάνω στους πίνακες κ.ά. Το σύνολο όλων αυτών των οντοτήτων ονομάζεται σχέδιο (scheme).

## Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)

Ο χειρισμός των δεδομένων αφορά εντολές για την ένθεση, την ενημέρωση την ανάκτηση και την διαγραφή των δεδομένων μέσα στη βάση. Έχοντας δηλαδή σχεδιάσει τους πίνακες της βάσης (δηλ. τη δομή της) μέσω της ΓΟΔ, τώρα μπορούμε με την ΓΧΔ να προσθέσουμε εγγραφές στους πίνακες ή να αναζητήσουμε δεδομένα ορίζοντας κριτήρια αναζήτησης, τα πεδία που θέλουμε να επιστραφούν, πως θέλουμε να συνδυαστούν οι πληροφορίες από τους διάφορους πίνακες κτλ. Μια εντολή SQL για αναζήτηση δεδομένων ονομάζεται query (δηλ. ερώτημα).

## Βασικά χαρακτηριστικά της MySQL

Η MySQL λειτουργεί με το μοντέλο Πελάτη/Εξυπηρετητή (Client/Server). Ο εξυπηρετητής δέχεται αιτήματα από τους πελάτες σχετικά με τη διαχείριση μίας ή περισσότερων βάσεων δεδομένων και προβαίνει στις απαιτούμενες ενέργειες.

Οι κύριες λειτουργίες διαχείρισης βάσεων δεδομένων τις οποίες υλοποιεί ο εξυπηρετητής κατόπιν αιτήσεως από τον πελάτη, δε διαφέρουν σημαντικά από τις αντίστοιχες λειτουργίες των άλλων εφαρμογών βάσεων δεδομένων και περιλαμβάνουν τη δημιουργία διαγραφή βάσεων δεδομένων, την εισαγωγή-τροποποίηση-διαγραφή πινάκων (tables) και πεδίων (fields), την εισαγωγή-τροποποίηση-διαγραφή εγγραφών (records) και τέλος την ανάκτηση δεδομένων από τη βάση με τη χρήση συγκεκριμένων κριτηρίων.

Για την υλοποίηση των παραπάνω λειτουργιών, η MySQL χρησιμοποιεί την SQL (Δομημένη Γλώσσα Ερωτήσεων – Structured Query Language). Η SQL είναι η ευρύτερα χρησιμοποιούμενη τυποποιημένη γλώσσα πρόσβασης στις βάσεις δεδομένων και αποτελείται από εντολές οι οποίες επιτρέπουν την ανάκτηση και ενημέρωση δεδομένων σε μια βάση. Σημειώνουμε ότι, εκτός από τη MySQL, η γλώσσα SQL συνεργάζεται με άλλα προγράμματα βάσεων δεδομένων όπως είναι η Access, Microsoft SQL Server, Oracle, Sybase και πολλά άλλα.

## Πλεονεκτήματα της MySQL

Μερικοί από τους κύριους ανταγωνιστές της MySQL είναι οι PostgreSQL, Microsoft SQL και Oracle. Η MySQL έχει πολλά πλεονεκτήματα, όπως χαμηλό κόστος, εύκολη διαμόρφωση και μάθηση και ο κώδικας προέλευσης είναι διαθέσιμος.

### Αξιοπιστία

Η MySQL είναι ένα μεγάλο Open Source project που αναπτύσσεται από μια μεγάλη εταιρία, και από πάρα πολλούς προγραμματιστές.

### Ασφάλεια

Απαιτείται πιστοποίηση για την είσοδο οποιουδήποτε χρήστη. Επίσης οι κωδικοί αποθηκεύονται κρυπτογραφημένοι.

### Απόδοση

Η MySQL είναι χωρίς αμφιβολία γρήγορη. Μπορείτε να δείτε την σελίδα δοκιμών <http://web.mysql.com/benchmark.html> . Πολλές από αυτές τις δοκιμές δείχνουν ότι η MySQL είναι αρκετά πιο γρήγορη από τον ανταγωνισμό.

### **Χαμηλό κόστος**

Η MySQL είναι διαθέσιμη δωρεάν , με άδεια ανοικτού κώδικα (Open Source) ή με χαμηλό κόστος , αν πάρετε εμπορική άδεια, αν απαιτείται από την εφαρμογή σας.

### **Μέγεθος βάσεις δεδομένων**

Η MySQL μπορεί να διαχειριστεί τεράστιες σε μέγεθος βάσεις δεδομένων καθώς το προεπιλεγμένο μέγεθος μιας βάσης που μπορεί να διαχειριστεί είναι 4GB το οποίο μπορεί να φτάσει έως και τα 8TB

### **Ευκολία Χρήσης**

Οι περισσότερες μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL. Αν έχετε χρησιμοποιήσει ένα άλλο σύστημα διαχείρισης βάσεων δεδομένων δεν θα έχετε πρόβλημα να προσαρμοστείτε σε αυτό.

### **Μεταφερσιμότητα**

Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα Unix όπως επίσης και στα Microsoft Windows .

### **Κώδικας Προέλευσης**

Όπως και με την PHP , μπορείτε να πάρετε και να τροποποιήσετε τον κώδικα προέλευσης της MySQL.

## **Μειονεκτήματα της MySQL**

Οι πρώτες εκδόσεις της MySQL έχαναν πολλά standards και δυνατότητες των συστημάτων RDBMS. Συνήθως δεν έπιαναν τον στόχο της εγγυημένης ταχύτητας και ακρίβειας στις εμπορικές συναλλαγές, αργότερα υπήρξαν μερικές κριτικές πάνω σ' αυτό, όσον αφορά τις μεταβολές στοιχείων και θέματα ακεραιότητας. Άλλοι λένε πως οι δυνατότητες όλο και αφήνουν κάτι απ' έξω, αλλά οι υποστηρικτές της επιμένουν πως είναι θέμα ρυθμίσεων και αρκεί να ρυθμίζεις ορισμένα πράγματα για να πετύχεις αυτό που θέλεις.

## Παραδείγματα της MySQL

Παρακάτω ακολουθούν κάποια παραδείγματα MySQL με στοιχεία από την δημιουργία της πτυχιακής μας:

### Δημιουργία Βάσης

```
CREATE DATABASE iatrio;
```

### Δημιουργία πίνακα

```
CREATE TABLE patience;
```

### Εισαγωγή στηλών

```
CREATE TABLE `patience` (  
  `patience_id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(25) DEFAULT NULL,  
  `address` varchar(45) DEFAULT NULL,  
  `phone` bigint(20) NOT NULL  
  `email` varchar(255) DEFAULT NULL
```

### Εισαγωγή στοιχείων

```
INSERT INTO `patience` VALUES (1,'Αίγλη','Αργοναυτών  
24',2421024471,'patience1@gmail.gr')
```

### Διαγραφή πίνακα

```
DROP TABLE `patience`;
```

### Διαγραφή στήλης

```
ALTER TABLE `patience` DROP `email`;
```

### Αλλαγή ιδιοτήτων

```
ALTER TABLE `patience`  
CHANGE `patience_id` `p_id` (11) NOT NULL AUTO_INCREMENT
```

## 14. APACHE SERVER

Ο Apache Web Server είναι αυτό ακριβώς που δηλώνει το όνομά του. Πρόκειται δηλαδή για έναν εξυπηρετητή (server) του παγκόσμιου Ιστού (Web). Με τον όρο server το μυαλό μας πηγαίνει ίσως σε ηλεκτρονικούς υπολογιστές που φιλοξενούν ιστοσελίδες και όχι άδικα. Ο όρος αυτός χρησιμοποιείται και για το μηχάνημα εξυπηρετητή (hardware) αλλά και για το πρόγραμμα (software). Στο σημείο αυτό θα ασχοληθούμε μόνο με το software και συγκεκριμένα με τον Apache.

Ο Apache εγκαθίσταται σε έναν υπολογιστή ο οποίος μπορεί να χρησιμοποιεί διάφορα λειτουργικά συστήματα όπως Linux, Unix, Microsoft Windows, GNU, FreeBSD, Solaris, Novell NetWare, Mac OS X, OS/2, TPF. Ο ρόλος του Apache είναι να αναμένει αιτήσεις από διάφορα προγράμματα – χρήστες (clients) όπως είναι ένας φυλλομετρητής (browser) ενός χρήστη και στη συνέχεια να εξυπηρετεί αυτές τις αιτήσεις “σερβίροντας” τις σελίδες που ζητούν είτε απευθείας μέσω μιας ηλεκτρονικής διεύθυνσης (URL), είτε μέσω ενός συνδέσμου (link). Ο τρόπος με τον οποίο ο Apache εξυπηρετεί αυτές τις αιτήσεις, είναι σύμφωνος με τα πρότυπα που ορίζει το πρωτόκολλο [HTTP \(Hypertext Transfer Protocol\)](#).

### Το ξεκίνημα και η εξέλιξη του Apache

Η περίοδος έναρξης της δημιουργίας του προγράμματος χρονολογείται στις αρχές του 1990, όταν άρχισε να αναπτύσσεται από τον Robert McCool, ως ένα project του [National Center for Supercomputing Applications \(NCSA\)](#) με το όνομα HTTPd (HTTP daemon). Το 1994 ο Robert McCool αποχώρησε από το NCSA με αποτέλεσμα το [NCSA HTTPd](#) να μείνει σχεδόν εγκαταλειμμένο, πέρα από κάποιες διορθώσεις (patches) που ανέπτυσαν και διένειμαν εκτός από τον McCool και άλλοι προγραμματιστές,. Το 1995 ανέλαβε το πρόγραμμα το [Ίδρυμα Λογισμικού Apache \(Apache Software Foundation\)](#), το οποίο διατηρεί την εποπτεία του έως και σήμερα.

### Ο Apache σήμερα

Ο Apache HTTP αναπτύσσεται από την “Κοινότητα Ανοιχτού Λογισμικού” και η εποπτεία, υποστήριξη, και διάθεση του προγράμματος γίνεται από το Apache Software Foundation. Το πρόγραμμα είναι ανοιχτού κώδικα (open source), κάτι που σημαίνει ότι σύμφωνα με την άδεια χρήσης του (license), διατίθεται δωρεάν και μπορούν να γίνουν ελεύθερα από το χρήστη προσθήκες και τροποποιήσεις στον κώδικα του.

### Η προέλευση του ονόματος του

Υπάρχουν δύο εκδοχές σχετικά με την προέλευση του ονόματος του. Η πρώτη εντοπίζεται στα πρώτα χρόνια της δημιουργίας του, τότε που ως NCSA HTTPd έπρεπε να αναπτύσσονται συνεχώς διορθώσεις (patches) για να ενσωματωθούν στον αρχικό του κώδικα με αποτέλεσμα να του δοθεί το όνομα a patchy server. Η δεύτερη εκδοχή σύμφωνα με το ίδρυμα Apache, αναφέρει ότι το όνομα αυτό δόθηκε προς τιμήν των ιθαγενών Ινδιάνων της Αμερικής και συμβολίζει το μαχητικό πνεύμα και την αντοχή.

## Χαρακτηριστικά και λειτουργίες του Apache HTTP

Ο Apache διαθέτει ποικιλία χαρακτηριστικών και μπορεί να υποστηρίξει μια μεγάλη γκάμα εφαρμογών με τις οποίες και συνεργάζεται. Οι δυνατότητες του προγράμματος αυτού καθαυτού και τα χαρακτηριστικά του δεν είναι και τόσο πολλά. Ένα από τα βασικότερα χαρακτηριστικά του όμως, το οποίο και του δίνει μεγάλες δυνατότητες, είναι ότι μπορεί να προσαρμόσει επάνω του πολλές προσθήκες προγραμμάτων (modules), τα οποία με τη σειρά τους παρέχουν διαφορετικές λειτουργίες. Μερικά από τα πιο γνωστά modules του Apache HTTP είναι τα modules πιστοποίησης, όπως για παράδειγμα τα mod\_access, mod\_auth, mod\_digest κ.λπ. Παρέχει επίσης SSL σε TLS μέσω των (mod\_ssl), και proxy module (mod\_proxy), πραγματοποιεί ανακατευθύνσεις διευθύνσεων (URL rewrites) μέσω του mod\_rewrite, καταγραφές συνδέσεων μέσω του mod\_log\_config, συμπίεση αρχείων μέσω του mod\_gzip και πολλά άλλα modules τα οποία διατίθενται είτε απ'ο το Apache Software Foundation, είτε από τρίτες εταιρίες λογισμικού.

Ένα άλλο χαρακτηριστικό – δυνατότητα του Apache HTTP, όπως έχω αναφέρω πιο πάνω, είναι ότι μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα. Ο Apache HTTP υποστηρίζει επίσης αρκετές διάσημες εφαρμογές και γλώσσες προγραμματισμού όπως [MySQL](#), [PHP](#), [Perl](#), [Python](#) κ.λπ.

Αυτά είναι μερικά από τα χαρακτηριστικά και τις λειτουργίες του που κάνουν τον Apache τον πιο δημοφιλή Web Server από το 1996 έως τις μέρες μας. Περισσότερο από το 50% των ιστοχώρων του παγκόσμιου ιστού, χρησιμοποιεί τον Apache ως εξυπηρετητή. Το υπόλοιπο ποσοστό καλύπτουν αντίστοιχα προγράμματα, όπως το Microsoft Internet Information Services (IIS), ο Sun Java System Web Server, ο Zeus Web Server κ.α.



## 15. Visual Studio

Όλες οι παραπάνω τεχνολογίες έχουν εφαρμοστεί με την βοήθεια του Visual Studio. Το Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment) όπου διατίθεται από την Microsoft και μπορεί κανείς να αναπτύξει εφαρμογές συμβατές με λειτουργικά συστήματα της Microsoft και με .NET πλατφόρμες. Με το πρόγραμμα αυτό, οι προγραμματιστές έχουν την δυνατότητα να δημιουργήσουν ιστοσελίδες (web sites), εφαρμογές διαδικτύου (web applications), διαδικτυακές υπηρεσίες (web services) καθώς και εφαρμογές κονσόλας (console applications). Επιπλέον υποστηρίζει πολλές γλώσσες προγραμματισμού με τις οποίες έχουμε την επιλογή να γράψουμε τον κώδικα μας. Αυτές είναι οι C, C++, C#, Visual Basic και J#.

Ίσως ένα από τα πιο σημαντικά χαρακτηριστικά του Visual Studio είναι η δυνατότητα που δίνεται στους προγραμματιστές για έλεγχο σφαλμάτων που πιθανόν να περιέχονται στον κώδικα (debugging). Ο μηχανισμός αυτός λειτουργεί και σε επίπεδο πηγαίου κώδικα και σε επίπεδο γλώσσας μηχανής. Επίσης μπορεί και χειρίζεται με την ίδια ευκολία κώδικες είτε είναι γραμμένοι με διαχειρίσιμο κώδικα (managed code) είτε είναι γραμμένοι σε φυσική γλώσσα (native code) καθώς και εφαρμογές που είναι γραμμένες σε γλώσσα προγραμματισμού που υποστηρίζεται από το Visual Studio. Επιπλέον μπορεί να επισυναφθεί σε μια διαδικασία οποία τρέχει και να μπορεί κάποιος να την παρακολουθήσει και να διορθώσει επιτόπου τυχόν σφάλματα. Σε περίπτωση που ο κώδικας είναι διαθέσιμος μπορούμε να τον βλέπουμε ταυτόχρονα με την διαδικασία που τρέχει, αλλιώς, σε αντίθετη περίπτωση, βλέπουμε την αντίστοιχη γλώσσα μηχανής. Επίσης ο debugger μπορεί και δημιουργεί έναν μηχανισμό συλλογής απορριμμάτων (garbage collection), τον οποίο ενεργοποιεί όταν χρειαστεί να τρέξει η διαδικασία ελέγχου σφαλμάτων. Επιπρόσθετα, ο μηχανισμός εντοπισμού σφαλμάτων μας επιτρέπει να σταματάμε το τρέξιμο του κώδικα σε ένα ορισμένο από εμάς σημείο και να παρακολουθούμε τις τιμές των τρέχων μεταβλητών που χρησιμοποιούνται στον κώδικα. Τέλος είναι δυνατή η διόρθωση του κώδικα καθώς ελέγχεται για σφάλματα.

Ένα άλλο πολύ σημαντικό χαρακτηριστικό του Visual Studio είναι από μόνο του το περιβάλλον όπου μπορεί να γράψει κανείς κώδικα (code editor). Καθώς γράφουμε κώδικά, παράλληλα τρέχει και μια λειτουργία, γνωστή ως IntelliSense, η οποία μπορεί να συμπληρώσει αυτόματα μεταβλητές, συναρτήσεις, μεθόδους, επαναληπτικούς βρόγχους και LINQ ερωτήματα. Η δυνατότητα αυτή υποστηρίζει τις γλώσσες προγραμματισμού που υποστηρίζει και το Visual Studio, XML, CSS και Javascript. Στον editor μπορούμε επίσης να προσθέσουμε κάποιον σελιδοδείκτη για να συνεχίσουμε από κει που μείναμε την τελευταία φορά αλλά και να συμπτύξουμε κάποιος μπλοκ κώδικα. Επίσης έχεις την δυνατότητα να δημιουργήσει κώδικα και να τον χρησιμοποιήσεις όπου και όποτε το επιθυμείς κάθε φορά που φτιάχνει κάποιος ένα καινούργιο έργο.

Τέλος, αξίζει να αναφέρουμε και μερικά ακόμα χαρακτηριστικά του Visual Studio τα οποία έχουν ενδιαφέρον. Η εφαρμογή περιλαμβάνει μια σειρά από οπτικούς σχεδιαστές που βοηθούν στην ανάπτυξη των εφαρμογών. Τα εργαλεία αυτά περιλαμβάνουν το Windows Forms Designer. Το Windows Forms Designer

χρησιμοποιείται για τη δημιουργία GUI εφαρμογών που χρησιμοποιεί Windows Forms. Έλεγχοι που εμφανίζουν δεδομένα (όπως κείμενο, πλαίσιο λίστας, προβολή πλέγματος, κλπ.) μπορεί να δεσμεύονται με πηγές δεδομένων, όπως βάσεις δεδομένων ή ερωτήματα. Ο σχεδιαστής δημιουργεί είτε C # ή VB.NET κώδικα για την εφαρμογή. Ο σχεδιαστής WPF, με την κωδική ονομασία Cider, εισήχθη με το Visual Studio 2008. Όπως και ο σχεδιαστής φορμών των Windows που υποστηρίζει το drag and drop. Χρησιμοποιείται για τη συγγραφή των user interfaces, με στόχο το Windows Presentation Foundation. Υποστηρίζει όλες τις λειτουργίες WPF, συμπεριλαμβανομένων των συνδέσεων δεδομένων και την αυτόματη διαχείριση της διάταξης. Παράγει XAML κώδικα για το UI. Το παραγόμενο XAML αρχείο είναι συμβατό με το Microsoft Expression Design. Ο κωδικός XAML συνδέεται με τον κωδικό χρησιμοποιώντας ένα κωδικό πίσω από το μοντέλο. Το Visual Studio περιλαμβάνει επίσης έναν επεξεργαστή website και σχεδιαστή που επιτρέπει την σχεδίαση ιστοσελίδων με μεταφορά και απόθεση widgets. Χρησιμοποιείται για την ανάπτυξη ASP.NET εφαρμογών και υποστηρίζει HTML, CSS και JavaScript. Χρησιμοποιεί έναν κώδικα για να συνδέεται με το ASP.NET. Από το Visual Studio 2008 και μετά, η διάταξη που χρησιμοποιείται από τον web designer, είναι από κοινού με το Microsoft Expression Web. Ένα άλλο σημαντικό εργαλείο που διαθέτει το πρόγραμμα είναι το designer Class, το οποίο χρησιμοποιείται από τον συγγραφέα ο οποίος μπορεί να επεξεργαστεί τις τάξεις (συμπεριλαμβανομένων των μελών της και την πρόσβασή τους) με τη χρήση του μοντέλου UML. Ο σχεδιαστής μπορεί να δημιουργήσει κώδικα κλάσης C # και VB.NET , και να ορίσει τις τάξεις και τις μεθόδους. Μπορεί επίσης να δημιουργήσει διαγράμματα κλάσεων από τις κατηγορίες χειρόγραφα. Τέλος διατίθεται ένας σχεδιαστής δεδομένων ο οποίος μπορεί να χρησιμοποιηθεί για να επεξεργαστείτε γραφικά σχήματα βάσεων δεδομένων, συμπεριλαμβανομένων των δακτυλογραφημένων πινάκων, των πρωτεύον και των ξένων κλειδιών και των περιορισμών. Μπορεί επίσης να χρησιμοποιηθεί για το σχεδιασμό ερωτημάτων από την προβολή γραφικών.

## 16. Συμπεράσματα και μελλοντικές επεκτάσεις

Η χρήση της τεχνολογίας των ηλεκτρονικών υπολογιστών στη μηχανογράφηση του ιατρικού φακέλου και τη διαχείριση των ιατρικών πληροφοριών που αφορούν στον ασθενή έχει αλλάξει σημαντικά τον τρόπο άσκησης της ιατρικής σήμερα. Τα γνωστά σε όλους μας πρακτικά μειονεκτήματα του κλασικού χάρτινου ιατρικού φακέλου και τα εμφανή πλεονεκτήματα της ηλεκτρονικής καταγραφής κι αρχειοθέτησης οδηγούν πλέον στην αλλαγή του σκηνικού. Σε όλες τις αναπτυγμένες χώρες έχει καθιερωθεί ο ηλεκτρονικός ιατρικός φάκελος, με τη διενέργεια ιατρικών τηλεδιασκέψεων για τη διάγνωση και τη λήψη θεραπευτικών αποφάσεων για ειδικά θέματα και την πραγματοποίηση σε παγκόσμια κλίμακα πολυκεντρικών μελετών κι επιδημιολογικών ερευνών, γεγονότα που θεωρούνται σχεδόν αυτονόητα. Οι Έλληνες ιατροί πρέπει να επιδείξουν ιδιαίτερο ενδιαφέρον για την μέθοδο αυτή, ενώ οι διοικητικές υπηρεσίες να επιδιώξουν άμεσα την εφαρμογή της, η οποία συνεπάγεται ελάττωση του κόστους και καλύτερη λογιστική διαχείριση του ασθενή, παρακολούθηση της λειτουργίας των τμημάτων και δυνατότητα διενέργειας ποιοτικού ελέγχου των προσφερόμενων υπηρεσιών υγείας.

Παρά το γεγονός ότι η παρούσα διπλωματική εργασία περιείχε ανάπτυξη λογισμικού, είναι σαφές ότι το ζήτημα δεν είναι δυνατόν να εξαντληθεί μέσω τις εργασίας. Κατά συνέπεια υπάρχουν αρκετά ζητήματα που θα μπορούσαν να αποτελέσουν μελέτη προς μελλοντικές επεκτάσεις. Κάποια από αυτά θα μπορούσε να ήταν η μελέτη αξιολόγησης των επιχειρησιακών αναγκών και ζητημάτων που θα πρέπει να αντιμετωπισθούν κατά την εφαρμογή ενός παρόμοιου συστήματος. Είναι ξεκάθαρο ότι με την εγκατάσταση του ηλεκτρονικού φακέλου ασθενή, σε ιατρεία και νοσοκομεία, θα επέλθουν και σημαντικές αλλαγές στην λειτουργικότητα. Γι' αυτό και θα πρέπει να εντοπισθούν αυτές οι αλλαγές, να μελετηθούν και να αντιμετωπισθούν με γνώμονα την διατήρηση του υψηλού επιπέδου λειτουργίας των χώρων αυτών. Ένα άλλο, εξίσου σοβαρό ζήτημα, είναι η πρόσβαση σε ευαίσθητα προσωπικά δεδομένα από το ιατρικό και παραϊατρικό προσωπικό που χειρίζεται τους ηλεκτρονικούς φακέλους αυτούς. Θα πρέπει να υπάρχει διαχωρισμός επιπέδων στην εφαρμογή και ανάλογα την ειδικότητα αυτουνού που χρησιμοποιεί την εφαρμογή αυτή, να έχει και την ανάλογη πρόσβαση στα δεδομένα του ασθενή. Επίσης, θα ήταν πολύ χρήσιμο, η εφαρμογή να εκμεταλλεύεται τα στατιστικά στοιχεία των εξετάσεων, των εγγραφόμενων ασθενών στην βάση, και βάση αυτά να προτείνει το κατάλληλο φάρμακο ή θεραπεία για την γρηγορότερη ανάρρωση του ασθενή ή ακόμα και να υπάρχει καλύτερη πρόληψη των αντίστοιχων ασθενειών.

## 17.Βιβλιογραφία

1. Guide to Medical Informatics, The Internet and Telemedicine by ENRICO COIERA, 1997
2. Αποστολάκης,Ι. (2002). Πληροφοριακά Συστήματα Υγείας. Εκδόσεις Παπαζήση, Αθήνα
3. Committee on Improving the Patient Record, Institute of Medicine, Tang PC, Hammond WE. A Progress Report on Computer-Based Patient Records in the United States. In: Dick RS, Steen EB, Detmer DE, editors. The Computer-Based Patient Record: An Essential Technology for Health Care. Rev ed. Washington, DC: National Academies Press; 1997. URL: <http://books.nap.edu/html/computer/commentary.html> [accessed 2005 Oct 15]
4. Ευρωπαϊκή Επιτροπή Προτυποποίησης, CEN/TC25/WG1/N8. Εργαστήριο Ιατρικής Φυσικής, Ιατρική Σχολή Πανεπιστημίου Αθηνών, Σπύρος Δευτεραίος /LSD\_PC/TELEMED/DOCS/HEATH/VARIOUS/HIF.DOC <http://asclepieion.mpl.uoa.gr/> [accessed 2005 Oct 27]
5. The Computer-based Patient Record - An Essential Technology for Health Care, National Academy Press, Washington, DC, Disk and Steen, 1991.
6. Παναγιώτης, Μάτσης. *Ηλεκτρονικός ιατρικός φάκελος*.
7. The Relationship between Electronic Health Record Use and Quality of Care over Time - Journal of the American Medical Informatics Association Volume 16 Number 4 July / August 2009
8. Electronic Health Record - Springer New York, 2013
9. C# 3.0 in a Nutshell, Third Edition By Joseph Albahari, Ben Albahari
10. Τοποθεσία Web – Asp.NET MVC <http://www.asp.net/mvc/tutorials/older-versions/overview/asp-net-mvc-overview>
11. Τοποθεσία Web – Csharp Station <http://www.csharp-station.com/Tutorial.aspx>
12. Benefits and drawbacks of electronic health record systems - Nir Menachemi and Taleah H Collum, Risk Manag Healthc Policy. 2011
13. MySQL Cookbook Paperback – January 27, 2007
14. MySQL in a Nutshell - O'Reill, 2nd Edition 2008
15. Wikipedia, «XAML,» 10 April 2014. Available: <http://en.wikipedia.org/wiki/XAML>.

