

ΑΝΩΤΑΤΟ
ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ



ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΠΟΛΥΜΕΣΩΝ

e-LIBRARY

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: « Διαδυκτιακή Εφαρμογή Ηλεκτρονική
Δανειστική Βιβλιοθήκη. »

Εισηγητής: Ιωάννης Παχουλάκης

Σπουδαστής: Παπαδανέλλης Γιώργος

ΑΜ: 638

Κεφάλαιο 1^ο

1.Εισαγωγή

Το αντικείμενο της πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εφαρμογής αυτοματοποίησης όλων των εργασιών μιας κλασσικής δανειστικής βιβλιοθήκης.

Η εφαρμογή δημιουργήθηκε σε τεχνολογία ASP .NET, με χρήση γλώσσας C#. Για την κατασκευή της επίσης, χρησιμοποιήθηκε βάση δεδομένων Microsoft Access με χρήση SQL Server η οποία επιτρέπει την είσοδο του προσωπικού της βιβλιοθήκης ώστε να μπορεί να γίνεται η διαχείριση των διαφόρων βιβλίων και των χρηστών της βιβλιοθήκης.

Μπορούμε να αναφερθούμε γενικά στην λειτουργία της εφαρμογής λέγοντας ότι είναι μία Web εφαρμογή, της οποίας η χρήση γίνεται από το προσωπικό της βιβλιοθήκης. Έτσι ο υπεύθυνος από την βιβλιοθήκη μπαίνει με την χρήση του Internet Explorer στην εφαρμογή και μπορεί να καταχωρήσει τα στοιχεία ενός χρήστη ώστε να γίνει μέλος, να ελέγξει την κατάσταση του λογαριασμού ενός χρήστη, να κάνει διάφορες αλλαγές στην Βάση Δεδομένων των βιβλίων και γενικά να εξυπηρετήσει ένα μέλος της βιβλιοθήκης, στο δανεισμό ενός βιβλίου. Αναλυτικότερα θα δούμε τις διάφορες λειτουργίες σε επόμενα κεφάλαια.

Επίσης θα γίνουν αναφορές στις διάφορες τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Επίσης θα υπάρξει αναφορά σχετικά με τον τρόπο λειτουργίας μιας Βιβλιοθήκης καθώς και με τις αρχές που πρέπει να τηρούνται μέσα σε αυτές, και από τις δύο πλευρές δηλαδή και από τους χρήστες και από του υπεύθυνους χρήσης της βιβλιοθήκης, το προσωπικό. Ένα ακόμα σημείο το οποίο θα επισημανθεί θα είναι το User Interface στο οποίο δουλεύουν οι χρήστες και ποιες οι αρχές

που το διέπουν για να είναι όσο το δυνατόν καλύτερα προσβάσιμο από ένα μέσο χρήστη.

Τέλος θα γίνουν αναφορές σε μελλοντικές προσθήκης και προοπτικές εξέλιξης μιας ηλεκτρονικής βιβλιοθήκης καθώς και σε τεχνολογίες οι οποίες εξελίσσονται και μπορούν να προστεθούν σε εφαρμογές τέτοιου είδους.

Κεφάλαιο 2^ο

2.Τεχνολογίες Υλοποίησης

Στο κεφάλαιο που ακολουθεί θα γίνει μία αναλυτική αναφορά στις τεχνολογίες, γλώσσες προγραμματισμού, και σε όλα τα εργαλεία που χρησιμοποιήθηκαν για να πραγματοποιηθεί η κατασκευή της Web εφαρμογής.

2.1 Microsoft Visual Studio

Το Visual Studio είναι ένα ολοκληρωμένο set από αναπτυξιακά εργαλεία για την κατασκευή ASP.NET Web εφαρμογές, XML Web Υπηρεσίες, desktop εφαρμογές και mobile εφαρμογές (εφαρμογές που χρησιμοποιούνται από κινητές συσκευές, κινητά τηλέφωνα, palm-tops). Visual Basic, Visual C++, Visual C#, Visual J# είναι γλώσσες, οι οποίες χρησιμοποιούν όλες το ίδιο ανεπτυγμένο προγραμματιστικό περιβάλλον (integrated development environment IDE), το οποίο δίνει την δυνατότητα να γίνεται χρήση εργαλείων και εργασιών που αλληλεπιδρούν, στην δημιουργία πολυγλωσσικών (mix-Language) εφαρμογών. Επιπλέον αυτές οι γλώσσες κάνουν χρήση της λειτουργικότητας του .NET Framework, το οποίο προωθεί πρόσβαση σε τεχνολογίες που είναι χρήσιμες στην ανάπτυξη ASP Web εφαρμογές και XML Web υπηρεσίες.

Το Visual Studio παρέχει χαρακτηριστικά που είναι πολύ σημαντικά για τους προγραμματιστές, όπως αντικειμενοστραφή προγραμματισμό, ακολουθίες χαρακτήρων, γραφικά, συστατικά γραφικού περιβάλλοντος χρήστη (GUI), χειρισμό εξαιρέσεων, πολυνηματισμό, πολυμέσα (ήχο, εικόνες, video), επεξεργασία αρχείων προκαθορισμένες δομές δεδομένων, επεξεργασία βάσεων δεδομένων, δικτύωση βασισμένη στο Internet, client/server και κατανεμημένη πληροφορική.

2.2 Γενικά για την .NET

Η .NET είναι η στρατηγική της Microsoft, για τις διαδικτυακές υπηρεσίες ώστε να συνδεθούν πληροφορίες, χρήστες, συστήματα και συσκευές μέσω ενός λογισμικού. Ενσωματωμένη πέρα από την πλατφόρμα της Microsoft, η τεχνολογία .NET παρέχει τη δυνατότητα να δημιουργήσει, να επεκτείνει, να διαχειριστεί, και να συνδέσει, ενισχυμένες σε ασφάλεια λύσεις με τις WEB υπηρεσίες. Οι λύσεις αυτές συνδεδεμένες επιτρέπουν στις επιχειρήσεις να ενσωματώσουν τα συστήματά τους γρηγορότερα και με τρόπο πιο εύκολο καθώς επίσης και να τα βοηθήσουν να πραγματοποιήσουν τις απαιτήσεις των πληροφοριών οποτεδήποτε, οπουδήποτε, για οποιαδήποτε συσκευή. Η πλατφόρμα αυτή της Microsoft περιλαμβάνει όλα όσα μια επιχείρηση χρειάζεται για να αναπτύξει και να επεκτείνει μια υπηρεσία που είναι συνδεδεμένη με IT αρχιτεκτονική όπως: servers για να φιλοξενούν τις Web υπηρεσίες, τα αναπτυξιακά εργαλεία για να τα δημιουργήσει και τις εφαρμογές για να τις χρησιμοποιούν, καθώς και ένα δίκτυο από συνεργάτες που μπορούν να βοηθήσουν την επιχείρηση, σε οποιαδήποτε πρόβλημα.



2.2.1 Τι είναι Web Υπηρεσίες

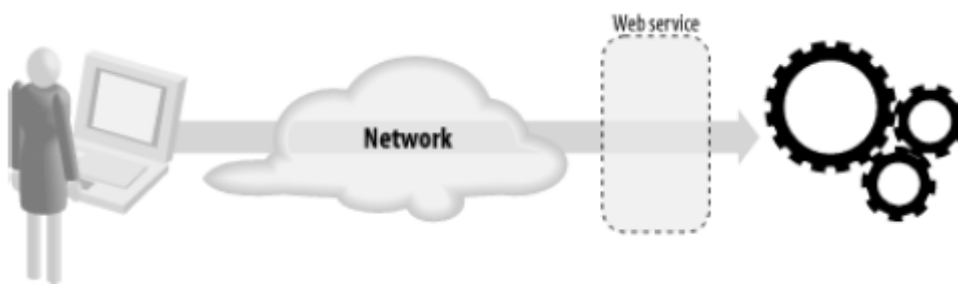
Αν ρωτήσουμε ένα ειδικό (service developer) τι είναι μία Web υπηρεσία θα πάρουμε μία απάντηση αρκετά ειδικευμένη, για παράδειγμα «Αυτόνομες ενότητες λογισμικού, σημασιολογικά συμπυκνώνουν ξεχωριστές λειτουργικότητες οι οποίες είναι προσιτές μέσω τυποποιημένων πρωτοκόλλων επικοινωνίας Διαδικτύου όπως το XML και το SOAP. Αλλά αν ρωτήσουμε έναν εκπρόσωπο μίας επιχείρησης ο οποίος έχει εφαρμόσει μία Web υπηρεσία, θα πάρουμε μία εντελώς διαφορετική απάντηση. Θα ακούσουμε ότι οι Web υπηρεσίες βοηθούν την επιχείρηση να επικοινωνεί με τους πελάτες, τους συνεργάτες και τους υπαλλήλους της. Επίσης επιτρέπουν στην εταιρία να επεκτείνει υπάρχουσες υπηρεσίες σε

νέους πελάτες. Βοηθούν την επιχείρηση να λειτουργήσει αποτελεσματικότερα με τους συνεργάτες και τους προμηθευτές της. Κάνουν οποιαδήποτε πληροφορία προσιτή σε κάθε υπάλληλο που την χρειάζεται. Μειώνουν το χρόνο και τη δαπάνη ανάπτυξης για τα νέα προγράμματα. Θα ακούσουμε πολύ περισσότερα για το τι μία Web υπηρεσία επιτρέπει σε μία επιχείρηση να κάνει, παρά το τι συγκεκριμένα είναι.

2.2.2 Οφέλη των Web Υπηρεσιών

Επιτρέποντας στις εφαρμογές, να μοιραστούν δεδομένα μεταξύ διαφόρων hardware πλατφόρμων και λειτουργικών συστημάτων, παρέχουν πολλά οφέλη που περιλαμβάνουν:

- Ανοίγουν τον δρόμο, σε νέες επιχειρηματικές ευκαιρίες δημιουργώντας εύκολες επικοινωνίες μεταξύ των συνεργατών.
- Εντυπωσιακές εμπειρίες στους χρήστες μέσω νέων έξυπνων συσκευών, στις οποίες φυσικά περιλαμβάνονται και οι Ηλεκτρονικοί Υπολογιστές.
- Εξοικονόμηση χρόνου και χρήματος μειώνοντας το χρόνο κατασκευής των εφαρμογών.
- Αύξηση του εισοδήματος των επιχειρήσεων καθιστώντας τις Web υπηρεσίες να είναι διαθέσιμες σε άλλες.



2.2.3 Συνδέοντας εφαρμογές με web Υπηρεσίες

Οι Web services είναι επαναστατικές με το πώς οι διάφορες εφαρμογές επικοινωνούν μεταξύ τους ή ευρύτερα πώς οι διάφοροι υπολογιστές επικοινωνούν μεταξύ τους, αφού παρέχει ένα καθολικό format δεδομένων το οποίο είναι αφήνει τα δεδομένα να προσαρμοστούν εύκολα ή να μετασχηματιστούν. Βασισμένες στην XML, την παγκόσμια γλώσσα

ανταλλαγής στο Διαδίκτυο, οι Web services μπορούν να επικοινωνούν μεταξύ διαφορετικών πλατφόρμων και λειτουργικών συστημάτων, ανεξάρτητα από την γλώσσα προγραμματισμού στην οποία η εφαρμογές γράφονται.

Κάθε web υπηρεσία είναι μία ξεχωριστή μονάδα κώδικα που χειρίζεται ένα περιορισμένο σύνολο στόχων-αποτελεσμάτων. Εντούτοις, αν και οι web υπηρεσίες παραμένουν ανεξάρτητες η μια από την άλλη, μπορούν αόριστα να συνδεθούν σε μια συνεργαζόμενη ομάδα που εκτελεί έναν ιδιαίτερο αποτέλεσμα.

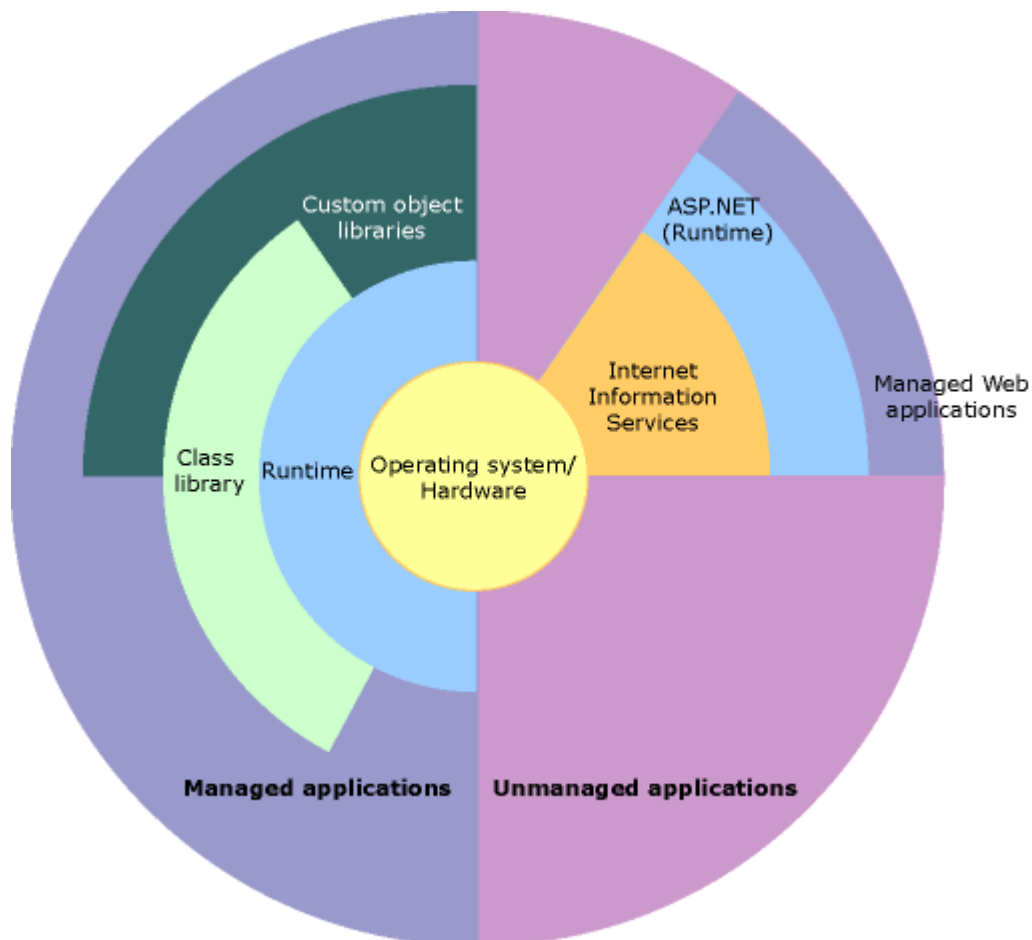
2.3 .NET Framework

Το .NET Framework είναι ένα ενιαίο συστατικό των Windows το οποίο υποστηρίζει την ανάπτυξη και το «τρέξιμο», επόμενης γενιάς εφαρμογών και XML Web υπηρεσίες. Το .Net Framework είναι σχεδιασμένο να εμπεριέχει τα παρακάτω αντικείμενα λειτουργιών:

- Να προσφέρει ένα object-oriented – αντικειμενοστραφή προγραμματισμό όπου κώδικας και αντικείμενα είναι καταχωρημένα και εκτελούνται τοπικά, εκτελούνται τοπικά αλλά διανέμονται στο Διαδίκτυο, ή εκτελούνται από απόσταση.
- Να προσφέρει ένα περιβάλλον, εκτέλεσης κώδικα που μειώνει την ανάπτυξη συμβατού λογισμικού και τα προβλήματα εκδόσεων λογισμικού.
- Να προσφέρει εκτέλεση κώδικα που προωθεί μία ασφαλή εκτέλεση του κώδικα μιας εφαρμογής για παράδειγμα, συμπεριλαμβανομένου και κώδικα που έχει δημιουργηθεί από κάποιο άγνωστο μέρος που δεν είναι και τόσο εμπιστευτικό.
- Να προσφέρει ένα περιβάλλον εκτέλεσης κώδικα στο οποίο αποτρέπονται προβλήματα από scripted ή interpreted περιβάλλοντα.
- Να προσφέρει την εμπειρία σε προγραμματιστές από μία ευρεία ποικιλία τύπων εφαρμογών όπως εφαρμογές βασισμένες σε Windows ή βασισμένες στο Web.

- Να κατασκευάζει επικοινωνιακά και βιομηχανικά πρότυπα που πιστοποιούν ότι κώδικας βασισμένος σε .NET Framework μπορεί να ενοποιηθεί με οποιοδήποτε άλλο κώδικα.

Το .NET Framework έχει δύο κύρια συστατικά: την κοινή επεξεργασία του κώδικα και την .NET framework βιβλιοθήκη κλάσεων (class library). Ο χρόνος τρεξίματος της γλώσσας προγραμματισμού είναι η οργάνωση του .NET framework. Μπορούμε να παρομοιώσουμε το χρόνο τρεξίματος (runtime) σαν έναν υπεύθυνο ο οποίος διευθύνει τον κώδικα κάποιας εφαρμογής τον χρόνο που εκτελείτε προσφέροντας υπηρεσίες που προέρχονται από τον πυρήνα, όπως για παράδειγμα, την διοίκηση της μνήμης του συστήματος σε σχέση με την εφαρμογή, την ασφάλεια του συστήματος, και την σωστή λειτουργία. Το παρακάτω διάγραμμα παρουσιάζει τις σχέσεις μεταξύ του runtime της γλώσσας και του class library στις διάφορες εφαρμογές και του όλου συστήματος του υπολογιστή. Επίσης παρουσιάζει πως οι λειτουργίες κώδικα διευθύνονται σε μία υψηλή αρχιτεκτονική.



2.3 Γλώσσα Προγραμματισμού C#

2.3.1 Γενικά για την C#

Η C# είναι μία ενιαία και ασφαλής αντικειμενοστραφής γλώσσα προγραμματισμού η οποία προσφέρει στους προγραμματιστές την δυνατότητα κατασκευής πολλών ασφαλών εφαρμογών, οι οποίες τρέχουν στο .NET Framework . Η C# μπορεί να χρησιμοποιηθεί για την δημιουργία παραδοσιακών απομακρυσμένων εφαρμογών (client applications) για τα Windows, XML Web υπηρεσίες, client- server εφαρμογές, εφαρμογές βάσης δεδομένων και πολλές πολλές άλλες. Η Microsoft Visual C# χρησιμοποιεί ένα προηγμένο κατασκευαστή κώδικα (code editor), εξελιγμένο κατασκευαστή user interface, ανεπτυγμένο debugger, και πολλά ακόμα εργαλεία που βοηθούν στην ταχύτερη κατασκευή εφαρμογών.

2.3.2 C# Γλώσσα

Η σύνταξη της C# είναι υψηλής εκφραστικότητας με λιγότερα από 90 λέξεις-κλειδιά (keywords), που συνεπάγεται στην ευκολία και την απλότητα στην εκμάθησή της. Είναι επίσης αναγνωρίσιμη σε οποιοδήποτε έχει ασχοληθεί με C++, C, ή Java. Προγραμματιστές οι οποίοι γνωρίζουν μία από αυτές τις γλώσσες είναι τυπικά έτοιμοι σε λίγο χρονικό διάστημα να δουλέψουν σε C#. Η σύνταξη της C# έχει πολλές ομοιότητες με αυτή της C++, προωθώντας περισσότερα δυναμικά στοιχεία όπως τύπους δεδομένων null, enumerations, επιπλέον δηλώσεις και ανώνυμες μεθόδους, και απ' ευθείας πρόσβαση στην μνήμη την οποία δυνατότητα δεν την βρίσκουμε στην Java. Η C# επίσης εμπεριέχει γενικές μεθόδους και τύπους δεδομένων τα οποία προσφέρουν ανεπτυγμένους τύπους ασφάλειας και εκτέλεσης, οι οποίοι «δραστηριοποιούν» μία συλλογή από κλάσης για να εμφανίσουν συγκεκριμένες συμπεριφορές διαφόρων εργαλείων μέσα σε μία εφαρμογή και που είναι επίσης εύκολο να χρησιμοποιηθούν από ένα μακρινό κώδικα.

Ως μία αντικειμενοστραφής γλώσσα, περιλαμβάνει επίσης και τις έννοιες της ενθυλάκωσης (encapsulation), της κληρονομικότητας (inheritance) και του πολυμορφισμού (polymorphism).

2.3.3 Object – oriented προγραμματισμός

Το **Object Model** που χρησιμοποιείται στη C# είναι παρόμοιο με άλλες αντικειμενοστραφείς γλώσσες (Object Oriented Languages). Η αντικειμενοστρέφεια είναι νεώτερη από το Δομημένο Προγραμματισμό, αλλά είναι πλέον πολύ σημαντική για τους προγραμματιστές σήμερα. Ο Object Oriented προγραμματισμός παρέχει μια τυποποιημένα γλώσσα και ένα πλαίσιο που είναι σημαντικό για να κάνει την εργασία τμημάτων λογισμικού (objects) ενωμένα μαζί σε ένα σύστημα. Οι Object Oriented έννοιες που μαθαίνονται για τη VB ισχύουν για τις περισσότερες γλώσσες που ένας προγραμματιστής είναι πιθανό να αντιμετωπίσει σήμερα.

Μια γλώσσα βασισμένη σε αντικείμενα έχει τις περισσότερες από τις ιδιότητες μιας αντικειμενοστραφούς γλώσσας, αλλά μπορεί να στερείται μερικών. Παραδείγματος χάριν η Visual Basic δεν έχει κληρονομικότητα, ενώ μια γλώσσα βασισμένη σε πρωτότυπα στηρίζεται σε αυτά αντί στις κλάσεις για να δημιουργήσει τα αντικείμενα.

Η αντικειμενοστρέφεια (Object Orientation) έχει αλλάξει τον τρόπο που οι βιομηχανίες λογισμικού προσεγγίζουν την ανάπτυξη λογισμικού, και ιδιαίτερα, πώς οι τεχνικοί προσεγγίζουν την ανάλυση, το σχεδιασμό, την επαναχρησιμοποίηση κώδικα, και την υλοποίηση.

Ο Object Oriented προγραμματισμός είναι μια προσέγγιση (όχι ένα συγκεκριμένο εργαλείο) που οργανώνεται γύρω από τα αντικείμενα παρά από τις εφαρμογές, τα δεδομένα ή τη λογική.

Objects

Ως αντικειμενοστραφής γλώσσα η C# , οργανώνεται γύρω από τα αντικείμενα. Τα αντικείμενα είναι προγραμματισμένοι κατασκευαστές οι οποίοι έχουν δεδομένα, ταυτότητα και συμπεριφορά. Τα δεδομένα των αντικειμένων περιλαμβάνονται στα πεδία τους, στις ιδιότητες και στα γεγονότα (events), ενώ οι συμπεριφορές των αντικείμενων ορίζονται από τις μεθόδους και το

interface τους. Τα αντικείμενα έχουν ταυτότητα δηλαδή δύο αντικείμενα με το ίδιο set δεδομένων δεν είναι απαραίτητα το ίδιο αντικείμενο.

Τα αντικείμενα έχουν τις εξής ιδιότητες:

- Οποδήποτε και αν χρησιμοποιείται στην C# είναι αντικείμενο, ακόμα και οι φόρμες των Windows και τα controls.
- Τα αντικείμενα δημιουργούνται από τα templates που ορίζονται από κλάσεις και structs.
- Τα αντικείμενα χρησιμοποιούν τις ιδιότητές τους για να λειτουργήσουν και να αλλάξουν τις πληροφορίες τις οποίες εμπεριέχουν.
- Συχνά τα αντικείμενα έχουν μεθόδους και γεγονότα τα οποία τους επιτρέπουν να κάνουν δραστηριότητες (actions).
- Όλα τα αντικείμενα στην C# ιεραρχοποιούνται από το Αντικείμενο.

Βλέποντας την ανάλυση και το σχεδιασμό του λογισμικού ως αντικείμενα είναι πολύ σημαντικό επειδή και οι άνθρωποι σκέφτονται σε σχέση με τα αντικείμενα. Όλα γύρω μας είναι αντικείμενα.

Classes (Κλάσεις)

Μια κλάση είναι ένα σύνολο παρόμοιων ενεργειών και ιδιοτήτων, και ένα αντικείμενο είναι ένα instance (μια περίπτωση) μιας κλάσης. Μπορούμε να σκεφτούμε μια κλάση ως σχεδιασμό ενός αντικειμένου που θα δημιουργηθεί στον κώδικά μας. Και οι κλάσεις και τα αντικείμενα έχουν ταυτότητα. Μια κλάση καθορίζει τις ιδιότητες και τις μεθόδους, ενώ ένα αντικείμενο τις χρησιμοποιεί.

Μια κλάση καθορίζει τα στοιχεία των δεδομένων και τη μορφή των δεδομένων που συνδυάζονται για να απεικονίσουν το τι ένα αντικείμενο «γνωρίζει». Παραδείγματος χάριν, μπορούμε να δημιουργήσουμε ένα αντικείμενο αποκαλούμενο 'Oscar' από μια κλάση που ονομάζεται Dog. Η κλάση Dog καθορίζει τι πρόκειται να είναι το αντικείμενο 'Oscar', και όλα τα «dog-related»

μηνύματα που ένα αντικείμενο Oscar μπορεί να ενεργήσει επάνω τους. Όλες οι αντικειμενοστραφείς γλώσσες έχουν κάποιες έννοιες. Καλούν μερικές φορές ένα «εργοστάσιο», να κατασκευάσει τα instances ενός αντικειμένου από τον καθορισμό μιας κλάσης. Μπορούμε να κάνουμε περισσότερα από ένα αντικείμενα αυτής της κλάσης, και να τα καλέσουμε Spot, Fido, Rover, κ.λπ. Η κλάση Dog καθορίζει τα μηνύματα που τα αντικείμενα Dog καταλαβαίνουν, όπως «bark», «fetch», και «roll-over».

Οι κλάσεις και τα αντικείμενα έχουν χαρακτηριστικά. Παραδείγματα χαρακτηριστικών τους για ένα άτομο είναι η ηλικία, το ύψος, το βάρος, και δείκτης νοημοσύνης. Αυτά τα χαρακτηριστικά αναφέρονται ως χαρακτηριστικά κλάσης και ιδιότητες αντικειμένου. Τα χαρακτηριστικά της κλάσης και οι ιδιότητες του αντικειμένου είναι επίσης γνωστές ως member fields. Επειδή ένα αντικείμενο έχει τη δυνατότητα να θέσει τις ιδιότητες, τα αντικείμενα λέγονται επίσης ότι έχουν την κυριότητα. Οι τρέχουσες τιμές των χαρακτηριστικών ενός αντικειμένου είναι η επικρατούσα κατάσταση τους.



Properties (Ιδιότητες)

Τα **Properties** (ιδιότητες) λένε για το αντικείμενο, πληροφορίες όπως είναι το όνομά του, η θέση του στον υπολογιστή, εάν είναι ορατό, ενεργό, ή το χρώμα του. Οι ιδιότητες είναι όπως τα επίθετα που περιγράφουν τα αντικείμενα (ουσιαστικά). Οι ιδιότητες στην .NET δηλώνονται με τη χρησιμοποίηση του ονόματος αντικειμένου, μιας περιόδου (.), και την επιθυμητή ιδιότητα. Για παράδειγμα αν θέλουμε να δηλώσουμε ότι ένα αντικείμενο είναι ορατό δηλαδή Visible, τότε θα χρησιμοποιήσουμε `public namespace Visible { get; set; }` και οι τιμές που παίρνει είναι true ή false. Η σημαντικότερη ιδιότητα κάθε αντικειμένου είναι γενικά το όνομά της. Το όνομα είναι αυτό που δένει το αντικείμενο σε ένα script και το πώς ένα αντικείμενο δηλώνεται από

οποιοδήποτε άλλο αντικείμενο. Η κωδικοποίηση της κλάσης πρέπει να λαμβάνεται υπόψιν κατά την ονομασία των αντικειμένων.

Methods (Μέθοδοι)

Οι κλάσεις και τα αντικείμενα έχουν ενέργειες. Αυτές οι ενέργειες αναφέρονται ως λειτουργίες κλάσης και **μέθοδοι** αντικειμένου. Τις αναφέρουμε γενικά ως μεθόδους (**methods**). Μέθοδος είναι το πώς ο κώδικας μπορεί να χρησιμοποιήσει ένα αντικείμενο κάποιας κλάσης. Οι μέθοδοι μπορούν να διαιρεθούν σε queries (ερωτήσεις) επιστρέφοντας την επικρατούσα κατάσταση και σε commands (εντολές) που την αλλάζουν (υπορουτίνα). Μερικές φορές η πρόσβαση στα δεδομένα ενός αντικειμένου είναι περιορισμένη στις μεθόδους της κλάσης του.

Μια μέθοδος είναι ουσιαστικά η εφαρμογή μιας υπηρεσίας αντικειμένου η οποία είναι απλά η δράση που ένα μήνυμα μεταφέρει. Είναι ο κώδικας, ο οποίος εκτελείται όταν το μήνυμα στέλνεται σε ένα συγκεκριμένο αντικείμενο. Ορίσματα παρέχονται συχνά ως τμήμα ενός μηνύματος. Στον Object Oriented προγραμματισμό στέλνουμε ένα μήνυμα από ένα αντικείμενο σε ένα άλλο. Οι μέθοδοι και οι ιδιότητες είναι κλήσεις μηνυμάτων. Οι παράμετροι μιας μεθόδου είναι το περιεχόμενο των μηνυμάτων.

Events (Γεγονότα)

Τα γεγονότα (**Events**) είναι το πως τα αντικείμενα πρέπει να ανταποκρίνονται. Μπορούν να είναι ερεθίσματα για μεθόδους αντικειμένων και αναφέρονται ως Object.Events. Για παράδειγμα όταν σε μία εφαρμογή ο χρήστης θα πρέπει να πατήσει ένα κουμπί τότε ενεργοποιείτε το γεγονός (event) **OnClick()**; Άλλα γεγονότα είναι παραδείγματος χάριν:

event EventHandler DoubleClick

event EventHandler MouseEnter

event MouseEventHandler MouseMove

και φυσικά πολλά πολλά άλλα.

Inheritance (Κληρονομικότητα)

Η αρχή μιας κλάσης καθιστά πιθανό να καθοριστούν οι υποκλάσεις που μοιράζουν μερικά ή όλα τα κύρια χαρακτηριστικά της κλάσης. Αυτό καλείται κληρονομικότητα (**inheritance**). Η κληρονομικότητα μας επιτρέπει επίσης να επαναχρησιμοποιήσουμε τον κώδικα αποτελεσματικότερα. Ένας μηχανισμός για τις υποκλάσεις και παρέχει έναν τρόπο να καθοριστεί μια υποκλάση ως εξειδίκευση ή υποκατηγορία ή επέκταση μιας γενικότερης κλάσης. Μια υποκλάση κληρονομεί όλα τα μέλη (members) των superclass (υπερκλάσεων) της, αλλά μπορεί να επεκτείνει τη «συμπεριφορά» τους και να προσθέσει τα νέα μέλη.

Encapsulation

Το **encapsulation** (ενθυλάκωση) είναι το κρύψιμο των στοιχείων και του κώδικα και καλείται συχνά ως «black box» προσέγγιση, δεδομένου ότι οι χρήστες μιας κλάσης δεν μπορούν να δουν μέσα στην κλάση (μπορούν να δουν μόνο το public interface της). Εξασφαλίζει επίσης ότι ο κώδικας εξωτερικά από μια κλάση βλέπει μόνο τις λειτουργικές λεπτομέρειες εκείνης της κλάσης, αλλά όχι τις λεπτομέρειες της υλοποίησης. Το Encapsulation επιτυγχάνεται διευκρινίζοντας ποιες κλάσεις μπορούν να χρησιμοποιήσουν τα μέλη ενός αντικειμένου. Το αποτέλεσμα είναι ότι κάθε αντικείμενο εκθέτει σε οποιαδήποτε κλάση ένα συγκεκριμένο interface. Τα μέλη καθορίζονται συχνά ως public, protected και private. Αυτό καθορίζεται από το εάν είναι διαθέσιμα σε όλες τις κλάσεις, τις υποκλάσεις ή μόνο την καθορισμένη κλάση. Μερικές γλώσσες επεκτείνονται περαιτέρω: Η Java χρησιμοποιεί protected λέξη κλειδί για να περιορίσει την πρόσβαση. Η C# και η VB.NET υποκαθιστούν μερικά μέλη σε κλάσεις χρησιμοποιώντας τις λέξεις κλειδιά ως internal (C#) ή ως Friend (VB.NET). Τα αντικείμενα αλληλεπιδρούν το ένα με το άλλο μέσω μηνυμάτων. Το μόνο πράγμα που ένα αντικείμενο γνωρίζει για ένα άλλο είναι το interface του αντικειμένου. Τα δεδομένα και η λογική κάθε αντικειμένου

είναι κρυμμένα από άλλα αντικείμενα. Αυτό επιτρέπει στον αναλυτή να χωρίσει την υλοποίηση ενός αντικειμένου από το interface του. Εφ' όσον το interface παραμένει ίδιο, οποιεσδήποτε αλλαγές στην εσωτερική υλοποίηση είναι προφανείς στο χρήστη.

Polymorphism (Πολυμορφισμός)

Ένα άλλο όφελος που προκύπτει από το διαχωρισμό της υλοποίησης από τη συμπεριφορά είναι ο πολυμορφισμός (**polymorphism**). Ο πολυμορφισμός είναι συμπεριφορά που ποικίλλει ανάλογα με την κλάση στην οποία η συμπεριφορά επιδρά, δηλαδή δύο ή περισσότερες κλάσεις μπορούν να αντιδράσουν διαφορετικά στο ίδιο μήνυμα. Ο πολυμορφισμός επιτρέπει σε δύο ή περισσότερα αντικείμενα για να αποκριθούν στο ίδιο μήνυμα. Με άλλα λόγια, ο πολυμορφισμός επιτρέπει σε οποιαδήποτε απόγονο κλάση για να επαναπροσδιορίσει οποιαδήποτε μέθοδο που κληρονομείται από την κλάση γονέων του. Η επίδραση είναι ότι ο πολυμορφισμός επιτρέπει σε ένα σταλμένο αντικείμενο να επικοινωνήσει με τα διαφορετικά αντικείμενα κατά σύμφωνο τρόπο χωρίς ανησυχία για το πόσες διαφορετικές υλοποιήσεις ενός μηνύματος υπάρχουν.

2.3.4 ASP.NET

Γενικά είναι ένα ενιαίο κοινά αποδεκτό Web development μοντέλο το οποίο περιέχει τις απαραίτητες υπηρεσίες για την κατασκευή αναπτυσσόμενων – κλάσεων Web εφαρμογών, με την χρήση του λιγότερου δυνατού κώδικα. Το ASP.NET είναι μέρος του .NET Framework και όταν προγραμματίζουμε έχουν την πρόσβαση σε κλάσεις του. Μπορούμε να προγραμματίσουμε εφαρμογές σε οποιαδήποτε γλώσσα είναι συμβατή με την CLR (Common Language RunTime), όπως οι γλώσσες που προαναφερθήκαμε όπως C#, Jscript, J#. Με αυτές τις γλώσσες έχουμε την δυνατότητα να αναπτύξουμε Web εφαρμογές, έχοντας πλεονέκτημα από το CLR, αναπτύσσοντας ασφαλέστερα και με μεγαλύτερη ιεραρχικότητα.

Το ASP.NET περιλαμβάνει:

- Framework για σελίδα και για τα controls.
- Ένα ASP.NET compiler (μεταγλωττιστή).
- Διαμόρφωση Ασφαλείας.
- State-management δραστηριότητες.
- Configuration της εφαρμογής.
- Debugging υποστήριξη.
- Framework για XML Web υπηρεσίες.
- Επεκτάσιμο περιβάλλον host και διαχείρισης ων εφαρμογών.
- Ένα επεκτάσιμο σχεδιαστικό περιβάλλον.



2.3.5 ADO.NET

Η ADO.NET είναι μια εξέλιξη του ADO μοντέλου πρόσβασης δεδομένων, που εξετάζει άμεσα τις απαιτήσεις χρηστών για τις εξελικτικές εφαρμογές. Η ADO.NET χρησιμοποιεί μερικά ADO αντικείμενα, όπως τα αντικείμενα σύνδεσης (**Connection Objects**) και εντολών (**Command Objects**), και εισάγει επίσης νέα αντικείμενα. Τα βασικά νέα αντικείμενα ADO.NET περιλαμβάνουν το **DataSet**, το **DataReader**, και το **DataAdapter**.

Η σημαντική διάκριση μεταξύ αυτού του εξελιγμένου σταδίου της ADO.NET και των προηγούμενων αρχιτεκτονικών στοιχείων είναι ότι υπάρχει ένα αντικείμενο -- το **DataSet** -- το οποίο είναι ξεχωριστό και ευδιάκριτο από οποιοδήποτε αποθηκευμένο δεδομένο. Λόγω αυτού, το **DataSet** λειτουργεί ως αυτόνομη οντότητα. Μπορείτε να σκεφτείτε το **DataSet** ως ένα μόνιμα αποσυνδεδεμένο recordset που δεν ξέρει τίποτα για την πηγή ή τον προορισμό των στοιχείων που περιέχει. Μέσα σε ένα **DataSet**, όπως σε μια βάση δεδομένων, υπάρχουν πίνακες, στήλες, σχέσεις, περιορισμοί, απόψεις, και ούτω καθ'εξής.

Ένα **DataAdapter** είναι το αντικείμενο που συνδέεται με τη βάση δεδομένων για να γεμίσει το **DataSet**. Κατόπιν, συνδέεται πίσω στη βάση δεδομένων για να ενημερώσει τα δεδομένα εκεί, βασισμένα στις διαδικασίες που πραγματοποιήθηκαν ενώ το **DataSet** κρατούσε τα δεδομένα. Στο παρελθόν, η επεξεργασία δεδομένων βασιζόταν κυρίως στη σύνδεση (connection). Τώρα, σε μια προσπάθεια να γίνουν οι πολυ-στρωματικές εφαρμογές αποδοτικότερες, η επεξεργασία δεδομένων μετατρέπεται σε μια προσέγγιση που βασίζεται στα μηνύματα τα οποία κυρίως περιστρέφονται γύρω από τα σημαντικότερα κομμάτια της πληροφορίας. Στο κέντρο αυτής της προσέγγισης βρίσκεται το **DataAdapter**, το οποίο παρέχει μια γέφυρα για να ανακτηθούν και να αποθηκευθούν τα δεδομένα μεταξύ ενός **DataSet** και του πηγής των αποθηκευμένων δεδομένων (Database). Αυτό το ολοκληρώνει με τη βοήθεια αιτημάτων σε κατάλληλες εντολές SQL που γίνονται στα αποθηκευμένα δεδομένα.

DataSet αντικείμενο βασισμένο στην XML εξασφαλίζει ένα συνεπές πρότυπο προγραμματισμού που λειτουργεί με όλα τα πρότυπα αποθήκευσης δεδομένων: επίπεδο, σχεσιακό και ιεραρχικό. Το κάνει αυτό μη έχοντας καμία «γνώση» της πηγής των δεδομένων του. Χωρίς καμία σημασία ποιά είναι η πηγή των δεδομένων στο **DataSet**, χειρίζεται μέσω του ίδιου συνόλου του τυποποιημένου APIs που εκτίθεται μέσω του **DataSet** και των δευτερευόντων αντικειμένων του.

Ενώ το **DataSet** δεν έχει καμία γνώση της πηγής δεδομένων του, ο διαχειρίζον provider (managed provider) έχει λεπτομερείς και συγκεκριμένες πληροφορίες. Ο ρόλος του διαχειρίζον provider είναι να συνδέσει, να γεμίσει, και να διατηρήσει το **DataSet** προς και από τις «αποθήκες» δεδομένων. Οι OLE DB και SQL Server .NET Data Providers (System.Data.OleDb και System.Data.SqlClient) που είναι μέρος του .Net Framework παρέχουν τέσσερα βασικά αντικείμενα: την Εντολή (**Command**), την Σύνδεση (**Connection**), τον **DataReader** και τον **DataAdapter**.

Παρακάτω βλέπουμε μερικά αντικείμενα που έχουν εξελιχθεί, και μερικά νέα. Αυτά τα αντικείμενα είναι:

- **Connections** - Για τη σύνδεση και τη διαχείριση εκτελέσεων και συναλλαγών σε μια βάση δεδομένων.
- **Commands** - Για την έκδοση εντολών SQL σε μια βάση δεδομένων.
- **DataReaders** - Για την ανάγνωση μίας read-only – forward-only ροής των δεδομένων (data records) από μια SQL Server πηγή .
- **DataSets** - Για την αποθήκευση και τον προγραμματισμό σε flat data, δεδομένα XML και τα σχεσιακά δεδομένα.
- **DataAdapters** - Για την ώθηση των δεδομένων σε ένα DataSet, και τη προσαρμογή των δεδομένων σε μια βάση δεδομένων.

Connections (Συνδέσεις)

Οι συνδέσεις χρησιμοποιούνται για «να μιλήσουν» στις βάσεις δεδομένων, και αναπαριστούνται από τον provider συγκεκριμένες κατηγορίες όπως είναι η SQL Connection. Οι εντολές πάνω από τις συνδέσεις και τα εφαρμοσμένα αποτελέσματα επιστρέφονται υπό μορφή μιας ροής που μπορούν να διαβαστούν από ένα DataReader αντικείμενο, ή να ωθηθούν σε ένα DataSet αντικείμενο.

Commands (Εντολές)

Οι εντολές περιέχουν την πληροφορία που υποβάλλεται σε μια βάση δεδομένων, και αντιπροσωπεύονται από provider με συγκεκριμένες κατηγορίες όπως SQL Command. Μια εντολή μπορεί να είναι μια κλήση αποθηκευμένης διαδικασίας (stored procedure), μια ενημερωμένη δήλωση, ή μια δήλωση που επιστρέφει τα αποτελέσματα. Μπορούμε επίσης να χρησιμοποιήσουμε τις παραμέτρους εισαγωγής και εξαγωγής, και να επιστρέψουμε τις τιμές ως τμήμα της σύνταξης εντολής.

DataReaders

Το αντικείμενο DataReader είναι κάπως συνώνυμο με ένα read only/forward only δρομέα πάνω από τα δεδομένα. Ο DataReader API υποστηρίζει επίπεδα καθώς επίσης και ιεραρχικά δεδομένα. Ένα DataReader αντικείμενο αφού εκτελέσει μια εντολή επιστρέφεται σε μία βάση δεδομένων. Η διάταξη του επιστρεφόμενου DataReader αντικειμένου είναι διαφορετική από ένα

recordset. Για παράδειγμα μπορούμε να χρησιμοποιήσουμε το DataReader για να παρουσιάσουμε τα αποτελέσματα μιας λίστας αναζήτησης σε μία ιστοσελίδα.

DataSets

Το DataSet αντικείμενο είναι παρόμοιο με το αντικείμενο ADO Recordset, αλλά ισχυρότερο, και με μια άλλη σημαντική διάκριση: το DataSet είναι πάντα αποσυνδεδεμένο. Το DataSet αντικείμενο αντιπροσωπεύει μια κρύπτη των δεδομένων, με τη βάση δεδομένων - σαν δομές όπως οι πίνακες, οι στήλες, οι σχέσεις, και οι περιορισμοί. Εντούτοις, αν και ένα DataSet μπορεί και συμπεριφέρεται σαν μια βάση δεδομένων, είναι σημαντικό να αναφερθεί ότι τα DataSet αντικείμενα δεν αλληλεπιδρούν άμεσα με τις βάσεις δεδομένων, ή άλλα αρχικά δεδομένα. Αυτό επιτρέπει σε αυτόν που κάνει ανάπτυξη να λειτουργήσει με ένα πρότυπο προγραμματισμού που είναι πάντα συνεπές, ανεξάρτητα το που τα δεδομένα ανήκουν. Τα δεδομένα που προέρχονται από μια βάση δεδομένων, ένα αρχείο XML, από κώδικα, ή την εισαγωγή από κάποιο χρήστη μπορούν να τοποθετηθούν στα DataSet αντικείμενα. Κατόπιν, καθώς γίνονται αλλαγές στο DataSet μπορούν να ακολουθηθούν και να ελεγχθούν πριν ενημερώνουν τα αρχικά δεδομένα.

Το DataSet έχει πολλά χαρακτηριστικά XML, συμπεριλαμβανομένης της δυνατότητας να παραχθούν και να καταναλωθούν τα στοιχεία XML και τα σχήματα (schemas) XML. Τα σχήματα XML μπορούν να χρησιμοποιηθούν για να περιγράψουν τα σχήματα που ανταλλάσσονται μέσω WebServices.

DataAdapters (OLEDB/SQL)

Το αντικείμενο DataAdapter λειτουργεί ως γέφυρα μεταξύ του DataSet και των αρχικών δεδομένων. Χρησιμοποιώντας provider - συγκεκριμένο SqlDataAdapter (μαζί με τα σχετικά SqlCommand και τα SqlConnection του) μπορεί να αυξηθεί η γενική απόδοση όταν δουλεύουμε με Microsoft SQL Server Databases. Για άλλες OLE DB-υποστηριζόμενες βάσεις δεδομένων, θα χρησιμοποιούσαμε το OleDbDataAdapter αντικείμενο και τα σχετικά αντικείμενα OleDbCommand και OleDbConnection.

Το DataAdapter αντικείμενο χρησιμοποιεί τις εντολές για να ενημερώσει τα αρχικά δεδομένα αφότου έχουν γίνει οι αλλαγές στο DataSet.

Χρησιμοποιώντας τη μέθοδο **Fill** του DataAdapter καλεί την εντολή SELECT, χρησιμοποιώντας τη μέθοδο **Update** καλεί την INSERT, UPDATE ή DELETE. Μπορούμε σαφέστατα να βάλουμε αυτές τις εντολές προκειμένου να ελεγχθούν οι δηλώσεις που χρησιμοποιούνται στο χρόνο εκτέλεσης για να επιλύσουν τις αλλαγές, συμπεριλαμβανομένης της χρήσης των αποθηκευμένων διαδικασιών (stored procedures). Για συγκεκριμένα σενάρια, ένα CommandBuilder αντικείμενο μπορεί να δημιουργεί αυτά στο χρόνο εκτέλεσης που βασίζεται σε μια ξεχωριστή δήλωση. Εντούτοις, αυτή η δημιουργία του χρόνου εκτέλεσης απαιτεί έναν πρόσθετο κυκλικό δρόμο στον server προκειμένου να μαζευτούν τα απαραίτητα μεταδεδομένα, τόσο ρητά παράγοντας τις INSERT, UPDATE και DELETE εντολές στο χρόνο σχεδιασμού, θα οδηγήσει στην καλύτερη απόδοση του χρόνου εκτέλεσης.

2.3.5 XML

Κάνοντας χρήση της XML, μιας τεχνολογίας βιομηχανικού προτύπου για την περιγραφή δεδομένων, οι προγραμματιστές του Visual Studio .NET μπορούν να δημιουργήσουν εφαρμογές υψηλών επιδόσεων που βασίζονται σε δεδομένα. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν ενσωματωμένα εργαλεία ADO.NET τα οποία προορίζονται για μια ποικιλία βάσεων δεδομένων, συμπεριλαμβανομένης της βάσης δεδομένων SQL Server, της βάσης δεδομένων Oracle ή οποιασδήποτε άλλης προέλευσης XML. Με την εσωτερική υποστήριξη για XML, το εργαλείο ADO.NET επιτρέπει στους προγραμματιστές να κάνουν κοινή χρήση δεδομένων σε διαφορετικές πλατφόρμες υπολογιστών. Επιπλέον, το Visual Studio .NET περιλαμβάνει το μηχανισμό Microsoft Data Engine (MSDE), μια βάση δεδομένων 100% συμβατή με τον SQL Server η οποία προσφέρει στους προγραμματιστές μια χρησιμοποιήσιμη βάση δεδομένων προγραμματισμού και υποστηρίζει τοπικά XML για μέγιστη διαλειτουργικότητα. Συνοπτικά τα οφέλη της XML είναι τα παρακάτω:

- **Απλότητα**

Οι πληροφορίες που κωδικοποιούνται σε XML είναι εύκολο να διαβαστούν και να κατανοηθούν, συν του ότι μπορούν να επεξεργαστούν εύκολα από τους υπολογιστές.

- **Openness**

Η XML είναι ένα W3C πρότυπο, που υποστηρίζεται από τους πρωτοπόρους στην αγορά βιομηχανίας λογισμικού.

- **Επεκτασιμότητα**

Δεν υπάρχει κανένα δεδομένο σύνολο tags. Νέα tags μπορούν να δημιουργηθούν δεδομένου ότι απαιτούνται.

- **Αυτό-περιγραφή**

Στις παραδοσιακές βάσεις δεδομένων, τα δεδομένα απαιτούν σχήματα (schemas) που δημιουργούνται από τον administrator της βάσης. Τα αρχεία XML μπορούν να αποθηκευτούν χωρίς τέτοιους καθορισμούς, επειδή περιέχουν meta - data υπό μορφή tags.

Η XML παρέχει μια βάση για την αναγνώριση συντακτών και ερμηνεύεται στο επίπεδο των στοιχείων. Οποιοδήποτε XML tag μπορεί να καταλαμβάνει έναν απεριόριστο αριθμό ιδιοτήτων όπως ο συντάκτης ή η έκδοση.

- **Περιέχει machine-readable πληροφορίες**

Τα tags, οι ιδιότητες και η δομή στοιχείων παρέχουν πληροφορίες που μπορούν να χρησιμοποιηθούν για να ερμηνεύσουν την έννοια του περιεχομένου, που ανοίγει νέες δυνατότητες για ιδιαίτερα αποδοτικές μηχανές αναζήτησης, intelligent data mining, agents, κ.λπ. Αυτό είναι ένα σημαντικό πλεονέκτημα πέρα από την HTML ή το απλό κείμενο, όπου οι πληροφορίες περιεχομένου είναι δύσκολο ή αδύνατο να αξιολογηθούν.

- **Χωρίζει το περιεχόμενο από την παρουσίαση**

Τα XML tags περιγράφουν τη σημασία της μη-παρουσίασης. Το ρητό της HTML είναι: "I know how it looks" («Ξέρω πως φαίνεται»), ενώ το ρητό της XML είναι: "I know what it means, and you tell me how it should look." («Ξέρω τι σημαίνει, και μου λέτε πώς πρέπει να φαίνεται.») Η όψη και η αίσθηση ενός αρχείου XML μπορούν να ελεγχθούν από XSL style sheets, επιτρέποντας την όψη ενός αρχείου (ή ενός πλήρους Web site) να αλλάζει χωρίς να επηρεάζεται το περιεχόμενο του αρχείου.

- **Υποστηρίζει πολύγλωσσα αρχεία και Unicode**

Αυτό είναι σημαντικό για τη διεθνοποίηση των εφαρμογών.

- **Διευκολύνει τη σύγκριση και τη συνάθροιση των δεδομένων**

Η δομή δέντρου των αρχείων XML επιτρέπει στα αρχεία να συγκριθούν και να αθροιστεί αποτελεσματικά στοιχείο με στοιχείο.

- **Μπορεί να ενσωματώσει πολλαπλούς τύπους αρχείων**

Τα XML αρχεία μπορούν να περιέχουν οποιοδήποτε πιθανό τύπο δεδομένων – από δεδομένα πολυμέσων (εικόνα, ήχος, βίντεο) έως active components (Java applets, ActiveX).

- **Μπορεί να ενσωματώσει υφιστάμενα δεδομένα**

Η χαρτογράφηση των υπάρχουσών δομών δεδομένων όπως είναι τα συστήματα αρχείων ή οι σχεσιακές βάσεις, σε XML γίνεται πολύ απλή. Η XML υποστηρίζει πολλαπλά format δεδομένων και μπορεί να καλύψει όλες τις υπάρχουσες δομές δεδομένων.

- **Παρέχει 'one-server view' για τα κατανεμημένα δεδομένα**

Τα XML αρχεία μπορούν να προκύψουν από τα ένθετα στοιχεία που κατανέμονται από τους πολλαπλούς remote servers. Η XML είναι αυτήν την περίοδο το πολυπλοκότερο format για τα κατανεμημένα δεδομένα - το World Wide Web μπορεί να φανεί ως μια τεράστια XML βάση δεδομένων.

2.3.6 Βάσεις Δεδομένων και SQL Server 2000

Λόγω της φύσης της εφαρμογής μας, ήταν λογικό από την αρχή η χρήση Βάσης Δεδομένων. Εφόσον η εφαρμογή θα χρησιμοποιεί δεδομένα όπως χρήστες και βιβλία για παράδειγμα, όλα αυτά θα αποθηκεύονται σε βάση δεδομένων. Όλα τα δεδομένα θα πρέπει να αποθηκεύονται σε ένα server ο οποίος θα είναι προσβάσιμος από τους χρήστες, ώστε να μπορούν να χρησιμοποιούν με εύκολο και προσιτό τρόπο όλες τις υπηρεσίες της εφαρμογής. Επιπλέον τα δεδομένα που χρησιμοποιούν οι χρήστες στις υπηρεσίες τους, τα οποία είναι τα βιβλία θα πρέπει και αυτά να συμπεριληφθούν στην Βάση Δεδομένων.

Το πρόγραμμα για την κατασκευή της Βάση Δεδομένων που θα χρησιμοποιήσουμε θα είναι η Microsoft Access 2003 μιας και είναι συμβατό με την Visual .Net και στην εφαρμογή μας θα χρησιμοποιήσουμε το Microsoft SQL Server 2005 διότι διαθέτει ορισμένα εργαλεία, στα οποία περιλαμβάνονται κάποια βοηθητικά προγράμματα γραμμής διαταγών.

Κεφάλαιο 3^ο

3. Βιβλιοθήκες

3.1 Ιστορική Αναδρομή στις Βιβλιοθήκες

Τις πρώτες ενδείξεις για την εμφάνιση βιβλιοθηκών στον ευρύτερο ελληνικό χώρο συνάγουμε από πήλινες πινακίδες της Μινωικής και της Μυκηναϊκής εποχής, που βρέθηκαν σε ανασκαφές. Οι πρώτες αναφορές σε βιβλιοθήκες εμφανίστηκαν στη Μεσοποταμία γύρω στο 3000 π.Χ.

Στον ελληνικό κόσμο τα πρώτα δείγματα ύπαρξης βιβλιοθηκών ανάγονται στη Μινωική και Μυκηναϊκή περίοδο (1400-1100 π.Χ.). Πήλινες πινακίδες και λίθινες επιγραφές βρέθηκαν στις ανασκαφές της Κνωσού από το 1950, γραμμένες στη γραμμική Β γραφή, την παλαιότερη ελληνική γραφή.

Στις Μυκήνες και στην Πύλο βρέθηκαν πήλινες πινακίδες σωριασμένες σε δωμάτια των ανακτόρων και άλλες μέσα σε πιθάρια τοποθετημένα σε ράφια.

Στην Πύλο βρέθηκαν σε δωμάτιο πινακίδες με αναφορές σε στοιχεία απογραφής και στρατιωτικές στατιστικές. Συνάγεται λοιπόν ότι επρόκειτο για επίσημο κυβερνητικό αρχείο. Βρέθηκαν όμως και σε δωμάτια σπιτιών, αποδεικνύοντας ότι οι κάτοχοί τους ήταν εξοικειωμένοι με εμπορικές συναλλαγές για να τις χρησιμοποιούν

Γύρω στα μέσα του 8^{ου} αι. π.Χ. συντελέστηκε στον ελληνικό χώρο ένα σημαντικό γεγονός. Οι Έλληνες δανείστηκαν από τους Φοίνικες το αλφάβητο και το προσάρμοσαν στις ανάγκες τους. Έπρεπε όμως να υπάρξουν σχολεία, δάσκαλοι, βιβλία για την εκμάθηση της γραφής και τη διάδοση της γνώσης. Ας μην ξεχνάμε ότι τον 8^ο αι. π.Χ. είχαν συντεθεί τα Ομηρικά Έπη, τα οποία απαγγέλλονταν προφορικά από τους ραψωδούς και μεταδίδονταν στις επόμενες γενιές. Τα πρώτα δείγματα ύπαρξης συλλογών βιβλίων ήταν οι ιδιωτικές μικρές συλλογές.

Πιο πολλές πληροφορίες μπορούμε να συνάγουμε για δημιουργία βιβλιοθηκών στην Αθήνα, που ήταν το κέντρο της πνευματικής ζωής στην Ελλάδα. Αυτό αποδεικνύεται από τη λογοτεχνική παραγωγή και τη διάδοση του δράματος, με αποκορύφωμα τα έργα των τριών τραγικών ποιητών Αισχύλου, Σοφοκλή και Ευριπίδη. Την εποχή των τριών τραγικών ποιητών τα δραματικά έργα

υπήρχαν διαθέσιμα για τους θεατές με την αντιγραφή, δείγμα ότι υπήρχε παραγωγή βιβλίων από τον 5^ο αι. π.Χ. στην Αθήνα.

Ο φιλόσοφος *Αριστοτέλης* ίδρυσε στην άλλη άκρη της Αθήνας το 325 π.Χ. (έμεινε ανοικτή μέχρι το 425 μ.Χ.) τη σχολή του που αρχικά ονομάστηκε *Λύκειον* και από τα χρόνια του Θεόφραστου, μαθητή και διαδόχου του στη διεύθυνση της σχολής, ονομάστηκε *Περίπατος*. Το Λύκειον λειτουργούσε παράλληλα με την Ακαδημία. Στις σχολές αυτές διασώζονται πολλά έργα και παραδόσεις των ίδιων των ιδρυτών τους, αλλά και της παλαιότερης ελληνικής γραμματείας. Έτσι ο Αριστοτέλης κατά τον Στράβωνα δημιούργησε τη μεγαλύτερη ιδιωτική βιβλιοθήκη για την υποστήριξη του διδακτικού προγράμματος του Λυκείου.

Η πιο φημισμένη βιβλιοθήκη του αρχαίου ελληνικού κόσμου ήταν η Βιβλιοθήκη της Αλεξάνδρειας. Στόχος της βιβλιοθήκης της Αλεξάνδρειας ήταν να αποτελέσει παρακαταθήκη των ελληνικών έργων και εργαλείων έρευνας, αλλά και να συγκεντρώσει τη γραμματεία όλων των λαών (π.χ. Αιγύπτιοι, Βαβυλώνιοι) και όλων των εποχών, μεταφρασμένη στα ελληνικά. Η Βιβλιοθήκη στην πορεία της ύπαρξής της συντέλεσε στην εξέλιξη της επιστημοσύνης στην Αλεξάνδρεια και παρήγαγε κριτικές και σχολιασμένες εκδόσεις Ελλήνων συγγραφέων. Μέχρι το απόγειο της δόξας της η βιβλιοθήκη είχε συγκεντρώσει 700.000 κυλίνδρους-βιβλία.

Μετά την πτώση της ρωμαϊκής αυτοκρατορίας καταστράφηκαν πολλά βιβλία από τις βιβλιοθήκες. Οι λόγοι για τους οποίους το βιβλίο δεν διακινήθηκε στους ρυθμούς των προηγούμενων χρόνων αναζητούνται στην προσπάθεια της Εκκλησίας να επιβάλει τη χριστιανική θρησκεία, εξοβελίζοντας έργα που θεώρησε ειδωλολατρικά. Τα κέντρα πώλησης των βιβλίων στον ρωμαϊκό κόσμο έπεσαν σε αδράνεια και η υψηλή τιμή πώλησης της περγαμηνής μείωσε τον ρυθμό κατασκευής βιβλίων. Οι διπλοβιβλιοθήκες του ελληνορωμαϊκού κόσμου συνέχισαν μέχρι τον 5^ο και 6^ο αι. μ.Χ. Έδωσαν τη θέση τους στις χριστιανικές βιβλιοθήκες.

Οι πρώτες χριστιανικές κοινότητες είχαν ενοριακές βιβλιοθήκες με βιβλία θεολογικά για την υποστήριξη της πίστης, όπως στην ενορία της Ρώμης και των Ιεροσολύμων. Η πρώτη μεγάλη *αυτοκρατορική* ή *παλατινή βιβλιοθήκη*

στην Κωνσταντινούπολη φαίνεται ότι κτίστηκε επί αυτοκρατορίας του Κωνσταντίου Β', γιου του Κωνσταντίνου τον 4^ο αι. Από τον 4^ο αι. μ.Χ. ο Παχώμιος, ιδρυτής του μοναχισμού, όρισε τους κανόνες της μοναστικής ζωής, καθιερώνοντας τη *μοναστηριακή βιβλιοθήκη*. Η μελέτη ήταν υποχρεωτική για τους μοναχούς. Ωστόσο σε κάθε μοναστήρι υπήρχαν τρία είδη βιβλιοθηκών:

- a. *Η βιβλιοθήκη του καθολικού*. Η κύρια βιβλιοθήκη του μοναστηριού με λειτουργικά βιβλία.
- b. *Η αρχαική βιβλιοθήκη*. Η πρώτη αρχαική βιβλιοθήκη για την οποία έχουμε ασφαλείς πληροφορίες είναι της Μονής των Ακοιμήτων στην Κωνσταντινούπολη, που ιδρύθηκε το 420. Διέθετε βιβλιοθήκη με Συνοδικές Αποφάσεις και βιβλία για το χρηστολογικό θέμα.
- c. *Η προσωπική βιβλιοθήκη των μοναχών*. Ορισμένοι μοναχοί διατηρούσαν στις μονές τις δικές τους συλλογές βιβλίων. Αυτοί ήταν κάποιοι λόγιοι ηγούμενοι ή σημαντικοί λαϊκοί που ακολούθησαν τον μοναχικό βίο προσκομίζοντας τις βιβλιοθήκες τους στα μοναστήρια.

Φθάνοντας στον 19^ο αι. οι δημόσιες βιβλιοθήκες κάνουν την εμφάνισή τους στην Ευρώπη, με την έννοια της γενικής συλλογής την οποία οποιοσδήποτε μπορεί να επισκεφθεί και να χρησιμοποιήσει. Τον 19^ο και τον 20^ο αι. οι βιβλιοθήκες ακολουθούν τη λογοτεχνική παραγωγή και την εξέλιξη της επιστημονικής έρευνας. Υπάρχει αυξημένο ενδιαφέρον για μελέτη και έρευνα. Νέα αντίληψη υπάρχει για τις υποχρεώσεις της βιβλιοθήκης ως δημόσιο ίδρυμα από τον 20^ο αι. Πρωτοπόρος της τάσης αυτής ήταν ο Antonio Panizzi, διευθυντής του British Museum, που όρισε ως στόχο της βιβλιοθήκης τη δημιουργία ενός κέντρου για τη διάδοση της γνώσης και του πολιτισμού. Οι ειδικές βιβλιοθήκες είναι περιορισμένες σε θεματική κάλυψη και σε κατηγορίες χρηστών.

3.1.2 Στατιστικά στοιχεία βιβλιοθηκών

Η Εθνική Στατιστική Υπηρεσία της Ελλάδος εκδίδει κάθε δύο χρόνια τις *Πολιτιστικές Στατιστικές*, στις οποίες περιλαμβάνονται στατιστικά στοιχεία για τις βιβλιοθήκες και τα μουσεία της χώρας. Σύμφωνα με την τελευταία έκδοση 1995-1997 (στοιχεία 1996) στην Ελλάδα λειτουργούν 734 βιβλιοθήκες όλων των ειδών. Σε αυτές απασχολούνται 1.903 βιβλιοθηκονόμοι αποκλειστικής απασχόλησης. Το σύνολο των συλλογών ανέρχεται σε 9.143.175 τίτλους βιβλίων και 157.679 τίτλων περιοδικών (συνολικά και όχι μοναδικοί τίτλοι). Από τις 734 βιβλιοθήκες οι 581 είναι ανοικτές στο κοινό, οι 212 χρησιμοποιούν το DDC ως σύστημα ταξινόμησης, 225 χρησιμοποιούν άλλο σύστημα. Προκαλεί έκπληξη ο αριθμός των 197 βιβλιοθηκών που όπως καταγράφουν, δεν ταξινομούν το υλικό τους.

3.1.3 Ανάπτυξη και Ενίσχυση

Στην ανάπτυξη των βιβλιοθηκών στην Ελλάδα συμβάλλουν εξαιρετικά οι δύο σχολές βιβλιοθηκονομίας των ΤΕΙ (Αθήνας και Θεσσαλονίκης) και το τμήμα Βιβλιοθηκονομίας και Αρχειονομίας του Ιονίου Πανεπιστημίου. Το 1961 ιδρύθηκε η ΧΕΝ όπου εκπαιδεύτηκαν οι πρώτοι βιβλιοθηκονόμοι. Το 1977 ιδρύθηκε το ΚΑΤΕΕ Αθηνών και το 1981 το ΚΑΤΕΕ Θεσσαλονίκης. Το 1983 αυτές οι δύο Σχολές Βιβλιοθηκονομίας έγιναν ΤΕΙ. Τέλος το 1993 ιδρύθηκε η Σχολή Αρχειονομίας και Βιβλιοθηκονομίας στο Ιόνιο Πανεπιστήμιο.

Οι βιβλιοθήκες στην Ελλάδα γνώρισαν ιδιαίτερη άνθιση τα τελευταία (1994-1999, 2000-2006, 2007-) χρόνια με την οικονομική ενίσχυση που προέρχεται από τα Κοινοτικά Πλαίσια Στήριξης. Ιδιαίτερα επωφελήθηκαν οι ακαδημαϊκές βιβλιοθήκες, ωστόσο σημειώνεται αρκετή πρόοδος και στα άλλα είδη βιβλιοθηκών.

3.2. Ψηφιακές Βιβλιοθήκες

3.2.1 Γενικά

Βιβλιοθήκες υπάρχουν σε πολλές μορφές και τύπους. Σήμερα, με τη ραγδαία εξέλιξη της τεχνολογίας των πολυμέσων, δημιουργούνται και ψηφιακές βιβλιοθήκες εικόνων, ήχου και βίντεο. Είναι δυνατόν ακόμα, να λάβουμε υπόψη μας ως βιβλιοθήκες τις συλλογές που είναι αποθηκευμένες σε βάσεις δεδομένων, βάσεις γνώσεως, βάσεις κειμένων ή World Wide Web. Ωστόσο, ένα συνεχώς αυξανόμενο ποσοστό του προϋπολογισμού των βιβλιοθηκών αφιερώνεται σε ηλεκτρονικές υπηρεσίες είτε αυτές είναι cd-rom, on line κατάλογοι κοινής πρόσβασης ή on line βάσεις δεδομένων. Για την ψηφιακή βιβλιοθήκη έχουν αναφερθεί στη βιβλιογραφία πληθώρα ορισμών, περισσότερο διαδεδομένη εκ των οποίων είναι η άποψη ότι η ψηφιακή βιβλιοθήκη περιγράφει και παρουσιάζει ή παριστάνει σε ηλεκτρονική μορφή, διαθέσιμα στο διαδίκτυο ή σε cd-rom, έργα που υπάρχουν σε διάφορες μορφές (έντυπη ή ηλεκτρονική μορφή, μουσικά έργα κ.λ.π.).

Η έννοια της ψηφιακής βιβλιοθήκης συνδέεται με τις έννοιες της ηλεκτρονικής βιβλιοθήκης, της εικονικής, της υβριδικής και με άλλες έννοιες που αφορούν όμως τη δυνατότητα χρήσης τεχνολογικών μέσων και διαδικασιών, προκειμένου να διατηρηθούν και να διαδοθούν τα επιτεύγματα της ανθρώπινης γνώσης και τεχνικής. Οι ψηφιακές βιβλιοθήκες είναι βάσεις δεδομένων που έχουν ψηφιακά αντικείμενα και είναι περισσότερο καταναμημένες, έχουν περισσότερη ομοιογένεια αλλά έχουν λιγότερες δυνατότητες αναζήτησης. Είναι επίσης και μηχανές αναζήτησης, που έχουν επίσημη διαδικασία καταχώρησης, έχουν περισσότερα αλλά όχι παραπλανητικά μεταδεδομένα, είναι όμως λιγότερο καταναμημένες.

Αν θέλαμε να συγκρίνουμε τις λειτουργίες των Συμβατικών Βιβλιοθηκών σε σχέση με αυτές των Ψηφιακών, θα επικεντρωνόμασταν στα εξής σημεία: Η λειτουργία μιας συμβατικής βιβλιοθήκης διαιρείται στα στάδια της Συλλογής, της Οργάνωσης και παρουσίασης, στο στάδιο της Πρόσβασης και Ανάκτησης πληροφοριών, στο στάδιο της Ανάλυσης, Σύνθεσης και Διανομής των πληροφοριών. Σε αντιστοιχία με τις ψηφιακές

βιβλιοθήκες, το στάδιο της Οργάνωσης και Παρουσίασης ασχολείται με την ταξινόμηση και σύνταξη των πληροφοριών κατά ένα τρόπο αντιληπτό στο χρήστη. Τα ζητήματα Πρόσβασης περιλαμβάνουν τη δημιουργία του φυσικού χώρου και την οργάνωση των υλικών μέσα σε αυτόν, όπως επίσης και να δημιουργούνται κάποια συστήματα ώστε να διευκολύνεται και να γίνεται αποτελεσματικότερη η Ανάκτηση Πληροφοριών. Στο τελευταίο στάδιο περιλαμβάνονται λειτουργίες όπως ανταπόκριση σε ερωτήσεις παραπομπών, παραγωγή περιλήψεων και επινόηση προγραμμάτων διάδοσης στο κοινωνικό σύνολο.

Οι διαφορές από μια Παραδοσιακή Βιβλιοθήκη εντοπίζονται στο ψηφιακό περιεχόμενο, στην κατάργηση των φυσικών περιορισμών, στο γεγονός ότι μια ψηφιακή βιβλιοθήκη στηρίζεται στο Διαδίκτυο και στον Παγκόσμιο ιστό, στην ενσωμάτωση, την ενοποίηση και την ολοκλήρωση των υπηρεσιών.

Χαρακτηριστικά μιας ψηφιακής βιβλιοθήκης αποτελούν τα ηλεκτρονικά-ψηφιακά μορφότυπα, η δικτυωμένη πληροφορία, η προφανής ανάπτυξη της βιβλιοθήκης με πολιτική ανάπτυξης συλλογών, συστηματική δόμηση και χαρακτηρισμό δεδομένων. Ακόμη η χρήση πολιτικών ως προς την πρόσβαση, τη μονιμότητα του υλικού, τη καθοδήγηση και τις αναφορές προς τον χρήστη.

Οι ψηφιακές βιβλιοθήκες γίνονται όλο και πιο απαραίτητες διότι η ανάπτυξη των εικονικών τάξεων και των πανεπιστημίων είναι γρήγορη. Επίσης είναι πολύτιμο στοιχείο ως προς την online εταιρική εκπαίδευση, την online εύρεση πληροφορίας και την online έρευνα.

Επιπλέον, είναι αναγκαίο να τονιστεί η σημασία της ψηφιακής βιβλιοθήκης ως προς το γεγονός ότι βοηθά να μειωθεί η αντίθεση μεταξύ των Εργαστηρίων, των Τάξεων, των Βιβλιοθηκών και στο ότι στηρίζονται οι λειτουργίες της Έρευνας, της Διδασκαλίας και της εύρεσης πληροφορίας και οι υπηρεσίες της Αναζήτησης (Searching) και του Ξεφυλλίσματος (Browsing). Ως προς τον ορθό σχεδιασμό των ψηφιακών βιβλιοθηκών χρειάζεται να ληφθούν υπόψη κάποια θέματα: τα πρότυπα που θα χρησιμοποιηθούν, τα μεταδεδομένα, η αναφορά στα «αντικείμενα» που περιέχονται στο WWW, το θέμα των πνευματικών δικαιωμάτων και της ψηφιοποίησης του υλικού.

Παρά όμως τα κάποια κολλήματα που παρουσιάζονται στην ψηφιακή βιβλιοθήκη ως προς τη διαλειτουργικότητα, την κινητικότητα των αντικειμένων,

τα περίπλοκα αρχεία και το γεγονός ότι είναι δύσκολο το διαθεματικό ψάξιμο, παρέχεται μεγάλη ποσότητα πληροφοριών, άμεση πρόσβαση σε κατανεμημένη ή μη πληροφορία, μείωση του κόστους έκδοσης, πρόσβασης, συντήρησης των συλλογών και επιτυγχάνεται μια νέα λειτουργικότητα με ενσωμάτωση νέων μορφών.

Η Αυτοματοποίηση των λειτουργιών των βιβλιοθηκών είναι εξαιρετικής σημασίας ζήτημα. Οι ψηφιακοί κατάλογοι σε σχέση με τους παραδοσιακούς υποστηρίζουν πολλές λέξεις-κλειδιά, δημιουργία επερωτήσεων προς τους χρήστες και ευέλικτη αναζήτηση πληροφοριών. Έτσι μια ψηφιακή βιβλιοθήκη αποτελεί όχι μόνο αυτοματοποίηση των λειτουργιών και υπηρεσιών μιας συνηθισμένης, αλλά απαιτεί επαναπροσδιορισμό των υπηρεσιών ή ακόμα και αντικατάσταση αυτών με άλλες. Σημαντικές υπηρεσίες μιας ψηφιακής βιβλιοθήκης, είναι επίσης η πρόσκτηση του υλικού, η καταλογογράφηση, η αποθήκευση και προστασία του υλικού, η αναζήτηση και εύρεσή του μέσω του αυτοματοποιημένου καταλόγου OPAC όσο και η διακίνηση του υλικού (Rights Management).

Οι προκλήσεις της ψηφιακής βιβλιοθήκης είναι ως προς την αποθήκευση, τη διεπαφή με το χρήστη, την ταξινόμηση και την ευρετηρίαση, την ανάκτηση πληροφοριών, την παράδοση του περιεχομένου και κάποια διοικητικά ζητήματα, αλλά με τη ραγδαία εξέλιξη της τεχνολογίας πιστεύεται ότι θα ξεπεραστούν. Σήμερα ήδη υπάρχει υποδομή διαδικτύου, η έρευνα είναι αρκετά προχωρημένη, υπάρχουν εξελιγμένα πρότυπα ψηφιακών βιβλιοθηκών και κατάλληλα εργαλεία λογισμικού.

3.3 Βασικές Αρχές Βιβλιοθηκονομίας

Ο κώδικας δεοντολογίας είναι ένας κώδικας ηθικής και δεσμεύει τον βιβλιοθηκονόμο ως επαγγελματία. Δεν είναι κώδικας νομικού χαρακτήρα. Στον κώδικα αυτό έχουν υποχρέωση συμμόρφωσης τα μέλη της Ένωσης Ελλήνων Βιβλιοθηκονόμων και Επιστημόνων Πληροφόρησης.

Ο κώδικας δεοντολογίας αντιπροσωπεύει την αυτονομία του επαγγέλματος και αναφέρεται στις υποχρεώσεις του βιβλιοθηκονόμου απέναντι στον χρήστη και στο επάγγελμα.

3.3.1 Υποχρεώσεις απέναντι στο χρήστη

- Ο βιβλιοθηκονόμος εξασφαλίζει στον χρήστη την πρόσβαση στις πληροφορίες που είναι διαθέσιμες για το κοινό χωρίς κανένα περιορισμό που δεν είναι επιβεβλημένος από νόμους ή κανονισμούς.
- Η πληροφόρηση που παρέχεται από τον βιβλιοθηκονόμο είναι πλήρης, αντικειμενική δηλαδή μη καθοδηγούμενη από απόψεις, ιδέες και αξίες του βιβλιοθηκονόμου ούτε εξωτερικών κέντρων πολιτικών ή οικονομικών.
- Στην διαδικασία παροχής εξυπηρέτησης από την βιβλιοθήκη, ο βιβλιοθηκονόμος δεν λαμβάνει υπόψη φύλο, ηλικία, κοινωνική κατάσταση, θρησκευτική πίστη, πολιτικές πεποιθήσεις.
- Ο βιβλιοθηκονόμος απορρίπτει κάθε είδος λογοκρισίας στα ντοκουμέντα που συγκεντρώνει, επεξεργάζεται και διαθέτει για την πληροφόρηση του κοινού.
- Ο βιβλιοθηκονόμος εγγυάται την εμπιστευτική παροχή της πληροφορίας και των πηγών που επιθυμεί ο χρήστης.
- Ο βιβλιοθηκονόμος κατά την εξέλιξη των επαγγελματικών του υποχρεώσεων δεν πρέπει να εμπλέκεται σε συγκρούσεις συμφερόντων και να χρησιμοποιεί για δική του και μόνο ωφέλεια πληροφορίες και πόρους.

- Είναι καθήκον του βιβλιοθηκονόμου να προάγει σε επίπεδο προσωπικό και συλλογικό την αυτονομία και την αποτελεσματικότητα των υπηρεσιών της βιβλιοθήκης ως εργαλείου της δημοκρατίας.

3.3.2 Υποχρεώσεις προς το επάγγελμα

- Ο βιβλιοθηκονόμος πρέπει να τιμά το επάγγελμα με την βαθιά συναίσθηση της κοινωνικής του χρησιμότητας.
- Ο βιβλιοθηκονόμος πρέπει να διαθέτει μία ευρεία και σε βάθος επαγγελματική κατάρτιση μέσω της οποίας παρέχει στο χρήστη υπηρεσίες υψηλής ποιότητας σε σχέση με τις καθορισμένες παραμέτρους αποτελεσματικότητας και άριστης χρήσης των πηγών πληροφόρησης.
- Η επαγγελματική γνώση πρέπει να έχει συνεχή ενημέρωση και με την συμμετοχή σε βιβλιοθηκονομικές ενώσεις και οργανισμούς.
- Στην επαγγελματική του δραστηριότητα, οι σχέσεις του βιβλιοθηκονόμου με τους συναδέλφους του πρέπει να βασίζονται στις αρχές της ορθότητας, του σεβασμού και του πνεύματος συνεργασίας.

3.3.3 Υποχρεώσεις προς το υλικό και την πληροφορία

- Ο βιβλιοθηκονόμος δεσμεύεται να προωθεί την αξιοποίηση και την προστασία του υλικού και της πληροφορίας.
- Ο βιβλιοθηκονόμος δεσμεύεται να διασφαλίζει την μετάδοση της γνώσης μέσω της ορθολογικής οργάνωσης των τεκμηρίων και δρώντας με αμεροληψία και επαγγελματισμό.
- Ο βιβλιοθηκονόμος, γνώστης του αντικειμένου στα πλαίσια του οποίου, ενεργεί, δεσμεύεται να προωθεί προσωπικά και σε πλαίσια συνεργασιών, την ολοκλήρωση των διαφόρων πληροφοριακών συστημάτων και την άρση των γεωγραφικών και οργανωτικών εμποδίων που περιορίζουν την διακίνηση των πληροφοριών και των τεκμηρίων.

Η διαδικτυακή εφαρμογή μας θα είναι μία εφαρμογή η οποία δεν θα είναι μόνο προσιτή από εξειδικευμένους επαγγελματίες- βιβλιοθηκονόμους αλλά μία εφαρμογή η οποία θα μπορεί να χρησιμοποιηθεί ακόμα και από κάποιον ο οποίος έχει τις βασικές γνώσεις στους υπολογιστές. Έτσι θα είναι μία εφαρμογή εύκολα προσβάσιμη στο κοινό χρήστη, υπάλληλο της βιβλιοθήκης, χωρίς αυτός να χρειάζεται να έχει εξεζητημένες γνώσης βιβλιοθηκονομίας, αλλά χωρίς να παύει η εφαρμογή να διαθέτει όλες τις υπηρεσίες τις οποίες θα αναζητήσει ένα επαγγελματίας.

Κεφάλαιο 4^ο

4. Δανειστική Ηλεκτρονική Βιβλιοθήκη e-LIBRARY

4.1.1 Περιγραφή της Βάσης Δεδομένων

Για την δημιουργία της εφαρμογής είναι απαραίτητη η δημιουργία Βάσης Δεδομένων στην οποία θα καταχωρηθούν όλα τα στοιχεία που χρειάζονται για την σωστή και καλύτερη λειτουργία της Web υπηρεσίας.

Εφόσον η υπηρεσία θα έχει ως στόχο την αποτελεσματική λειτουργία μίας ηλεκτρονικής βιβλιοθήκης δανεισμού και διαχείρισης των βιβλίων της, τα περιεχόμενα της βιβλιοθήκης θα είναι τέτοια ώστε να εξυπηρετούν αυτό τον σκοπό. Τα βασικά στοιχεία μιας τέτοιας βάσης αναφέρονται παρακάτω:

Έτσι θα περιλαμβάνει τα στοιχεία των μελών-χρηστών της βιβλιοθήκης. Οι χρήστες θα είναι φοιτητές του τμήματος καθώς και οποιοσδήποτε άλλος έχει την εξουσιοδότηση να την χρησιμοποιήσει όπως καθηγητές, ερευνητικό προσωπικό και διάφοροι άλλοι υπάλληλοι του τμήματος. Στα στοιχεία των μελών θα συμπεριλαμβάνονται βασικά στοιχεία δηλαδή όνομα, επίθετο και ο αριθμός μητρώου, καθώς επίσης διεύθυνση κατοικίας, τηλέφωνο, ηλεκτρονική διεύθυνση

(e-mail), συνθηματικό(username) και κωδικό πρόσβασης (password), και τέλος διάφορες πρόσθετες πληροφορίες-λεπτομέρειες, που μπορεί να είναι απαραίτητες να σημειωθούν.

Επιπλέον η βάση θα εμπεριέχει πίνακα όπου θα αναγράφονται οι πληροφορίες για τα βιβλία που περιλαμβάνει η βιβλιοθήκη. Έτσι τα στοιχεία που πρέπει να υπάρχουν μέσα στην βάση δεδομένων θα είναι: Ο τίτλος του βιβλίου, ο κωδικός ISBN ο οποίος είναι ένας κωδικός που έχουν όλα τα βιβλία και είναι μοναδικός για κάθε βιβλίο, την ημερομηνία έκδοσης του βιβλίου, τον συγγραφέα και μία μικρή περίληψη.

Ακόμα, μία θα υπάρχει πίνακας με τα στοιχεία των συγγραφέων των βιβλίων. Τα στοιχεία αυτά είναι το όνομα, το επίθετο, και διάφορες επιπλέον πληροφορίες.

Τέλος για το δανεισμό των βιβλίων υπάρχει πίνακας δανεισμού όπου θα αναφέρονται η ημερομηνία δανεισμού, ημερομηνία επιστροφής του βιβλίου καθώς και το όνομα του χρήστη και το βιβλίο που έχει δανειστεί.

Για την σωστή λειτουργία της βάσης δεδομένων και κατ' επέκταση της εφαρμογής, πρέπει να επιλυθούν σημαντικά προβλήματα που δημιουργούνται αν προσπαθήσουμε να απαντήσουμε σε ερωτήσεις όπως:

- Τι κάνουμε με τα βιβλία που έχουν παραπάνω από ένα αντίτυπα;
- Τι κάνουμε με τους συγγραφείς που έχουν συμμετάσχει στην συγγραφή παραπάνω από ένα βιβλία;
- Τι κάνουμε με τα βιβλία που έχουν παραπάνω από ένα συγγραφείς;
- Πρέπει τα βιβλία να κατηγοριοποιηθούν;

Έτσι δημιουργείται η ανάγκη δημιουργίας επιπλέον πινάκων που θα συνδέονται με τους υπάρχοντες για την καλύτερη επικοινωνία των στοιχείων της Βάσης. Αυτοί οι πίνακες θα είναι:

Πίνακας για τα αντίτυπα κάθε βιβλίου, στο οποίο θα υπάρχει και ένα στοιχείο για κάποιες συμπληρωματικές επιπλέον πληροφορίες.

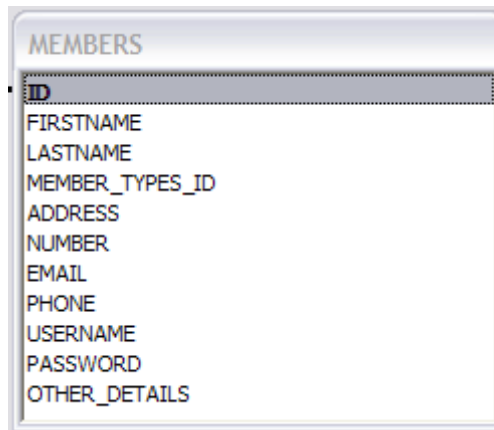
Πίνακας για την περίπτωση που ένα βιβλίο έχει παραπάνω από έναν συγγραφείς.

Επίσης πίνακα για τις κατηγορίες των βιβλίων, ώστε να διευκολυνθεί η αναζήτηση από τους χρήστες.

4.1.2 Οι Πίνακες της Βάσης.

Η Βάση Δεδομένων αποτελείται από τους παρακάτω πίνακες:

- Πίνακας MEMBERS

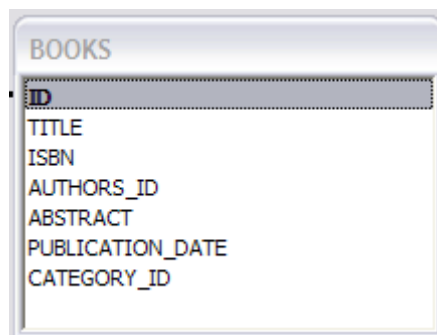


MEMBERS
ID
FIRSTNAME
LASTNAME
MEMBER_TYPES_ID
ADDRESS
NUMBER
EMAIL
PHONE
USERNAME
PASSWORD
OTHER_DETAILS

σχήμα 4.1

Στον παραπάνω πίνακα γίνεται εισαγωγή των στοιχείων ενός χρήστη-μέλους της Βιβλιοθήκης. Τα πεδία που περιλαμβάνει ο πίνακας είναι Όνομα, Επίθετο, Διεύθυνση και Αριθμός, Τηλέφωνο, e-mail, συνθηματικό (username), κωδικός (password), το είδος του μέλους δηλαδή φοιτητής, καθηγητής ή άτομο του προσωπικού, και ένα πεδίο λεπτομέρειες όπου μπορεί να γραφτεί εκεί κάποια διευκρίνηση για κάποιο συγκεκριμένο μέλος. Πρωτεύον κλειδί είναι το ID, το οποίο παίρνει αυτόματα έναν αύξοντα αριθμό στην βάση μας.

- Ο Πίνακας BOOKS



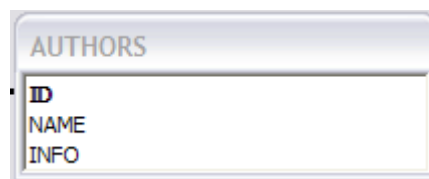
BOOKS
ID
TITLE
ISBN
AUTHORS_ID
ABSTRACT
PUBLICATION_DATE
CATEGORY_ID

σχήμα 4.2

Ο πίνακας BOOKS όπως βλέπουμε στο σχήμα 4.2, περιέχει τα πεδία TITLE, τον τίτλο του βιβλίου, το πεδίο ISBN που είναι ο κωδικός που έχουν όλα τα βιβλία, από την έκδοσή τους, το πεδίο AUTHORS_ID που παραπέμπει στο συγγραφέα του βιβλίου, το πεδίο ABSTRACT όπου γράφετε μια μικρή

περίληψη του βιβλίου, το πεδίο PUBLICATION_DATE δηλαδή την ημερομηνία έκδοσης του βιβλίου και το πεδίο CATEGORY_ID που συνδέεται με τον πίνακα CATEGORY σχήμα 4.4, την κατηγορία δηλαδή που ανήκει το βιβλίο. Πρωτεύον κλειδί είναι ID που παίρνει αυτόματα ένα αριθμό στην βάση δεδομένων.

- Ο πίνακας AUTHORS



σχήμα 4.3

Ο πίνακας AUTHORS περιλαμβάνει τις πληροφορίες για τους συγγραφείς. Περιλαμβάνει ένα πεδίο με το Όνομα του συγγραφέα και ένα πεδίο με διάφορες άλλες πληροφορίες. Επίσης περιλαμβάνει το πεδίο ID το οποίο είναι και το πρωτεύον κλειδί.

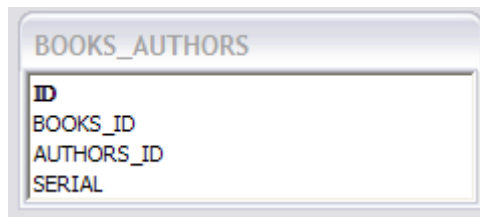
- Ο πίνακα CATEGORY



σχήμα 4.4

Ο πίνακας CATEGORY περιλαμβάνει τις πληροφορίες για τις κατηγορίες των βιβλίων. Έχει το πεδίο CATEGORY_NAME, δηλαδή το όνομα της κατηγορίας καθώς και το πεδίο PARENT_ID το οποίο χρησιμεύει για να δηλώσουμε τις σχέσεις μεταξύ των κατηγοριών. Δηλαδή ο αριθμός του πεδίου μας παραπέμπει στη κατηγορία με την οποία έχει σχέση η κατηγορία αυτή.

- Ο πίνακας BOOKS_AUTHORS

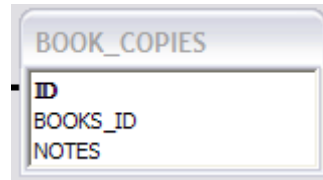


BOOKS_AUTHORS	
ID	
BOOKS_ID	
AUTHORS_ID	
SERIAL	

σχήμα 4.5

Ο πίνακας αυτός χρησιμεύει στην σχέση βιβλίων και συγγραφέων. Υπάρχει το BOOKS_ID και το AUTHORS_ID. Έτσι δηλώνουμε εδώ ποια σχέση υπάρχει μεταξύ συγγραφέα και βιβλίου. Υπάρχει ακόμα το ID ως πρωτεύον κλειδί.

- Ο πίνακας BOOKS_COPIES

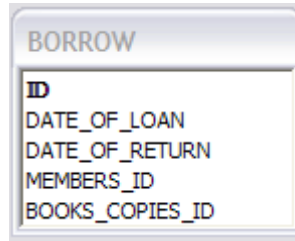


BOOK_COPIES	
ID	
BOOKS_ID	
NOTES	

σχήμα 4.6

Ο πίνακας BOOK_COPIES αποτελείται από το πεδίο BOOKS_ID που παραπέμπει στο ID του βιβλίου και το πεδίο NOTES το οποίο συμπληρώνεται με σημειώσεις που μπορεί να χρησιμεύουν στην βιβλιοθήκη. Ο πίνακας αυτός χρησιμεύει για να γνωρίζουμε το αριθμό των αντιτύπων που υπάρχουν από ένα συγκεκριμένο βιβλίο.

- Ο πίνακας BORROW

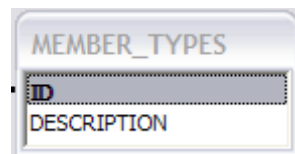


BORROW	
ID	
DATE_OF_LOAN	
DATE_OF_RETURN	
MEMBERS_ID	
BOOKS_COPIES_ID	

σχήμα 4.7

Ο πίνακας αυτός περιέχει όλες τις πληροφορίες που χρησιμεύουν για τον δανεισμό ενός βιβλίου από κάποιο μέλος. Συμπεριλαμβάνει τα πεδία DATE_OF_LOAN δηλαδή την ημερομηνία δανεισμού, DATE_OF_RETURN δηλαδή την ημερομηνία επιστροφής, το MEMBERS_ID τον κωδικό του χρήστη, και το BOOKS_COPIES_ID το αριθμό των αντιτύπων.

- Ο πίνακας MEMBER_TYPES



MEMBER_TYPES	
ID	
DESCRIPTION	

σχήμα 4.8

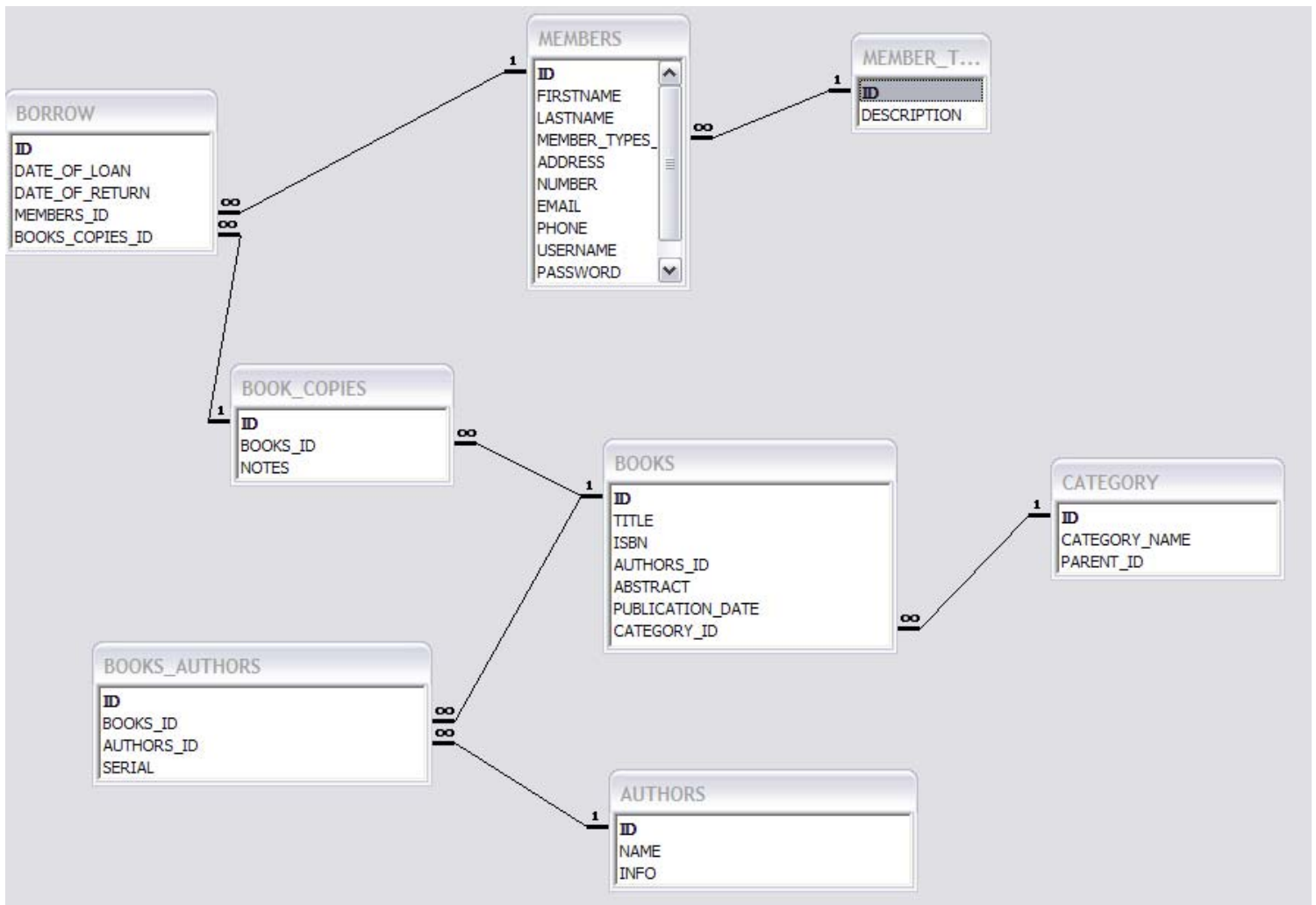
Ο παραπάνω πίνακας περιγράφει τους τύπους των μελών. Έχει ένα πεδίο ID που είναι και πρωτεύον και ένα πεδίο DESCRIPTION το οποίο περιγράφει τον τύπο των μελών. Αν είναι Φοιτητές, Καθηγητές ή Προσωπικό.

4.1.3 Διάγραμμα βάσης δεδομένων – Συσχετίσεις πινάκων

Το διάγραμμα της βάσης δεδομένων είναι μια γραφική αναπαράσταση των πινάκων και των μεταξύ τους σχέσεων (Σχήμα 4.8). Η σχεδίασή του είναι ένας καλός τρόπος για να έχουμε μια εποπτική εικόνα της δομής και των συσχετίσεων της βάσης δεδομένων. Εκτός αυτού είναι ένας βολικός τρόπος τεκμηρίωσης της βάσης, επειδή εκεί απεικονίζονται αυτόματα όποιες αλλαγές και να κάνουμε. Μια σχέση λειτουργεί με το ταίριασμα των δεδομένων σε πεδία-κλειδιά — συνήθως ένα πεδίο με το ίδιο όνομα και στους δύο πίνακες. Στις περισσότερες περιπτώσεις, αυτά τα πεδία που ταιριάζουν αποτελούν το πρωτεύον κλειδί του ενός πίνακα, το οποίο παρέχει ένα μοναδικό αναγνωριστικό για κάθε εγγραφή και ένα ξένο κλειδί για τον άλλο πίνακα. Η σχέση ένα-προς-πολλά είναι η πιο συνηθισμένη. Σε μια σχέση ένα-προς-πολλά, μια εγγραφή του Πίνακα Α μπορεί να έχει πολλές εγγραφές που ταιριάζουν στον Πίνακα Β, αλλά μια εγγραφή του Πίνακα Β έχει μόνο μία εγγραφή που της ταιριάζει στον Πίνακα Α. Σε μια σχέση πολλά-προς-πολλά, μια εγγραφή του Πίνακα Α μπορεί να έχει πολλές εγγραφές που ταιριάζουν στον Πίνακα Β και μια εγγραφή του Πίνακα Β μπορεί να έχει πολλές εγγραφές που ταιριάζουν στον Πίνακα Α. Σε μια σχέση ένα-προς-ένα, κάθε εγγραφή του Πίνακα Α μπορεί να έχει μόνο μια εγγραφή που ταιριάζει στον Πίνακα Β και κάθε εγγραφή του Πίνακα Β μπορεί να έχει μία μόνο εγγραφή που ταιριάζει στον Πίνακα Α. Αυτός ο τύπος σχέσης δεν είναι συνηθισμένος, επειδή οι περισσότερες πληροφορίες που σχετίζονται με αυτόν τον τρόπο περιλαμβάνονται συνήθως στον ίδιο πίνακα.

Στο σχήμα 4.9 παρουσιάζονται οι σχέσεις μεταξύ των πινάκων της Βάσης Δεδομένων. Έτσι μπορούμε να δούμε ότι ο Πίνακας MEMBERS συνδέεται με τον πίνακα MEMBERS_TYPES με μια σχέση ένα- προς- πολλά δηλαδή ένα τύπος χρήστη μπορεί να υπάρχει σε πολλούς χρήστες. Επίσης ο πίνακας MEMBERS συνδέεται με τον πίνακα BORROW με σχέση ένα- προς- πολλά, που σημαίνει ότι ένας χρήστης μπορεί να δανειστεί πολλά βιβλία. Ο πίνακας BORROW συνδέεται με σχέση ένα- προς- πολλά με τον πίνακα BOOK_COPIES και αυτό γιατί τα βιβλία έχουν πολλά αντίτυπα και έτσι δύο

χρήστες μπορούν να δανειστούν το ίδιο βιβλίο ταυτόχρονα λόγω των αντιτύπων. Ο πίνακας BOOKS συνδέεται με τον πίνακα BOOKS_COPIES με μία σχέση ένα- προς- πολλά και αυτό είναι λογικό αφού ένα βιβλίο υπάρχει σε πολλά αντίτυπα. Επίσης ο πίνακας BOOKS συνδέεται με μία σχέση ένα-προς-πολλά με τον πίνακα CATEGORY και έτσι πολλά βιβλία μπορούν να υπάρχουν σε μία κατηγορία. Τέλος ο πίνακας BOOK_AUTHORS συνδέεται με τον πίνακα BOOKS με μία σχέση ένα- προς- πολλά, και με τον πίνακα AUTHORS πάλι με μία σχέση ένα- προς- πολλά. Αυτό γίνεται γιατί υπάρχει η πιθανότητα ένας συγγραφέας να έχει γράψει πολλά βιβλία καθώς επίσης ένα βιβλίο να έχει γραφεί από πολλούς συγγραφείς.



σχήμα 4.9

4.1.4 Βάση Δεδομένων «ΑΝΑΚΟΙΝΩΣΕΙΣ»

Πολλές φορές είναι χρήσιμο στις διάφορες WEB εφαρμογές να υπάρχει υπομενού στο οποίο να παρουσιάζονται διάφορα νέα ή διάφορες ανακοινώσεις οι οποίες να είναι πιο εύκολα προσβάσιμες στους χρήστες. Για την δημιουργία ενός τέτοιου υπομενού στην εφαρμογή μας είναι απαραίτητο να δημιουργηθεί μία Βάση Δεδομένων η οποία να περιέχει τέτοια δεδομένα.

Αναγνωριστικό	TEXT
1	Ανακοίνωση Σεμιναρίου από την Βιβλιοθήκη του ΤΕΙ 11-12-2007.
2	Ανακοίνωση εισαγωγής νέου υλικού.
3	Ημερίδα "Νέες Τεχνολογίες" 12-11-2007.

* ματη Αρίθμηση)

σχήμα 4.10

Έτσι η Βάση ΑΝΑΚΟΙΝΩΣΕΙΣ που αποτελείται από έναν πίνακα στον οποίο καταχωρούνται όλα τα δεδομένα τα οποία θέλουμε να εμφανίζονται σε σημεία μέσα στην εφαρμογή μας. Βέβαια αυτά τα δεδομένα θα είναι ακόμα πιο χρήσιμα αν σε μελλοντική χρήση η εφαρμογή γίνει προσβάσιμη μέσω διαδικτύου, που σε αυτήν την περίπτωση πληθαίνουν οι χρήστες και οι απαιτήσεις τους είναι περισσότερες.

4.2 Διεπαφή Χρήστη - User Interface

Ο όρος διεπαφή *χρήστη* (*User Interface*) ορίζει το τμήμα με το οποίο αλληλεπιδρά ο χρήστης. Υπάρχουν πολλοί τύποι: διεπαφές γραμμής διαταγών (εντολών), διεπαφές οδηγούμενες από μενού, και διεπαφές γραφικών. Αναλύοντας το παραπάνω ορισμό και προσπαθώντας να τον κάνουμε πιο συγκεκριμένο, αναφερόμενοι στις ψηφιακές βιβλιοθήκες, θα λέγαμε πως το user interface είναι το περιβάλλον με το οποίο έχει πρόσβαση και αλληλεπιδρά ο χρήστης με την ψηφιακή βιβλιοθήκη και το περιεχόμενό της. Η δημιουργία ενός user interface είναι σημαντική υπόθεση αν θέλουμε η ψηφιακή βιβλιοθήκη να μην είναι απλά ένα αποθετήριο δεδομένων. Είναι ο χώρος όπου μπορεί ο χρήστης να πραγματοποιήσει διάφορες δραστηριότητες. Το user interface είναι ο ενδιάμεσος κρίκος ανάμεσα στους χρήστες και σε μια ψηφιακή βιβλιοθήκη. Ένα καλά σχεδιασμένο user interface μπορεί να βελτιώσει την επικοινωνία του χρήστη με την ψηφιακή βιβλιοθήκη και να μειώσει την προσπάθεια που χρειάζεται για να βρεθούν συγκεκριμένα αντικείμενα (κείμενα, ήχος, εικόνα) στην οθόνη.

4.2.1 Σχεδίαση της οθόνης (screen design)

Η σχεδίαση της οθόνης αναφέρεται στον τρόπο με τον οποίο παρουσιάζεται η πληροφορία σ' αυτήν. Σε γενικές γραμμές, δεν ενδιαφέρει μόνο τι τοποθετούμε πάνω στην οθόνη αλλά και ο τρόπος παρουσίασης. Για παράδειγμα ένα γραφικό user interface προσφέρει πιο αποτελεσματική αλληλεπίδραση μεταξύ χρηστών και ψηφιακών βιβλιοθηκών. Ο τρόπος με τον οποίο η πληροφορία είναι τοποθετημένη στην οθόνη μπορεί να επηρεάσει την αλληλεπίδραση του χρήστη με την ψηφιακή βιβλιοθήκη. Για παράδειγμα, τόσα πολλά σημεία κάνουν την εξερεύνηση δύσκολη, ενώ άσχημα απεικονιζόμενα κουμπιά και εικόνες μπορεί να δημιουργήσουν σύγχυση και παρανόηση.

4.2.2 Πλοήγηση (navigation)

Η πλοήγηση είναι εύκολη όταν οι χρήστες μπορούν να κινηθούν γύρω από το σύστημα. Όσο ο αριθμός της πληροφορίας αυξάνεται, η δομή της αποθήκευσης γίνεται περισσότερο περίπλοκη. Οι χρήστες μπορεί εύκολα να χαθούν μέσα στην εφαρμογή προσπαθώντας να ανακτήσουν πληροφορία από αυτές.

4.2.3 Τρόποι Παρουσίασης Ψηφιακών Δεδομένων

Ιεραρχικά δομημένα κείμενα

Λέγοντας ότι τα κείμενα είναι ιεραρχικά δομημένα (Hierarchically structured documents), εννοούμε ότι υπάρχουν αρχεία τα οποία περιέχονται σε καταλόγους ή φακέλους καθένας από τους οποίους μπορεί να περιέχει και άλλα αρχεία.

4.2.4 Διαδικασίες Πρόσβασης και Αναζήτησης

Η αναζήτηση (Searching) αφορά τη διαδικασία κατά την οποία ο χρήστης προσπαθεί να εντοπίσει την πληροφορία που τον ενδιαφέρει με διάφορους όρους (λέξεις κλειδιά) και την πληροφορία που του δίνεται ως απάντηση.

Τύποι ερωτημάτων:

- Boolean ερωτήματα

Μέσα σε ένα Boolean ερώτημα οι όροι συνδυάζονται χρησιμοποιώντας τους συνδέσμους ΚΑΙ (AND), Ή (OR) και ΟΧΙ (NOT) και οι απαντήσεις στο ερώτημα είναι κομμάτια κειμένου τα οποία ικανοποιούν τη οριζόμενη συνθήκη. Το ΚΑΙ είναι το πιο συνηθισμένο σε ένα τύπο ερωτήματος Boolean.

- Ranked ερωτήματα

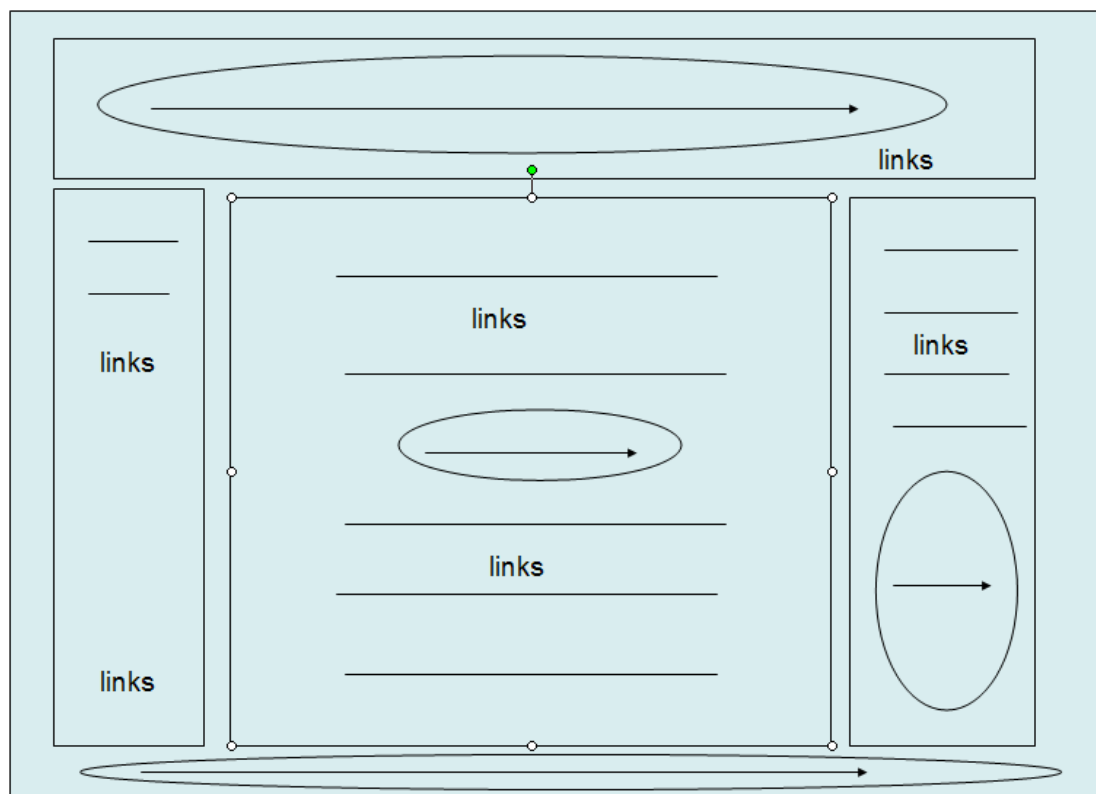
Σε μια ranked ανάλυση η λίστα των προς αναζήτηση όρων αντιμετωπίζεται σαν ένα μικρό αυτόνομο κείμενο. Τα κομμάτια του κειμένου που είναι περισσότερο «όμοια» με αυτά που αναζητούνται τοποθετούνται ανά βαθμίδα σύμφωνα με το βαθμό που αυτά ταιριάζουν.

Διαφορετικά interfaces ερωτημάτων μπορεί να είναι διαθέσιμα ανάλογα με τα διαφορετικό έργο που αυτά προσφέρουν δίνοντας την δυνατότητα στους χρήστες να επιλέξουν διάφορους τρόπους αναζήτησης.

Interfaces φυλλομέτρησης με βάση αλφαβητικές λίστες και ημερομηνία. Σε πολλές περιπτώσεις είναι χρήσιμο να γίνεται αναζήτηση με λίστες αλφαβητικές δηλαδή εμφανίζονται σε μέρος της σελίδας τα γράμματα του αλφαβήτου και κάθε ένα εμφανίζει αποτελέσματα σε σχέση με αυτό. Το ίδιο μπορεί να γίνει και με φυλλομέτρηση με ημερομηνία.

Visual user interface ή graphical user interface είναι η διεπαφή χρήστη με γραφικά. Είναι ένα εικαστικό περιβάλλον υπολογιστή που αναπαριστά προγράμματα, αρχεία και επιλογές με εικόνες γραφικών όπως εικονίδια, μενού και πλαίσια διαλόγου στην οθόνη. Ο χρήστης μπορεί να επιλέξει και να ενεργοποιήσει τις επιλογές αυτές δείχνοντάς τες και πατώντας το πλήκτρο του ποντικιού, ή συχνά με το πληκτρολόγιο.

Το παρακάτω σχήμα δείχνει την μορφή της εφαρμογής μας. Βλέπουμε ότι πάνω υπάρχει το λογότυπο της εφαρμογής. Ακριβώς από κάτω υπάρχει ένα navigation tool το οποίο μας εμφανίζει το σημείο στο οποίο είμαστε μέσα στην εφαρμογή μας και μπορούμε να πλοηγηθούμε μέσα σε αυτή. Στα αριστερά υπάρχει ένα ακόμα navigation tool pop up το οποίο εμφανίζει σε ποια σημεία μπορούμε να πάμε. Στο κέντρο υπάρχει το κυρίως μέρος της εφαρμογής στο οποίο εμφανίζονται τα περιεχόμενα της εφαρμογής. Τέλος στα δεξιά υπάρχει χώρος για την εισαγωγή διαφόρων εργαλείων. Υπάρχουν φωτογραφίες και λογότυπα και links σε πολλά σημεία μέσα στην εφαρμογή. Στο κάτω μέρος υπάρχει ένα ακόμα λογότυπο.



σχήμα 4.11

Και στο παρακάτω σχήμα 4.12 βλέπουμε την πραγματική μορφή της εφαρμογής.

e-LIBRARY
ΗΛΕΚΤΡΟΝΙΚΗ ΔΑΝΕΙΣΤΙΚΗ ΒΙΒΛΙΟΘΗΚΗ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ

Αρχική

Εγγραφή

Εισόδος

Αρχική

Welcome

Καλωσήλθατε στην ηλεκτρονική δανειστική βιβλιοθήκη του Ε.Π.Π. Για να μπορείτε να χρησιμοποιήσετε τις διάφορες υπηρεσίες πρέπει να είστε μέλος.

Τελευταία Νέα

- Την Κυριακή **20-09-2007** προστέθηκαν στην Βαση δεδομένων του **e-LIBRARY** **20** νέοι τίτλοι.[Περισσότερα](#)
- Από την Δευτέρα **30** Νοεμβρίου θα λειτουργεί στο **site** νέα Υπηρεσία **Newsletter**. [Περισσότερα](#)

Χρήσιμοι Σύνδεσμοι

Παρακάτω υπάρχουν κάποιοι χρήσιμοι σύνδεσμοι που θα σας βοηθήσουν να πλοηγηθείτε καλύτερα στην ιστοσελίδα.

- [Βιβλία - Books](#)
- [Υπηρεσίες - Services](#)
- [Σύνδεσμοι - Links](#)

Νέα - News

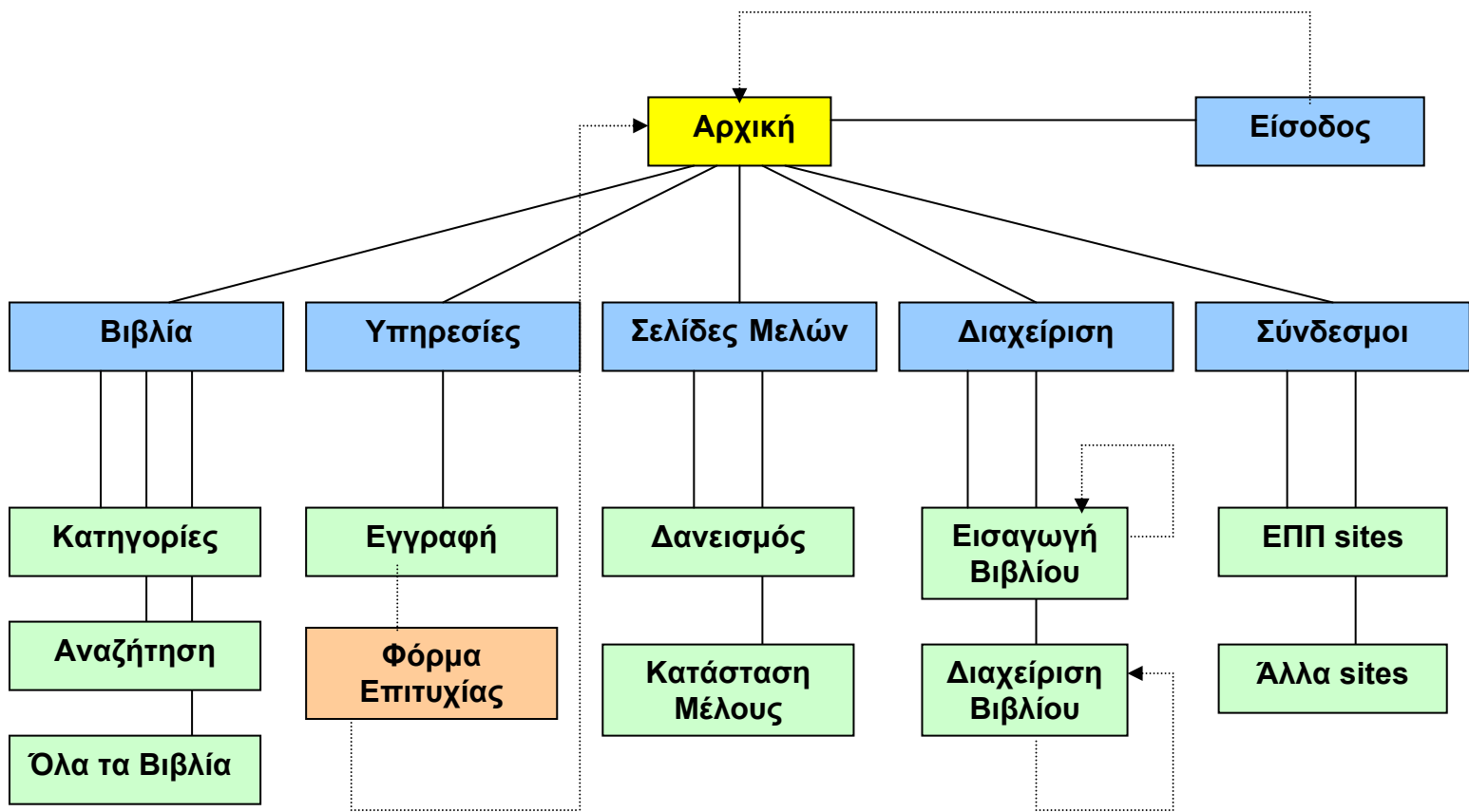
- Ανακοίνωση Σεμιναρίου από την Βιβλιοθήκη του ΤΕΙ 11-12-2007.
- Ανακοίνωση εισαγωγής νέου υλικού.
- Ημερίδα "Νέες Τεχνολογίες" 12-11-2007.

Copyright George Papadanelis
2007

Σχήμα 4.12

4.3 Site Map

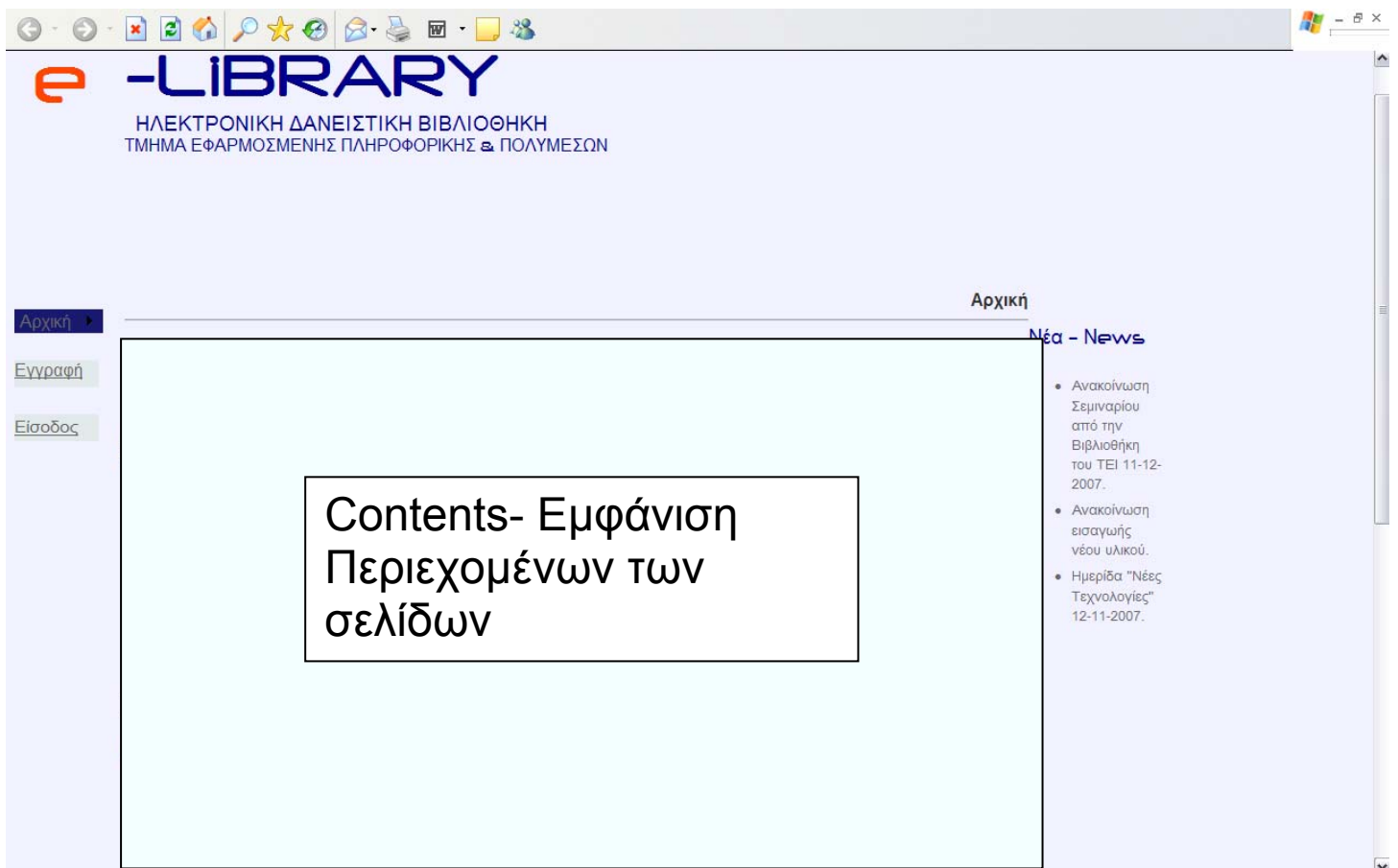
Στο παρακάτω σχήμα παρουσιάζονται οι σύνδεσμοι μεταξύ των σελίδων της εφαρμογής και πως αλληλοεπιδρούν μεταξύ τους



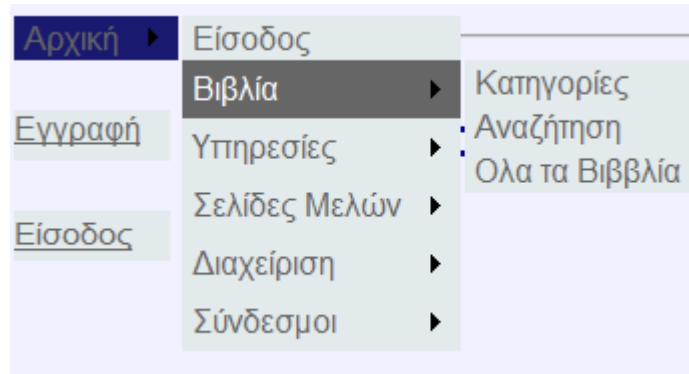
Σχήμα 4.13

4.3.1 Master Page

Παρατηρούμε ότι ο χρήστης αρχικά μπαίνει στην αρχική σελίδα. Η Αρχική σελίδα περιλαμβάνει όλα εκείνα τα στοιχεία τα οποία χρειάζεται ο χρήστης, για να μπορεί να πλοηγηθεί σε όλη την εφαρμογή. Στην εφαρμογή υπάρχει μία Master Page η οποία εμφανίζει τα στοιχεία πλοήγησης, τα λογότυπα και τα διάφορα links τα οποία φαίνονται σε όλη την εφαρμογή. Στο σχήμα 4.14 μπορούμε να παρατηρήσουμε όλα αυτά τα περιεχόμενα.



Περιγραφή Master Page. Μπορούμε να παρατηρήσουμε στην κορυφή της σελίδα ότι υπάρχει το λογότυπο της εφαρμογής e-LIBRARY. Αμέσως μετά υπάρχουν τα menu πλοήγησης. Στα αριστερά υπάρχει ένα αναδιπλωμένο menu το οποίο παρουσιάζει όλα τα links στα οποία μπορεί να μεταβεί ο χρήστης αμέσως κλικάρωντας το κάθε link. Στο σχήμα 4.15 βλέπουμε το συγκεκριμένο menu.



σχήμα 4.15

Αναλόγως σε ποια σελίδα είμαστε παρουσιάζεται και με διαφορετικό χρώμα ώστε να ξέρει ο χρήστης που ακριβώς βρίσκεται. Ακριβώς από κάτω από το menu υπάρχουν δύο links με τα οποία ο χρήστης μπορεί αμέσως να μεταβεί σε δύο πολύ χρήσιμες σελίδες, την Εγγραφή και την Είσοδο. Την Εγγραφή για την εγγραφή κάποιου μέλους στην εφαρμογή και την Είσοδο για την είσοδο κάποιου γραμμένου μέλους. Τα συγκεκριμένα links θα τα αναλύσουμε παρακάτω.

Επιπλέον στην Master Page υπάρχει ένα SiteMapPath το οποίο βοηθά τον χρήστη στο να γνωρίζει που ακριβώς βρίσκετε μέσα στην εφαρμογή και να μπορεί να μετακινείται σε προηγούμενες σελίδες. Το συγκεκριμένο path παίρνει πληροφορίες από το menu και τις εμφανίζει στην κορυφή της σελίδας. Στο σχήμα 4.16 μπορούμε να δούμε το path.

[Αρχική](#) : [Σύνδεσμοι](#) : ΕΠΠ sites

σχήμα 4.16

Στα δεξιά της σελίδας εμφανίζεται ένα panel με την ονομασία Νέα, το οποίο παίρνει πληροφορίες από μία Βάση Δεδομένων (Ανακοινώσεις) και εμφανίζει διάφορα νέα και Ανακοινώσεις τα οποία είναι χρήσιμα στον χρήστη, όπως φαίνεται στο σχήμα 4.17.

Νέα - News

- Ανακοίνωση Σεμιναρίου από την Βιβλιοθήκη του ΤΕΙ 11-12-2007.
- Ανακοίνωση εισαγωγής νέου υλικού.
- Ημερίδα "Νέες Τεχνολογίες" 12-11-2007.

σχήμα 4.17

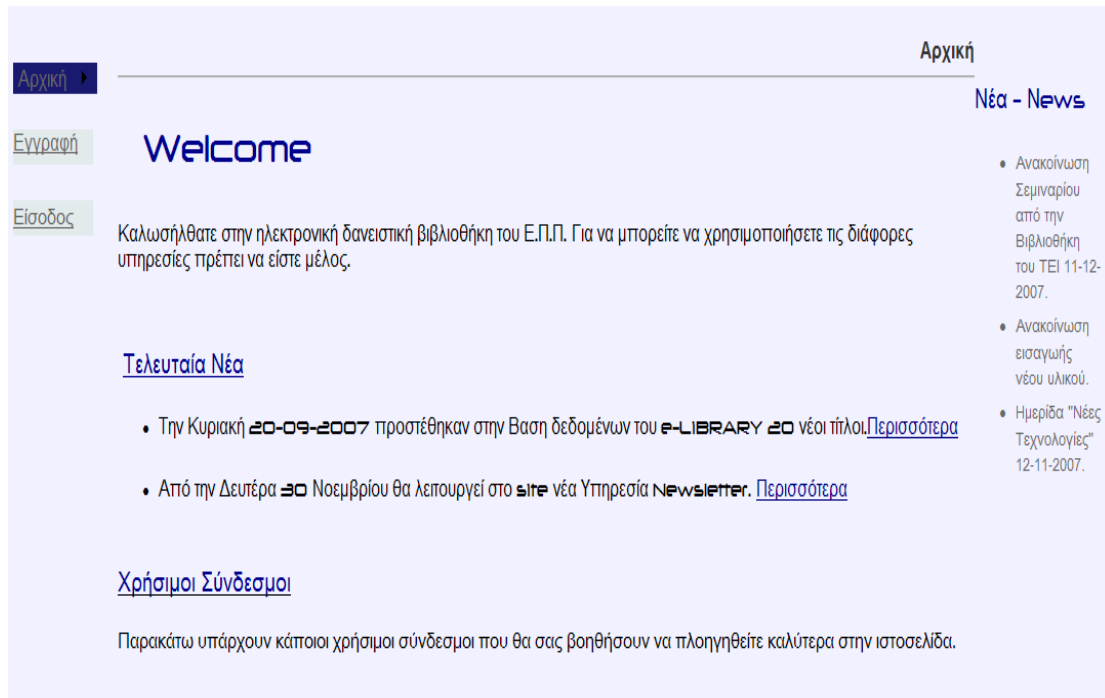
Στο κέντρο της σελίδας εμφανίζονται τα περιεχόμενα της εκάστοτε σελίδας, στην οποία βρισκόμαστε. Ο χρήστης κάνοντας κλικ σε κάποιο link βλέπει τα περιεχόμενα του σε αυτό το σημείο.

Τέλος στο κάτω μέρος της σελίδας υπάρχει το λογότυπο Copyright.

Copyright George Papadanelis
2007

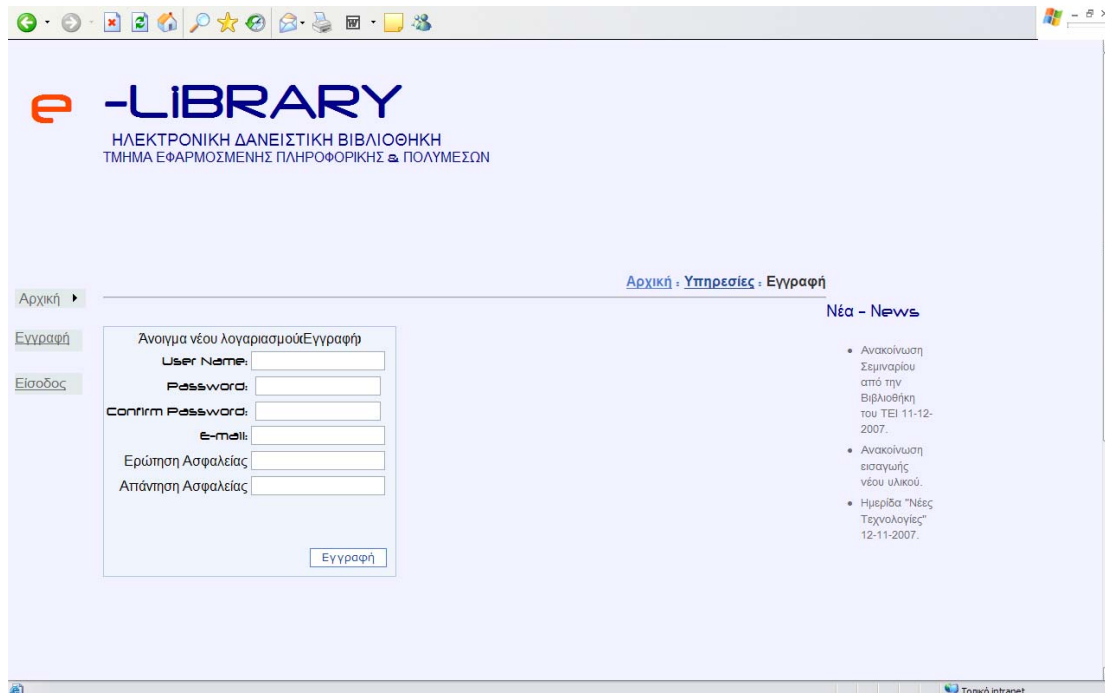
4.3.2 Αρχική Σελίδα

Στην Αρχική Σελίδα μπορούμε να παρατηρήσουμε ένα μήνυμα καλωσορίσματος καθώς επίσης και διάφορες πληροφορίες σχετικά με την εφαρμογή. Παρατηρούμε ότι υπάρχουν διάφορα links τα οποία μας ενημερώνουν, σχετικά με τις τελευταίες εξελίξεις στην εφαρμογή καθώς και διάφορα χρήσιμα links με τα οποία μπορούμε να μεταβούμε σε διάφορα σημεία της εφαρμογής μας. Στο σχήμα 4.19 εμφανίζεται η αρχική σελίδα.



σχήμα 4.19

4.3.3 Εγγραφή



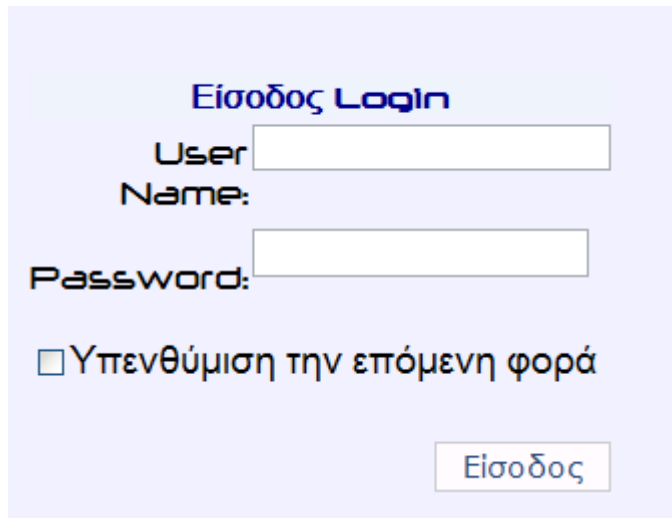
σχήμα 4.20

Ο χρήστης πατώντας το link Εγγραφή μεταφέρεται στην φόρμα εγγραφής, για την εφαρμογή. Ο χρήστης θα πρέπει να εισάγει μια σειρά από στοιχεία τα οποία χρειάζονται για να γίνει σωστή η εγγραφή στην υπηρεσία. Ζητείται

username, password, e-mail, και όλα τα στοιχεία τα οποία είναι απαραίτητα για την ορθή και ασφαλή εγγραφή κάποιου ώστε να μπορεί να κάνει χρήση των υπηρεσιών της εφαρμογής. Μετά την εισαγωγή των στοιχείων ο χρήστης πατάει το κουμπί Εγγραφή και αν είναι σωστά όλα, επιστρέφεται φόρμα που ενημερώνει τον χρήστη ότι η Εγγραφή του έγινε με επιτυχία. Σε περίπτωση που ο χρήστης δεν εισάγει κάποιο στοιχείο τότε η εφαρμογή τον ενημερώνει ότι δεν το έχει εισάγει με τα γνωστά σημεία «*» δίπλα από το συγκεκριμένο πεδίο. Επίσης αν υπάρχει κάποιο στοιχείο ίδιο μέσα στην Βάση δεδομένων, όπως κάποιο ίδιο username ή μία ηλεκτρονική διεύθυνση τότε εμφανίζεται ένα μήνυμα στην φόρμα που ενημερώνει τον χρήστη για το πρόβλημα που έχει παρουσιαστεί. Υπάρχουν validation tools τα οποία ελέγχουν την ορθότητα των στοιχείων που εισάγει ο χρήστης για την σωστή λειτουργία της εφαρμογής.

4.3.4 Είσοδος

Για να εισέλθει ένας χρήστης μέσα στην εφαρμογή θα πρέπει να κάνει Log in σε αυτή. Από το link Είσοδος μεταφέρεται στην φόρμα Login στην οποία του γίνεται ερώτηση, ώστε να εισάγει το συνθηματικό (username) και το κωδικό του (password). Αφού γίνει ο έλεγχος των στοιχείων είτε μπαίνει κανονικά μέσα στην εφαρμογή και χρησιμοποιεί της υπηρεσίες είτε εμφανίζεται μήνυμα λάθους για τα στοιχεία του. Αναλόγως ποια στοιχεία είναι λάθος ενημερώνεται ο χρήστης. Όταν ο χρήστης συνδεθεί εμφανίζεται το username του στην κορυφή της σελίδας. Επίσης υπάρχει check box για το αν επιθυμεί ο χρήστης η εφαρμογή να θυμάται τα στοιχεία του για μελλοντική επίσκεψη.



Είσοδος Login

User Name:

Password:

Υπενθύμιση την επόμενη φορά

Με την χρήση του εργαλείου Login View έχουμε την δυνατότητα να εμφανίζουμε τα στοιχεία του χρήστη σε όποιο σημείο της εφαρμογής μας θέλουμε καθώς επίσης με το εργαλείο LoginStatus μπορούμε να βλέπουμε την κατάσταση του χρήστη, αν είναι δηλαδή συνδεδεμένος ή όχι.

4.3.5 Βιβλία

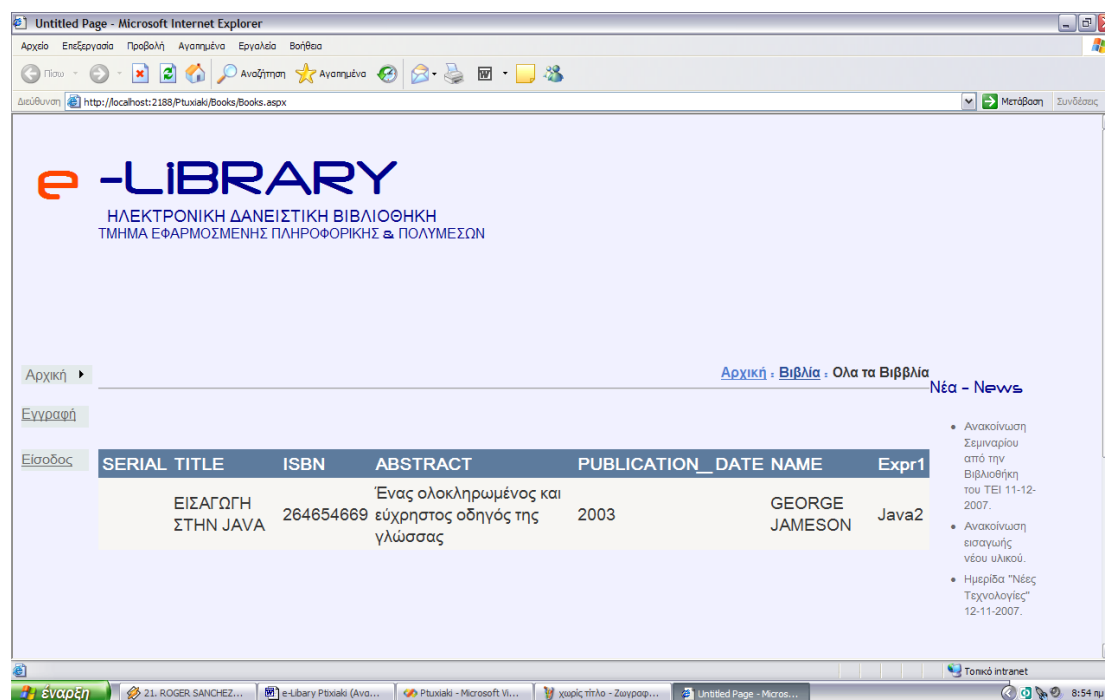
Μπαίνοντας ο χρήστης στην σελίδα Βιβλία, στην αρχή εμφανίζεται η Default σελίδα η οποία παρουσιάζει τα περιεχόμενα που υπάρχουν μέσα στην σελίδα αυτή και τι μπορεί ο χρήστης να κάνει από εδώ. Μετά από ένα μήνυμα καλωσορίσματος υπάρχουν τα διάφορα link που εμφανίζονται και στο αναδιπλωμένο menu.



Έτσι ο χρήστης μπορεί να μπει μέσα στη σελίδα Κατηγορίες Βιβλίων και να δει τα βιβλία που υπάρχουν χωρισμένα σε συγκεκριμένες κατηγορίες. Ο χρήστης επιλέγει μία συγκεκριμένη κατηγορία επιλέγοντας από μία λίστα και εμφανίζονται από κάτω τα βιβλία της συγκεκριμένης Κατηγορίας σε Grid View Data Store. Από ένα DropDownList με περιεχόμενα τις κατηγορίες των Βιβλίων μέσα από την Βάση Δεδομένων επιλέγετε η κατηγορία και με Auto refresh αλλάζουν τα δεδομένα που περιέχονται στο GridView. Τα περιεχόμενα του GridView συνδέονται με την Βάση Δεδομένων με τον Πίνακα BOOKS_AUTHORS, με τα δεδομένα δηλαδή των Βιβλίων.

Στην Σελίδα Αναζήτηση ο χρήστης μπορεί να κάνει αναζήτηση ενός Βιβλίου. Πληκτρολογώντας κάποια στοιχεία κάποιου βιβλίου τα οποία ο χρήστης θέλει να αναζητήσει και πατώντας το κουμπί αναζήτησης εμφανίζονται όλα τα βιβλία που έχουν σχέση με τις λέξεις κλειδιά. Η αναζήτηση κάνει search στην Βάση Δεδομένων ψάχνοντας να βρει όλα τα στοιχεία εκείνα που έχουν σχέση με τις λέξεις που εισάγαμε στο Text Box. Από τα στοιχεία που εμφανίζονται ο χρήστης μπορεί να επιλέξει όποιο από αυτό θέλει και να το προσθέσει στο λογαριασμό του δηλαδή να το δανειστεί.

Τέλος ο χρήστης επιλέγοντας το link Όλα τα Βιβλία μπορεί να δει όλα τα Βιβλία τα οποία υπάρχουν στην Βάση Δεδομένων. Με την βοήθεια ενός GridView και συνδέοντας το με την Βάση Δεδομένων μας καθώς και με την χρήση Μεθόδων μπορούμε να εμφανίσουμε όλα τα δεδομένα. Στο παρακάτω σχήμα φαίνονται τα δεδομένα.



σχήμα 4.23

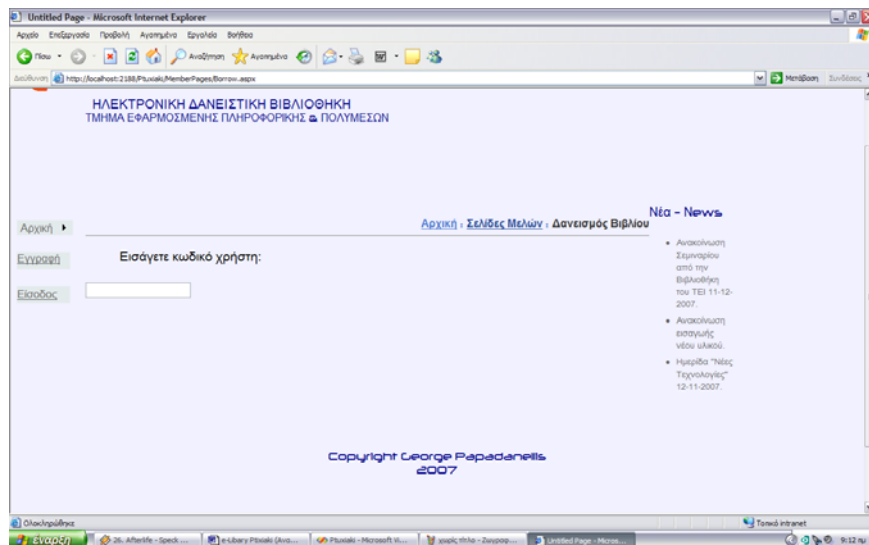
4.3.6 Υπηρεσίες

Στην σελίδα αυτή εμφανίζονται οι Υπηρεσίες που υπάρχουν στην Εφαρμογή. Εδώ υπάρχει η υπηρεσία Εγγραφή με την οποία μπορούμε να εισάγουμε ένα χρήστη. Ένας χρήστης αφού γραφτεί στην εφαρμογή μπορεί να χρησιμοποιήσει την εφαρμογή πάντα με τον χειρισμό της από κάποιο υπάλληλο της βιβλιοθήκης. Έτσι ο κάθε χρήστης ανεξαρτήτως της ιδιότητάς του μπορεί να δανειστεί ένα βιβλίο από την δανειστική Βιβλιοθήκη. Ο χρήστης επιλέγει το βιβλίο από την Βιβλιοθήκη παραδείγματος χάριν από την Βιβλιοθήκη του ΤΕΙ και το πηγαίνει στον υπάλληλο ο οποίος έχει την εφαρμογή μπροστά του. Ο υπάλληλος βάζει τα στοιχεία του χρήστη στην εφαρμογή και τα στοιχεία του Βιβλίου και έτσι καταχωρείτε στην κατάσταση

του χρήστη το Βιβλίο. Ο χρήστης ενημερώνεται για το πότε πρέπει να επιστρέψει το βιβλίο. Όταν τώρα κάποιο μέλος επιστρέφει ένα βιβλίο ο υπάλληλος βγάζει από τον λογαριασμό του χρήστη το συγκεκριμένο βιβλίο. Οι λειτουργίες αναζήτηση και εμφάνιση των βιβλίων χρησιμεύουν σε περιπτώσεις που κάποιο μέλος ζητήσει διάφορες πληροφορίες σχετικά με κάποιο βιβλίο ή ψάχνει κάποιο βιβλίο. Το σχήμα 4.20 παρουσιάζει την φόρμα εισαγωγής στοιχείων για την εγγραφή κάποιου μέλους.

4.3.7 Σελίδες Μελών

Στις σελίδες αυτές ο υπάλληλος μπαίνει για να ελέγξει την κατάσταση κάποιου μέλους εισάγοντας τα στοιχεία του. Έτσι με την εισαγωγή των στοιχείων ενός χρήστη εμφανίζεται η κατάστασή του και ο υπάλληλος μπορεί είτε να προσθέσει κάποιο βιβλίο που θέλει να δανειστεί στον λογαριασμό του είτε να διαγράψει κάποιο βιβλίο από το λογαριασμό του.



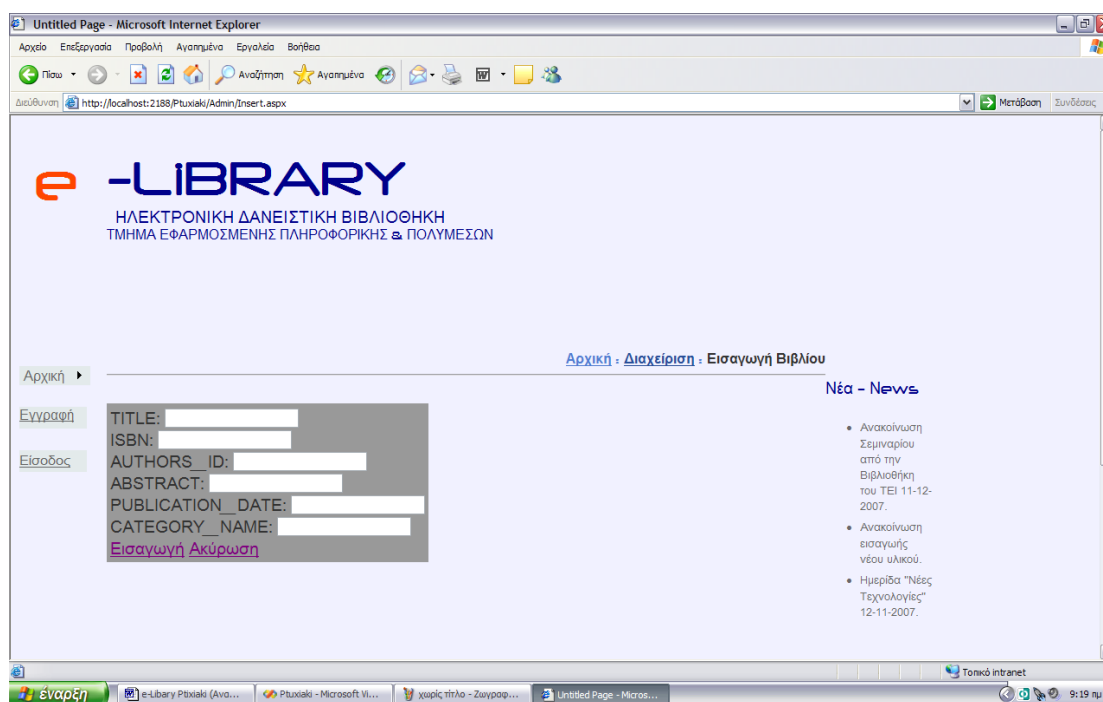
Στο παραπάνω σχήμα φαίνεται η σελίδα Δανεισμός Βιβλίου.

4.3.8 Διαχείριση

Στην Διαχείριση ο υπάλληλος έχει την δυνατότητα μέσα από δύο επιλογές να επιλέξει αν θέλει να κάνει Εισαγωγή ενός νέου τίτλου στην Βάση Δεδομένων ή αν θέλει να αλλάξει κάποιες πληροφορίες σε κάποιο υπάρχον τίτλο.

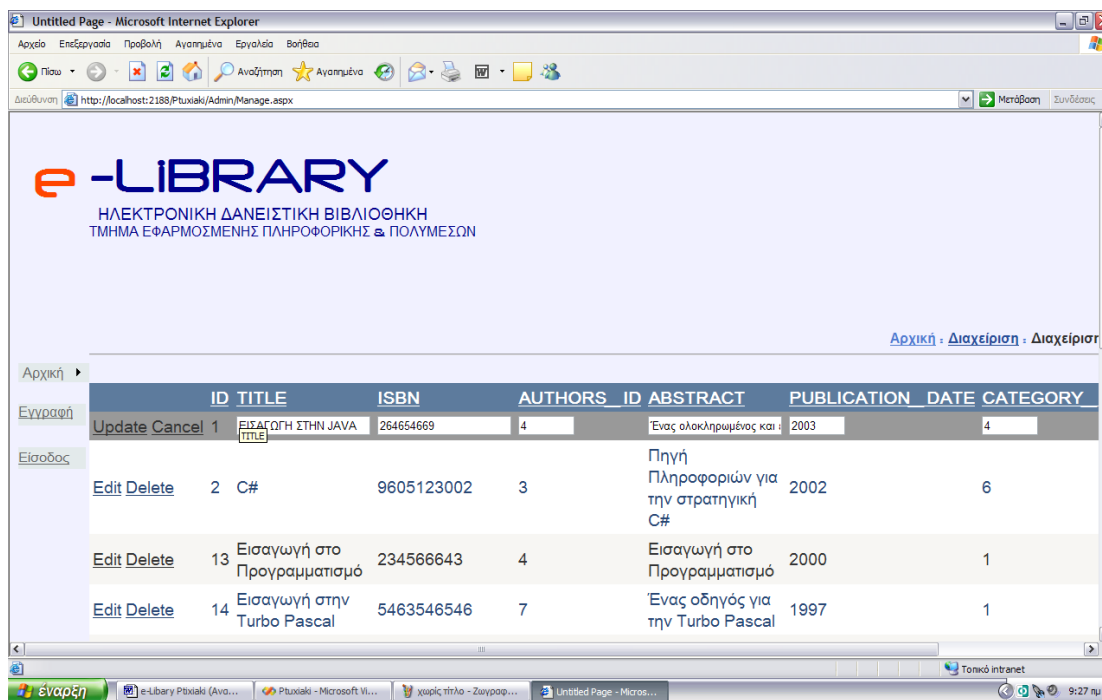
Έτσι επιλέγοντας την επιλογή Εισαγωγή Βιβλίου του εμφανίζεται η φόρμα εισαγωγής νέου τίτλου που περιέχει όλες τις πληροφορίες που πρέπει να εισάγει ο υπάλληλος για να γίνει η σωστή εισαγωγή στην Βάση Δεδομένων.

Μετά την εισαγωγή των διαφόρων στοιχείων υπάρχουν οι επιλογές Εισαγωγή πατώντας την γίνεται η εισαγωγή του βιβλίου ή να κάνει Ακύρωση ώστε να επιστρέψει στην άδεια φόρμα εισαγωγής.



Επιλέγοντας την επιλογή Διαχείριση Βιβλίου ο υπάλληλος μπορεί να αλλάξει όποια πληροφορία επιθυμεί σε κάποιον τίτλο επιλέγοντας το Edit δίπλα σε κάποιο βιβλίο. Στην σελίδα εμφανίζονται οι τα βιβλία σε μορφή Grid View με τις επιλογές Edit και Delete δίπλα σε κάθε τίτλο. Επιλέγοντας το Edit εμφανίζονται όλα τα στοιχεία του βιβλίου σε Επεξεργάσιμη μορφή. Ο χρήστης αλλάζει όποια πληροφορία θέλει και πατάει το Update για να γίνει η ανανέωση στην Βάση Δεδομένων. Επίσης υπάρχει η επιλογή Delete για την διαγραφή ενός βιβλίου.

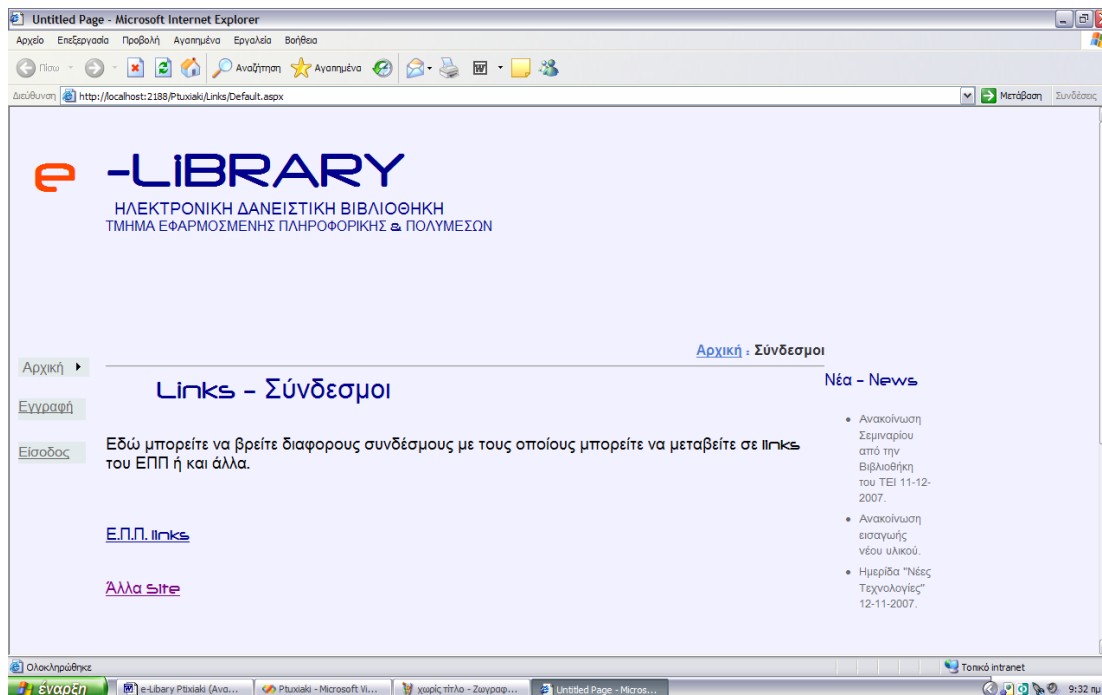
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
e-LIBRARY
ΗΛΕΚΤΡΟΝΙΚΗ ΔΑΝΕΙΣΤΙΚΗ ΒΙΒΛΙΟΘΗΚΗ



Στο σχήμα μπορούμε να δούμε όλες τις παραπάνω επιλογές.

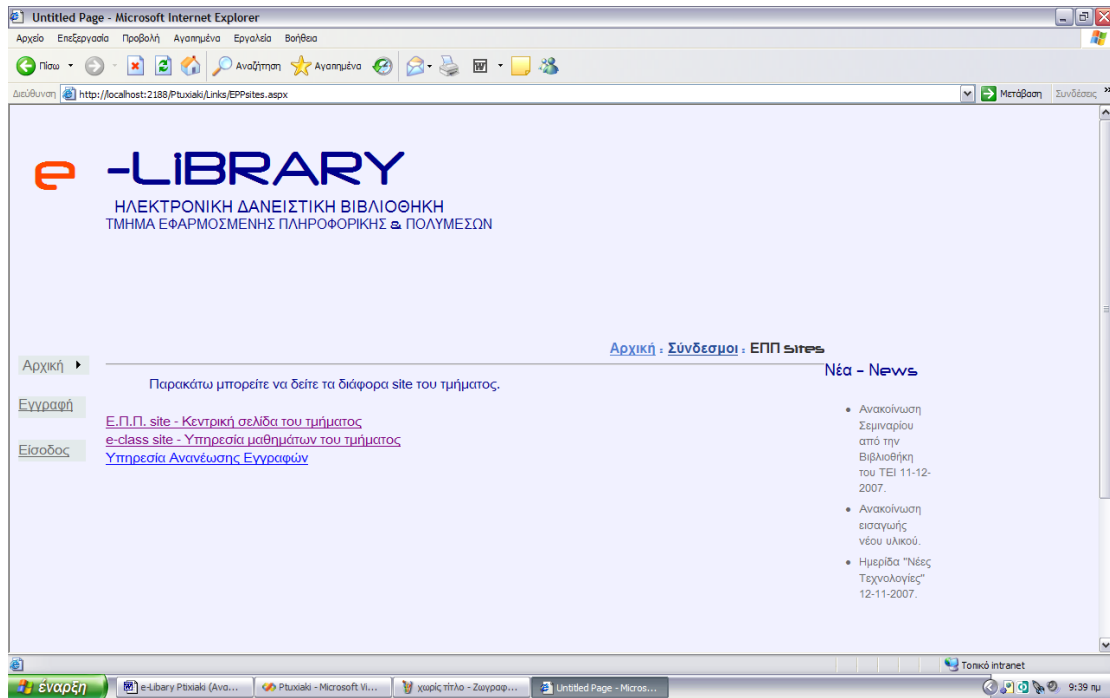
4.3.9 Σύνδεσμοι

Στην σελίδα αυτή εμφανίζεται μία συλλογή από site στα οποία μπορεί να μεταβεί ο χρήστης. Υπάρχουν οι σύνδεσμοι για μεταφορά σε υπηρεσίες του τμήματος ΕΠΠ αλλά και σε άλλα χρήσιμα site.

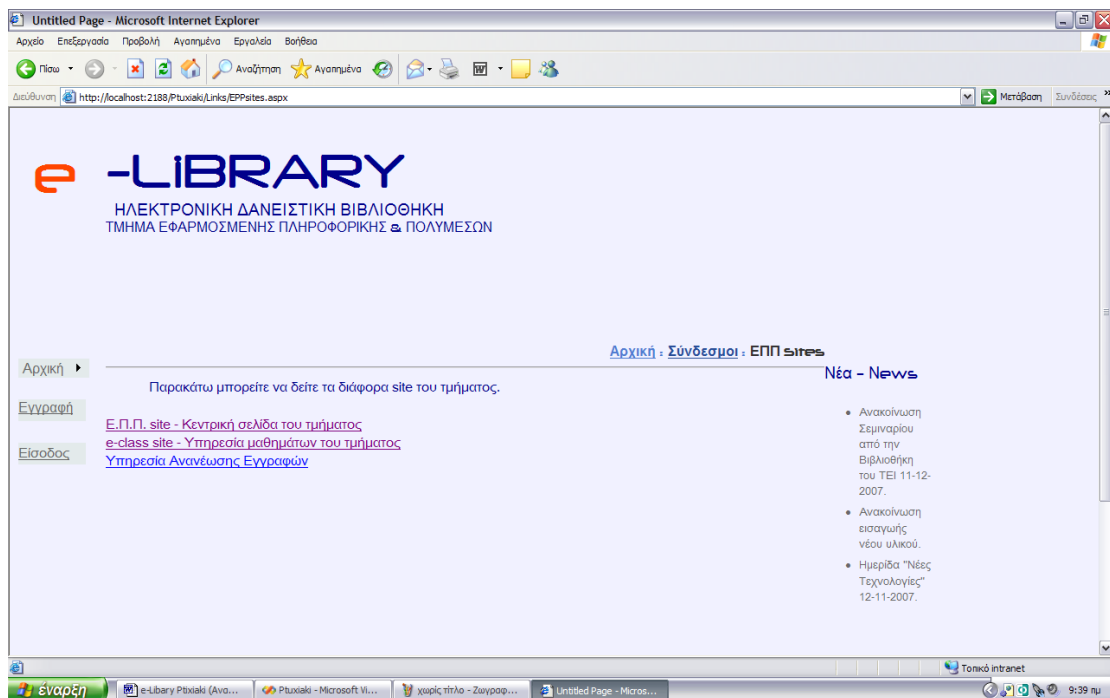


σχήμα 4.27

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
e-LiBRARY
ΗΛΕΚΤΡΟΝΙΚΗ ΔΑΝΕΙΣΤΙΚΗ ΒΙΒΛΙΟΘΗΚΗ



Τα links για το ΕΠΠ



Τα links για τα υπόλοιπα site.

Κεφάλαιο 5^ο

5.1 Μελλοντικές Προσθήκες – Μελλοντικές Χρήσεις

5.1.1 Μελλοντικές Προσθήκες

Η εφαρμογή στο μέλλον μπορεί να δεχθεί διάφορες προσθήκες, τόσο στο υλικό της βιβλιοθήκης που θα μπορεί να περιέχει εκτός από βιβλία και επιπλέον ψηφιακό υλικό όπως DVD-Rom CD-Rom κτλ, όσο όμως και στην ανάπτυξη της Βάσης Δεδομένων και την εμφάνιση των στοιχείων της.

Έτσι σε μία μελλοντική προσθήκη θα μπορούσαμε να πούμε ότι θα υπάρχει και οπτικό υλικό στην εμφάνιση των στοιχείων ενός βιβλίου, με την δυνατότητα εισαγωγής εικόνας μέσω scanner, του βιβλίου που εισάγουμε στην εφαρμογή, καθώς και πολλές περισσότερες πληροφορίες. Για παράδειγμα κατά την εμφάνιση ενός τίτλου θα εμφανίζονται ακόμα περισσότερες πληροφορίες για τον συγγραφέα όπως άλλα βιβλία που έχει γράψει αυτός ή επίσης βιβλία με παρόμοιο περιεχόμενο όπως επίσης και κάποιες σελίδες από το βιβλίο. Η παρακάτω εικόνα εμφανίζει μία τέτοια εμφάνιση βιβλίου:

The screenshot shows a web browser window with the title 'Village-Level Brickmaking: Appendices: Appendix 5 - Woodlots'. The address bar shows a URL from 'www.sadl.uleth.ca'. The page content includes a navigation menu with 'HOME', 'HELP', and 'PREFERENCES'. Below this is a search bar and a list of subjects: 'Trees', 'Fuel', 'Information', 'Firewood', 'Woodlot', 'Used', and 'Government'. The main content area features a book cover for 'Village Level Brickmaking' by Anne Beamish/Will Donovan. The cover image shows a person operating a brick-making machine. Below the cover are buttons for 'EXPAND TEXT', 'EXPAND CONTENTS', 'NO HIGH-LIGHTING', and 'DETACH'. To the right of the cover is a table of contents for 'Village-Level Brickmaking' with 'Appendix 5 - Woodlots' highlighted. The text below the table of contents discusses the use of firewood for brickmaking and its environmental impact.

Στο μέλλον η εφαρμογή θα μπορούσε να είναι προσβάσιμη και μέσω internet, σε μία λειτουργία της βιβλιοθήκης από απόσταση. Έτσι κάποιος χρήστης θα μπορούσε να μπει από το σπίτι του και να κατεβάσει κάποιο βιβλίο σε ηλεκτρονική μορφή. Η βάση δεδομένων θα μπορούσε να περιέχει και βιβλία σε ηλεκτρονική μορφή ώστε οποιοσδήποτε εξουσιοδοτημένος χρήστης να έχει την δυνατότητα να «δανειστεί» κάποιο βιβλίο. Επίσης θα μπορούσε να περιέχει υπηρεσίες e-learning με video ή διάφορα site.

Μία άλλη υπηρεσία που θα μπορούσε να προστεθεί να είναι η ενημέρωση των χρηστών μέσω e-mail για την κατάσταση του λογαριασμού τους και την έγκυρη επιστροφή του βιβλίου. Επίσης η αποστολή e-mail newsletter το οποίο θα ενημερώνει τους χρήστες για νέους τίτλους ή επιπλέον αντίτυπα που θα προστεθούν στην βιβλιοθήκη.

5.1.2 Μελλοντικές Χρήσεις

Στο η ανάπτυξη των ψηφιακών Βιβλιοθηκών θα είναι ραγδαία. Με την εξέλιξη της εικονικής πραγματικότητας και της τεχνικής νοημοσύνης καθώς και των ταχυτήτων στην διακίνηση των πληροφοριών μέσω διαδικτύου, οι υπηρεσίες θα εξελίσσονται όλο και περισσότερο και η χρήστες θα απολαμβάνουν όλο και περισσότερο τις διάφορες υπηρεσίες. Οι προσβάσεις σε βιβλιοθήκες θα είναι πιο εύκολες καθώς και οι υπηρεσίες δανεισμού θα είναι πιο λειτουργικές. Παρακάτω φαίνεται μία εικόνα μιας εικονικής βιβλιοθήκης όπου ο χρήστης μπαίνει στην βιβλιοθήκη κανονικά σαν να βρίσκεται εκεί και κάνει την αναζήτησή του στους διαδρόμους της ψάχνοντας μόνος του τον τίτλο του βιβλίου. Μπορούμε να φανταστούμε μία τέτοια βιβλιοθήκη σαν ένα παιχνίδι στο οποίο το ρόλο του κεντρικού ήρωα τον έχει ο χρήστης και μπορεί να κάνει ότι αυτός επιθυμεί μέσα στην βιβλιοθήκη. Όταν ο χρήστης βρει το βιβλίο που επιθυμεί, το επιλέγει και μπορεί να το έχει στον υπολογιστή του σε πραγματικό χρόνο.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
e-LIBRARY
ΗΛΕΚΤΡΟΝΙΚΗ ΔΑΝΕΙΣΤΙΚΗ ΒΙΒΛΙΟΘΗΚΗ



Αναζήτηση βιβλίου μέσα σε μια εικονική Βιβλιοθήκη.



Επιλογή Βιβλίου.

Κεφάλαιο 6^ο

6.1 Ανακεφαλαίωση

Με τη εξέλιξη της τεχνολογίας με την πάροδο των χρόνων οι διάφορες υπηρεσίες θα εξελίσσονται και αυτές όλο και περισσότερο. Κατά την διάρκεια των προηγούμενων λίγων χρόνων η ποιότητα του σχεδιασμού των user interfaces των ψηφιακών βιβλιοθηκών έχει αυξηθεί σημαντικά. Τα πρότυπα για το γραφικό σχεδιασμό γίνονται κάθε χρόνο και καλύτερα. Μια καλή υποστήριξη για τους χρήστες είναι κάτι παραπάνω από μια διακοσμητική διαδικασία. Κομψή σχεδίαση, κατάλληλη λειτουργικότητα και συστήματα που ανταποκρίνονται, δημιουργούν μια πολύ μεγάλη διαφορά στη δυναμικότητα των ψηφιακών βιβλιοθηκών. Οι ψηφιακές βιβλιοθήκες πρέπει να παρέχουν αναζήτηση και συστήματα ανάκτησης ώστε οι χρήστες να μπορούν να έχουν εύκολη πλοήγηση στις πληροφορίες. Οι χρήστες πρέπει να νιώθουν άνετα και να λαμβάνουν συνεχή πληροφόρηση από το σύστημα με σκοπό την ανεύρεση της ζητούμενης πληροφορίας. Μια ψηφιακή βιβλιοθήκη είναι τόσο καλή όσο καλό είναι το interface που παρέχει στους χρήστες της.

Κεφάλαιο 7^ο

7.1 Βιβλιογραφία

- ΨΗΦΙΑΚΕΣ ΒΙΒΛΙΟΘΗΚΕΣ ΚΑΙ USER INTERFACES ΜΑΘΗΜΑ: «ΨΗΦΙΑΚΕΣ ΒΙΒΛΙΟΘΗΚΕΣ» ΚΑΘΗΓΗΤΗΣ: Σ. ΚΑΠΙΔΑΚΗΣ.
- ΟΙ ΒΙΒΛΙΟΘΗΚΕΣ ΣΤΗΝ ΕΛΛΑΔΑ ΑΠΟ ΤΗΝ ΑΡΧΑΙΟΤΗΤΑ ΜΕΧΡΙ ΣΗΜΕΡΑ: ΣΥΝΤΟΜΗ ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ Ημερίδα για τα 50 χρόνια από την ίδρυση του Ιδρύματος Ευγενίδου 19 Μαΐου 2006.
- Σύστημα Βιβλιοθηκών Α.Π.Θ.
- Microsoft MSDN Library for Visual Studio, C#, .NET Framework,
- ASP.NET Website Programming, C# Edition Problem, Design, Solution 2002
- ASP.NET Database Programming Weekend Crash Course (2002)
- Visual Web Developer 2005 Express Edition Beta
- www.msdn.com
- www.microsoft.com
- www.csharpcorner.com
- Proficiency Series for Visual Studio.NET 2003
- VB.NET for Beginners to Programming Workbook and Videos
- C# Beginner to Programming Workbook and Videos
- Τμήμα Βιβλιοθηκονομίας Ιόνιο Πανεπιστήμιο.
- .NET Framework Class Library
- Beginning C#

Περιεχόμενα

Κεφάλαιο1.....	σελ 2
1.Εισαγωγή.....	σελ2
κεφάλαιο2.....	σελ4
2.Τεχνολογίες Υλοποίησης.....	σελ4
2.1 Microsoft Visual Studio.....	σελ4
2.2 Γενικά για την .NET.....	σελ5
2.2.1 Τι είναι Web Υπηρεσίες.....	σελ5
2.2.2 Οφέλη των Web Υπηρεσιών.....	σελ6
2.2.3 Συνδέοντας εφαρμογές με web Υπηρεσίες.....	σελ6
2.3 .NET Framework.....	σελ7
2.3 Γλώσσα Προγραμματισμού C#.....	σελ8
2.3.1 Γενικά για την C#.....	σελ8
2.3.2 C# Γλώσσα.....	σελ8
2.3.3 Object – oriented προγραμματισμός.....	σελ9
2.3.4 ASP.NET.....	σελ15
2.3.5 ADO.NET.....	σελ16
2.3.6 XML.....	σελ20
2.3.7 Βάσεις Δεδομένων και SQL Server 2000.....	σελ22
Κεφάλαιο 3. Βιβλιοθήκες.....	σελ24
3.1 Ιστορική Αναδρομή στις Βιβλιοθήκες.....	σελ24
3.1.2 Στατιστικά στοιχεία βιβλιοθηκών.....	σελ27
3.1.3 Ανάπτυξη και Ενίσχυση	σελ27
3.2. Ψηφιακές Βιβλιοθήκες.....	σελ28
3.3 Βασικές Αρχές Βιβλιοθηκονομίας.....	σελ31
3.3.1 Υποχρεώσεις απέναντι στο χρήστη.....	σελ31
3.3.2 Υποχρεώσεις προς το επάγγελμα.....	σελ32
3.3.3 Υποχρεώσεις προς το υλικό και την πληροφορία.....	σελ32
Κεφάλαιο 4 ^ο	σελ34
4. Δανειστική Ηλεκτρονική Βιβλιοθήκη e-LiBRARY.....	σελ34
4.1.1 Περιγραφή της Βάσης Δεδομένων.....	σελ34

4.1.2 Οι Πίνακες της Βάσης.....σελ35	σελ35
4.1.3 Διάγραμμα βάσης δεδομένων – Συσχετίσεις πινάκων.....σελ40	σελ40
4.1.4 Βάση Δεδομένων «ΑΝΑΚΟΙΝΩΣΕΙΣ».....σελ42	σελ42
4.2 Διεπαφή Χρήστη - User Interface.....σελ43	σελ43
4.2.1 Σχεδίαση της οθόνης (screen design).....σελ44	σελ44
4.3.10 Πλοήγηση (navigation).....σελ45	σελ45
4.3.11 Τρόποι Παρουσίασης Ψηφιακών Δεδομένων Ιεραρχικά δομημένα κείμενα.....σελ44	σελ44
4.3.12 Διαδικασίες Πρόσβασης και Αναζήτησης.....σελ44	σελ44
4.4 Site Map.....σελ48	σελ48
4.3.1 Master Page.....σελ48	σελ48
4.3.2 Αρχική.....σελ50	σελ50
4.3.3 Εγγραφή.....σελ52	σελ52
4.3.4 Είσοδος.....σελ53	σελ53
4.3.5 Βιβλία.....σελ54	σελ54
4.3.6 Υπηρεσίες.....σελ56	σελ56
4.3.7 Σελίδες Μελών.....σελ57	σελ57
4.3.8 Διαχείριση.....σελ58	σελ58
4.3.9 Σύνδεσμοι.....σελ60	σελ60
Κεφάλαιο 5.....σελ61	σελ61
5.1 Μελλοντικές Προσθήκες – Μελλοντικές Χρήσεις.....σελ61	σελ61
5.1.2 Μελλοντικές Χρήσεις.....σελ62	σελ62
6 .Ανακεφαλαίωση.....σελ64	σελ64
7.Βιβλιογραφία.....σελ65	σελ65
Περιεχόμενα.....σελ66	σελ66

