



Υλοποίηση διαδικτυακού τόπου για τη δημοσίευση επιστημονικών εγγράφων



Γκουτζουρέλας Ιωάννης

Και

Σωτηρίου

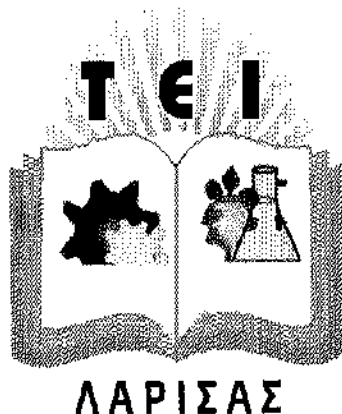
Δημήτρης

Τ.Ε.Ι. Λάρισας

Τμήμα τεχνολογίας

πληροφορικής

τηλεπικοινωνιών



ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΛΑΡΙΣΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

‘ΥΛΟΠΟΙΗΣΗ ΔΙΑΔΙΚΤΥΑΚΟΥ ΤΟΠΟΥ ΓΙΑ ΤΗ
ΔΗΜΟΣΙΕΥΣΗ ΕΠΙΣΤΗΜΟΝΙΚΩΝ ΕΓΓΡΑΦΩΝ’

ΓΚΟΥΤΖΟΥΡΕΛΑΣ ΙΩΑΝΝΗΣ
ΣΩΤΗΡΙΟΥ ΔΗΜΗΤΡΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΣΩΜΑΡΑΣ ΧΡΗΣΤΟΣ

ΛΑΡΙΣΑ 2010

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Λάρισα,

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

- 1.
- 2.
- 3.

Πίνακας περιεχομένων

Πίνακας περιεχομένων.....	4
Εισαγωγή.....	4
ΚΕΦΑΛΑΙΟ 1. Υπάρχουσες τεχνολογίες.....	6
1.1 PHP.....	6
1.2 ASP και .NET Framework.....	8
1.3 JSP	11
1.4 AJAX Technologies.....	13
ΚΕΦΑΛΑΙΟ 2. Υλοποιήσεις e-learning.....	15
2.1 Sakai	16
2.2 eConf.....	18
2.3 OLAT.....	19
.....	21
2.4 Bibliotheca.....	23
2.5 Συγκρίσεις και συμπεράσματα.....	27
ΚΕΦΑΛΑΙΟ 3. Σχεδίαση βάσης δεδομένων.....	28
ΚΕΦΑΛΑΙΟ 4. Περιβάλλον ανάπτυξης.....	31
ΚΕΦΑΛΑΙΟ 5. Παρουσίαση εφαρμογής.....	40
6.1 Αναλυτική επεξήγηση των αρχείων JSP.....	64
6.2 Αναλυτική επεξήγηση κώδικα Java.....	76
6.3 Αναλυτική επεξήγηση Struts configuration.....	85
.....	93
Βιβλιογραφία - Πηγές.....	94

Εισαγωγή

Η πτυχιακή αυτή εργασία αφορά τη σχεδίαση και ανάπτυξη μιας διαδικτυακής εφαρμογής για τη δημοσίευση επιστημονικών εγγράφων. Τα έγγραφα αυτά θα είναι της μορφής Word ή PDF. Ο χρήστης θα μπορεί μέσα σε μια δυναμική ιστοσελίδα να ανεβάζει εργασίες, σημειώσεις, επιστημονικά άρθρα και γενικότερα έγγραφα που αφορούν την εκπαίδευση και την έρευνα.

Οι χρήστες θα μπορούν να επισκέπτονται τη σελίδα αυτή και να αναζητούν άρθρα που τους ενδιαφέρουν ή να πλοηγούνται μέσα από κατηγορίες και υποκατηγορίες. Τα άρθρα θα βρίσκονται μέσα σε καταλόγους στο λειτουργικό σύστημα, αλλά οι πληροφορίες οι σχετικές τόσο με τα άρθρα, όσο και τις κατηγορίες και υποκατηγορίες, όσο και οι πληροφορίες σχετικά με τους χρήστες που έχουν πρόσβαση, θα βρίσκονται σε μια SQL βάση δεδομένων.

Για κάθε επιστημονικό έγγραφο ή σημείωση που ο χρήστης (ή ο διαχειριστής του συστήματος) θα ανεβάζει στο site επιπλέον πληροφορίες θα ζητούνται. Για παράδειγμα θα πρέπει να δίνονται ορισμένες λέξεις-κλειδιά οι οποίες θα χρησιμοποιούνται για την ανεύρεση του εγγράφου αυτού σε αναζητήσεις. Επίσης το έγγραφο θα πρέπει να τοποθετείται σε κατάλληλη κατηγορία-υποκατηγορία. Ο διαχειριστής του συστήματος θα έχεις δυνατότητες σχετικά με τη διαχείριση χρηστών και ομάδων, διαχείριση εγγράφων και κατηγοριών - υποκατηγοριών.

Οι τεχνολογίες που θα χρησιμοποιηθούν για την ανάπτυξη της διαδικτυακής αυτής εφαρμογής είναι η SQL, η Java και η JSP. Λεπτομέρειες σχετικά με την τελική επιλογή των τεχνολογιών αλλά και για την υλοποίηση θα παρατεθούν στη συνέχεια της πτυχιακής αυτής εργασίας.

ΚΕΦΑΛΑΙΟ 1. Υπάρχουσες τεχνολογίες

Οι διαθέσιμες τεχνολογίες οι οποίες θα μπορούσαν να χρησιμοποιηθούν όσον αφορά την ανάπτυξη της δυναμικής διαδικτυακής εφαρμογής είναι: PHP [1], ASP [2] και JSP [3]. Όλες οι παραπάνω τεχνολογίες έχουν τη δυνατότητα να δημιουργούν δυναμικές σελίδες με πλούσια αλληλεπίδραση του χρήστη και να στηριχτούν σε πληθώρα σχεσιακών βάσεων δεδομένων.

Πέρα όμως από τις γνωστές αυτές τεχνολογίες τελευταία όλο και περισσότερες AJAX εφαρμογές εμφανίζονται στο διαδίκτυο προσφέροντας καινοτόμες τεχνολογίες, βελτιωμένα γραφικά περιβάλλοντα, βελτίωση στον τομέα της ταχύτητας κ.α. Οι τεχνολογίες αυτές θα παρουσιαστούν μετά τις προαναφερθείσες και στο τέλος του κεφαλαίου θα γίνει η τελική επιλογή της γλώσσας υλοποίησης.

1.1 PHP

Η PHP αποτελεί μια γλώσσα [προγραμματισμού](#) για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο. Ο κώδικας της γλώσσας περνά από επεξεργασία από ένα συμβατό [διακομιστή](#), για παράδειγμα τον Apache [4], ώστε να παραχθεί σε πραγματικό χρόνο το περιεχόμενο που θα σταλεί στον φυλλομετρητή του επισκέπτη σε μορφή κώδικα [HTML](#).

Τα αρχεία PHP έχουν την κατάλληλη κατάληξη .php, .php4, .phtml. Για να λειτουργήσει η ενσωμάτωση κώδικα σε ένα αρχείο HTML πρέπει να γίνει η κατάλληλη ρύθμιση στα MIME types [5] του διακομιστή ιστοσελίδων. Διαφορετικά θα εμφανίσει στον φυλλομετρητή τον κώδικα χωρίς καμία επεξεργασία. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο διακομιστής να είναι ρυθμισμένος για να επεξεργάζεται κώδικα PHP. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα [λειτουργικά συστήματα Linux](#) και [Microsoft Windows](#), υποστηρίζει εξ ορισμού τον κώδικα PHP. Παρόμοια υποστήριξη υπάρχει και από άλλους εμπορικούς ή μη διακομιστές

Η PHP ως γλώσσα προγραμματισμού ξεκίνησε το [1995](#), όταν ένας φοιτητής, ο [Rasmus Lerdorf](#) δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα [script](#) [6] με

όνομα php.cgi. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. [7]

Η εξέλιξη του script αυτού αποτέλεσε τη γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το [1997](#) η PHP/FI στην έκδοση 2.0 (βασισμένη αυτή τη φορά στη γλώσσα C) υπήρχε σε περισσότερους από 50.000 [ιστότοπους](#). Η νέα έκδοση της 3.0 είχε την ονομασία PHP. Μια νέα εταιρία από τους Zeev και Andi (με την ονομασία Zend από τα αρχικά των ονομάτων τους) συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP.

Το 1998 ακολούθησε η έκδοση 4 της PHP και τον Ιούλιο του [2004](#) διατέθηκε η έκδοση 5. Τη στιγμή αυτή οι περισσότεροι ιστότοποι χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP ενώ η έκδοση 6 βρίσκεται σε εξελικτικό στάδιο.

Τα οφέλη που έδωσε η γλώσσα αυτή στους σχεδιαστές ιστοτόπων συγκριτικά με τη γλώσσα HTML ήταν η δυνατότητα ορισμού μεταβλητών, η χρήση αυτών των μεταβλητών, η δυνατότητα συγγραφής μεθόδων, η εύκολη διασύνδεση με βάσεις δεδομένων και άλλα πολλά. Τη στιγμή αυτή εκατομμύρια sites την χρησιμοποιούν για την παροχή υπηρεσιών και δυναμικών ιστοσελίδων και αποτελεί μια γλώσσα που θα μπορούσε με αξιώσεις να χρησιμοποιηθεί για την ανάπτυξη της δικής μας δυναμικής διαδικτυακής εφαρμογής.

1.2 ASP και .NET Framework

Το .NET Framework είναι μία βιβλιοθήκη της Microsoft για λειτουργικά συστήματα Windows. Περιέχει εξελιγμένες λειτουργίες και οι εφαρμογές που δημιουργούνται με αυτό είναι συμβατές με όλα τα συστήματα που υποστηρίζουν και έχουν εγκατεστημένο το .NET Framework.

Η αρχιτεκτονική του .NET Framework χωρίζεται σε 4 τμήματα:

- α) Το Common Language Runtime (CLR)
- β) Επιμέρους βιβλιοθήκες
- γ) Υποστήριξη για αρκετές γλώσσες προγραμματισμού
- δ) Την ASP.NET.

Το .NET Framework έχει ως στόχο να ικανοποιήσει τρεις στόχους. Πρώτον να κάνει τις εφαρμογές των Windows αξιόπιστες και να βελτιώσει την ασφάλειά τους. Δεύτερον να απλουστεύσει της ανάπτυξης Web εφαρμογών και υπηρεσιών (Web services), οι οποίες μάλιστα να μπορούν να εκτελούνται σε φορητές συσκευές. Τέλος να παρέχει ένα σύνολο βιβλιοθηκών που να μπορούν να λειτουργούν με πολλές γλώσσες προγραμματισμού.

Common Language Runtime

Σχεδόν κάθε γλώσσα προγραμματισμού συνοδεύεται από έναν compiler και ένα περιβάλλον runtime, με εξαίρεση τις interpreted γλώσσες προγραμματισμού όπως η java και η PHP. Η δουλειά του compiler είναι να μεταφράζει τον κώδικα σε εκτελέσιμα αρχεία. Από την άλλη το runtime παρέχει υπηρεσίες του λειτουργικού συστήματος στο εκτελέσιμο αρχείο. Οι υπηρεσίες αυτές είναι ενσωματωμένες σε ένα επίπεδο runtime. Τέτοιες λειτουργίες μπορεί να είναι η διαχείριση μνήμης, η εγγραφή και ανάγνωση αρχείων, η επικοινωνία μέσω ιντερνέτ κλπ.

Προτού εμφανιστεί το .NET Framework, κάθε γλώσσα προγραμματισμού είχε το δικό της runtime. Η Visual Basic για παράδειγμα, είχε το MSVBVM60.DLL, ενώ η Visual C++ το MSVCRT.DLL. Ο εκτελέσιμος κώδικας της εφαρμογής και το runtime ενσωματωνόταν και

τοποθετούταν στο σύστημα του χρήστη κατά την εγκατάσταση. Το βασικό μειονέκτημα ήταν ότι κάθε περιβάλλον ήταν σχεδιασμένο για μία γλώσσα προγραμματισμού. Οι λειτουργίες από το περιβάλλον μιας γλώσσας δεν μπορούσαν να χρησιμοποιηθούν σε μία άλλη. Η ενοποίηση λοιπόν των runtimes ήταν ένας από τους βασικούς στόχους του .NET Framework. Η λύση στο πρόβλημα αυτό ήρθε από το Common Language Runtime (CLR). Το runtime αυτό παρέχει όλες τις δυνατότητες των παλαιότερων runtimes, αλλά λειτουργεί σε κάθε γλώσσα προγραμματισμού που ανήκει στην οικογένεια του .NET Framework. Ένα επιπλέον χαρακτηριστικό του CLR είναι ότι επιτρέπει στις γλώσσες να συνεργάζονται μεταξύ τους.

.NET Class Libraries

Η επαναχρησιμοποίηση κώδικα είναι μια δοκιμασμένη τεχνική που χρησιμοποιείται από προγραμματιστές. Πολλές βιβλιοθήκες είχαν αναπτυχθεί με το Win32 API ή το MFC και οι προγραμματιστές μπορούσαν εύκολα και γρήγορα να αναπτύσσουν εφαρμογές. Η κάθε βιβλιοθήκη όμως αντιστοιχούσε σε μια γλώσσα προγραμματισμού. Με την έλευση του .NET Framework πολλές κλάσεις και βιβλιοθήκες δόθηκαν στους προγραμματιστές για να επαναχρησιμοποιηθούν. Οι βιβλιοθήκες αυτές με την ονομασία .NET Class Libraries περιέχουν κώδικα για διαχείριση νημάτων (threads), εγγραφή/ανάγνωση αρχείων (File I/O), υποστήριξη βάσεων δεδομένων, διαχείριση XML, δομές δεδομένων (στοίβες, ουρές κλπ).

.NET Γλώσσες προγραμματισμού

Το .NET Framework παρέχει εργαλεία και βιβλιοθήκες που βοηθάν την κατασκευή κώδικα. Οι συμβατές γλώσσες με την πλατφόρμα αυτή είναι η C#, η VB.NET, η Jscript.NET κ.α. Ακόμα και γλώσσες που δε δημιουργήθηκαν από την Microsoft όπως η COBOL και η Perl υποστηρίζονται από το .NET Framework.

ASP.NET

Η τεχνολογία Active Server Pages (ASP) αναπτύχθηκε από Microsoft για τη δημιουργία δυναμικών ιστοσελίδων. Η αρχική έκδοση της ASP χρησιμοποιούσε VB Script στον

εξυπηρετητή και με την εκτέλεση του Script δημιουργούσε τη σελίδα ανάλογα με τα στοιχεία που εισήγαγε ο χρήστης.

Η νέα ASP.NET βελτίωσε κατά πολύ την αρχική υλοποίηση. Ενώ αρχικά ο κώδικας HTML και το script βρίσκονταν στο ίδιο αρχείο, στην ASP.NET βρίσκονται σε διαφορετικά. Επίσης προστέθηκε υποστήριξη Web Forms, που επιτρέπει στον προγραμματιστή με drag & drop να δημιουργήσει φόρμες. Στην ASP ο κώδικας μεταγλωττίζεται στον εξυπηρετητή και μετά φορτώνεται σαν ένα html αρχείο χωρίς ο χρήστης να μπορεί να δει τον αρχικό κώδικα.

Πώς δουλεύουν οι δυναμικές σελίδες

Όταν ένας φυλλομετρητής ζητά ένα στατικό αρχείο (π.χ. HTML), τότε ο εξυπηρετητής το αποστέλλει στον πελάτη. Όταν όμως ζητά ένα δυναμικό αρχείο όπως PHP ή ASP ή JSP, ο εξυπηρετητής καλεί τον κώδικα του αρχείου αυτού, εκτελεί τα scripts που εμπεριέχει και αποστέλλει στον φυλλομετρητή του πελάτη το αποτέλεσμα της εκτέλεσης.

1.3 JSP

Η JSP είναι η αντίστοιχη τεχνολογία της Sun Microsystems για δυναμικές ιστοσελίδες. Διαθέτει μηχανισμούς για επεξεργασία και ανάκτηση δεδομένων τόσο από βάσεις δεδομένων όσο και από αρχεία XML. Η σύνταξη της JSP είναι πανομοιότυπη με την Java κάνοντας την ευκολοκατανόητη για τους προγραμματιστές που διαθέτουν ήδη εμπειρία με τη Java.

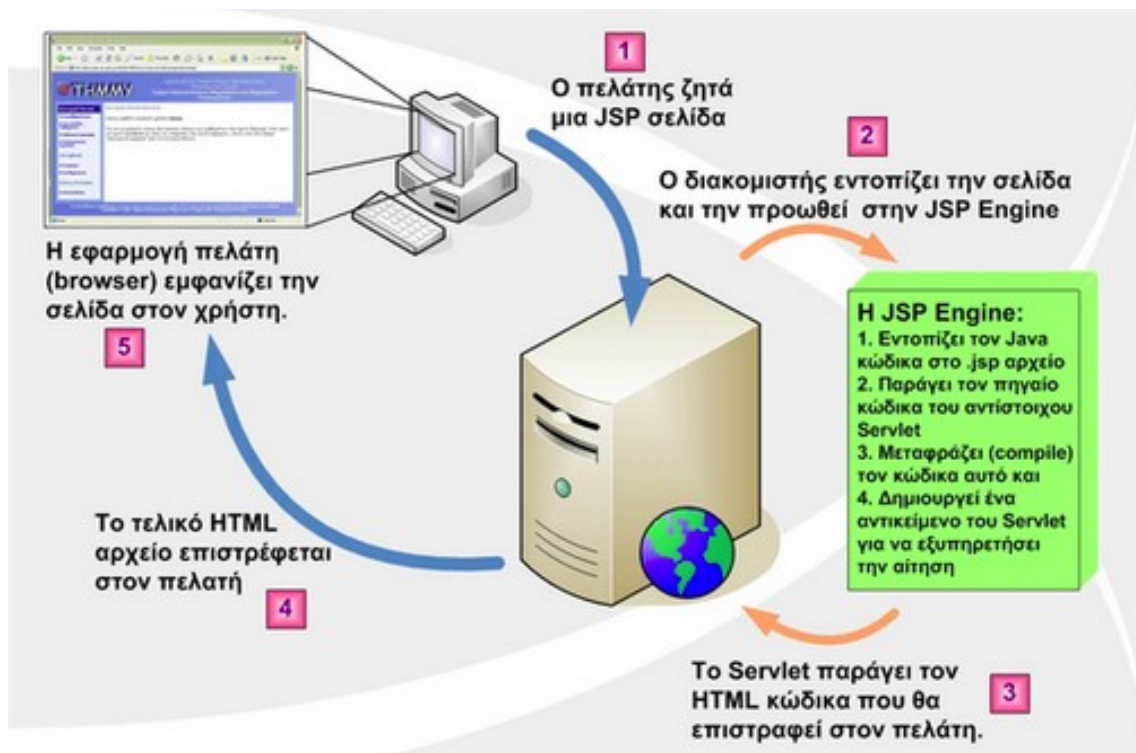
Η JSP από μόνη της δεν μπορεί να κατασκευάσει ιστοσελίδες. Όπως η PHP και η ASP, έτσι και αυτή αποτελεί μία συμπληρωματική τεχνολογία για την κατασκευή διαδικτυακών εφαρμογών. Η παραγόμενη σελίδα που φτάνει στο φυλλομετρητή του χρήστη περιέχει κώδικα HTML, JavaScript και CSS. Η JSP όμως επιτρέπει τη μίξη κώδικα HTML και κώδικα Java στην ίδια σελίδα.

Σε μια άλλη τεχνολογία της Sun, στα Java Servlets ο χρήστης θα πρέπει να γράψει αρκετές εντολές `out.print` για να δημιουργήσει τη δυναμική ιστοσελίδα. Αυτό είναι μια αρκετά σύνθετη και κουραστική διαδικασία, ειδικά αν η σελίδα περιέχει πίνακες ή άλλα σύνθετα HTML στοιχεία. Η προσέγγιση της PHP είναι «Η δυναμική σελίδα να σχηματίζεται απευθείας με τη χρήση HTML και **μόνο όπου χρειάζεται** να χρησιμοποιείται ο κώδικας της PHP». Αυτή ακριβώς την προσέγγιση έχει ο κώδικας JSP.

Οι JSP σελίδες δημιουργούνται λοιπόν από τον προγραμματιστή και τοποθετούνται στο φάκελο από όπου ο εξυπηρετητής σερβίρει τις ιστοσελίδες. Όταν ο χρήστης ζητά μια στατική σελίδα, τότε ο εξυπηρετητής του την προσφέρει όπως ακριβώς την έχει αποθηκευμένη. Όταν όμως του ζητά μια JSP σελίδα ο εξυπηρετητής διαδικτύου που χρησιμοποιείται (π.χ. ο Tomcat) την μεταφράζει παρασκηνακά με χρήση κάποιου servlet και την εκτελεί. Η μετάφραση σε servlet γίνεται μόνο την πρώτη φορά που ζητάμε την σελίδα και έτσι η επίδοση της εφαρμογής δεν επιβαρύνεται συγκριτικά με τα servlets. Αφού δηλαδή ετοιμαστεί ο κώδικας της JSP ένας compiler τον μετατρέπει σε Java Servlets. Η διαδικασία αυτή περιγράφεται στο σχήμα 2-1.

Η τεχνολογία JSP έχει πολλά πλεονεκτήματα, τα περισσότερα από τα οποία «κληρονομεί» από την γλώσσα στην οποία βασίζεται (Java). Γενικά:

- παρέχει ένα εύρωστο περιβάλλον ανάπτυξης δικτυακών εφαρμογών
- είναι ανεξάρτητη από την πλατφόρμα / λειτουργικό σύστημα
- είναι ανεξάρτητη από τον διακομιστή (web server)
- προσφέρει τη δυνατότητα επαναχρησιμοποίησης του κώδικα, αφού κληρονομεί από την Java τις τεχνολογίες JavaBeans και EJB.



Σχήμα 1-1: Εξυπηρέτηση αίτησης για JSP σελίδα

1.4 AJAX Technologies

Το AJAX σημαίνει Asynchronous JavaScript and XML. Χρησιμοποιεί δηλαδή την JavaScript, μια γλώσσα που αρχικά σχεδιάστηκε για να προσφέρει μικρές δυνατότητες scripting στην πλευρά του πελάτη αλλά εξελίχθηκε σε μια τεχνολογία που υποστηρίζεται πλήρως από τους περισσότερους φυλλομετρητές και προσφέρει πλέον πολλές δυνατότητες.

Η μεταφορά των δεδομένων από τον πελάτη στον εξυπηρετητή γίνεται με XML μηνύματα. Τέλος η λέξη Asynchronous σημαίνει ότι ο φυλλομετρητής του χρήστη και ο εξυπηρετητής δεν δουλεύουν συγχρονισμένα. Οι κλασσικές εφαρμογές είναι synchronous. Όταν ο χρήστης χρησιμοποιεί κάποιο component στο γραφικό περιβάλλον μιας κλασσικής εφαρμογής, αυτομάτως η εντολή που δίνει πραγματοποιείται. Στις ασύγχρονες εφαρμογές, ο χρήστης χρησιμοποιεί κάποιο component αλλά υπάρχει ένα χρονικό περιθώριο μέχρι να εκπληρωθεί η επικοινωνία μέσω του διαδικτύου και πραγματικά να υλοποιηθεί η επιθυμία του χρήστη.

Βέβαια οι AJAX εφαρμογές έχουν μηχανισμούς για να μειώνουν αυτό το «χρόνο αντίδρασης» που παρατηρείται σε όλες τις διαδικτυακές εφαρμογές. Το pre-loading είναι ένας από αυτούς τους μηχανισμούς. Αφού δηλαδή ο χρήστης συνδεθεί στη σελίδα που προσφέρει μια AJAX εφαρμογή και «κατέβει» ο κώδικάς της στον τοπικό υπολογιστή, ομοίως κατεβαίνουν και δεδομένα. Κατεβάζοντας επιπλέον δεδομένα (όπως εικόνες που θα πρέπει να εμφανιστούν στο χρήστη όταν αυτός κάνει μια επιλογή, όπως να αλλάξει καρτέλα) η εφαρμογή φαίνεται και λειτουργεί πιο ζωντανά από αντίστοιχες εφαρμογές γραμμένες σε άλλες τεχνολογίες όπως PHP, JSP και ASP.

Οι εφαρμογές AJAX έχουν εμφανιστεί τα τελευταία χρόνια σε πολλές δημοφιλείς ιστοσελίδες, χωρίς ο περισσότερος κόσμος να γνωρίζει την ύπαρξη τους, μιας και φαίνονται σαν τις γνωστές δυναμικές εφαρμογές του παρελθόντος, αλλά με αξιοσημείωτες βελτιώσεις.

Τέτοιες εφαρμογές αποτελούν το Gmail (Η σελίδα που χρησιμοποιούν οι χρήστες για να αποκτήσουν πρόσβαση στα email τους στην εταιρία Google), τα Google Maps και από το 2008 το MSN Live (Η υπηρεσία της Microsoft που χρησιμοποιούν οι χρήστες για τα emails τους τύπου Hotmail και MSN) καθώς και το Yahoo email και τα Yahoo maps. Πολλές

δηλαδή από τις σύγχρονες δημοφιλείς υπηρεσίες είναι πλέον γραμμένες σαν AJAX εφαρμογές και εκατομμύρια χρήστες απολαμβάνουν τα οφέλη της τεχνολογίας αυτής, χωρίς να χρειάζεται από την πλευρά τους κάποια ιδιαίτερη τροποποίηση στο λογισμικό του υπολογιστή τους.

Για τη συγγραφή κώδικα σε μορφή AJAX υπάρχουν οι εξής δυνατότητες:

1. Συγγραφή JavaScript «με το χέρι»
2. Χρήση έτοιμων βιβλιοθηκών που με απλά tags επιτρέπουν τη συγγραφή AJAX εφαρμογών.
3. Συγγραφή κώδικα σε μια γνωστή γλώσσα προγραμματισμού και μετατροπή του σε AJAX μέσω κάποιου Toolkit

Στην πρώτη περίπτωση, η συγγραφή κώδικα JavaScript χωρίς την χρήση κάποιου εργαλείου είναι πραγματικά δύσκολη υπόθεση ακόμα και για δυνατούς προγραμματιστές. Ο λόγος είναι η μη ύπαρξη εργαλείων για debugging.

Η χρήση έτοιμων βιβλιοθηκών είναι ο πιο εύκολος τρόπος ιδικά σε περιπτώσεις που απλές δυνατότητες της AJAX είναι απαραίτητες. Για παράδειγμα η Adobe έχει εμφανίσει ένα library με την ονομασία AIR που επιτρέπει τους προγραμματιστές να δημιουργήσουν αυτό που ονομάζουν «Rich Internet Applications that run outside the browser on multiple operating systems».

ΚΕΦΑΛΑΙΟ 2. Υλοποιήσεις e-learning

Η εφαρμογή που θα αναπτυχθεί στα πλαίσια της πτυχιακής αυτής εργασίας αφορά κατά κάποιο βαθμό συστήματα e-learning. Στο παρελθόν έχουν υλοποιηθεί και υλοποιούνται ακόμα και σήμερα συστήματα υποβοήθησης μάθησης σε εφαρμογές ανοιχτού λογισμικού. Προτού αναλύσουμε τα specifications του συστήματος που θα αναπτύξουμε θα παρουσιάσουμε ορισμένα από τα υπάρχον λογισμικά για να ανακαλύψουμε τις δυνατότητες που παρέχουν στους χρήστες τους και τον τρόπο υλοποίησής τους. Η εμπειρία αυτή κρίνεται απαραίτητη ώστε να αποφασίσουμε τον τρόπο αλληλεπίδρασης του λογισμικού που θα αναπτύξουμε με τον τελικό χρήστη.

Ορισμένες από τις εφαρμογές που θα παρουσιάσουμε δεν θα έχουν άμεση σχέση με το θέμα «Υλοποίηση διαδικτυακού τόπου για τη δημοσίευση επιστημονικών εγγράφων» αλλά οι τεχνολογίες και οι τεχνοτροπίες που έχουν υλοποιηθεί παρουσιάζουν ενδιαφέρον και η κατανόηση των δυνατοτήτων τους ίσως φανεί χρήσιμη στην πορεία.

2.1 Sakai

Το Sakai [11] αναπτύσσεται από τα University of Michigan, το Indiana University, το MIT, το Stanford, το uPortal Consortium και το Open Knowledge Initiative (OKI) και με τη βοήθεια του Andrew W. Mellon Foundation και συνολικές χορηγίες μεγαλύτερες των 7 εκατομμυρίων δολαρίων παρουσίασε για πρώτη φορά τις δυνατότητές του τον Ιούλιο του 2004.

Το Sakai αποτελεί ένα πακέτο με λογισμικά που έχουν σχεδιαστεί για να βοηθάν τους ερευνητές, τους καθηγητές σε πανεπιστήμια και τους μαθητές να δημιουργούν web-sites για συνεργασιακή μάθηση. Χρησιμοποιώντας τον φυλλομετρητή τους, οι χρήστες μπορούν να χρησιμοποιήσουμε μέσα από μια συλλογή εργαλείων αυτά που είναι τα πλέον χρήσιμα για τις δικές τους ανάγκες.

Μια χρήση του Sakai είναι η δημιουργία ενός site όπου θα μπορεί ο χρήστης του να κάνει ανακοινώσεις, αλλά και να παρουσιάζει εργασίες, reports και διάφορα άλλα κείμενα προς το κοινό του. Αυτή η συγκεκριμένη δυνατότητα του Sakai μοιάζει σε μεγάλο βαθμό με αυτό που θέλουμε να πετύχουμε στην πτυχιακή αυτή εργασία.

Οι δυνατότητες όμως είναι πραγματικά πολλές για το λογισμικό αυτό έχει σχεδιαστεί για να υποστηρίζει πολλές διαφορετικές χρήσεις. Ένας καθηγητής μπορεί να το χρησιμοποιήσει για να δημιουργήσει forums όπου οι μαθητές του θα συζητάν λεπτομέρειες υλοποίησης κάποιας εργασίας. Ένας μαθητής μπορεί να το χρησιμοποιήσει για να εργάζεται ηλεκτρονικά πάνω σε εργασίες, κείμενα και γενικότερα στα ερευνητικά του ενδιαφέροντα.

Sakai Product Support Community Try Sakai News / Events Contact Us

Sakai is distributed as free and open source software under the Educational Community License

DENVER 2010
Sakai

Sakai 2010 Conference - Denver, CO
Schedule — Full Text Schedule

Choosing Sakai

Using Sakai

Contributing to Sakai

News

Sakai Foundation Names Ian Dolphin as Executive Director

SoftChalk and Sakai Foundation Announce Integration and Collaboration Partnership

Event Calendar

07/19/10 Campus Technology 2010 - 17th Annual Education Technology Conference

Εικόνα 2.1 Sakai project

Συγκεντρωτικά το Sakai μπορεί να κάνει τα παρακάτω

- Εργασίες. Ένας καθηγητής μπορεί να κάνει ανάθεση εργασιών και να λαμβάνει υποβολές των εργασιών online.
- Chat. Δυνατότητες συζήτησης σε πραγματικό χρόνο.
- Συζητήσεις. Δυνατότητα συζητήσεων σε μη πραγματικό χρόνο (φόρουμ).
- Κοινή χρήση (sharing) σε κείμενα και μελέτες ανάμεσα σε μέλη.
- Αποστολή e-mails, αποθήκευση βαθμολογιών. Προγραμματισμός εργασιών, on-line τεστ και quizzes, profile χρηστών και άλλα.
- Παρουσιάσεις (σαν του powerpoint) σε ζωντανό κοινό, αλλά και αποθήκευσή τους για μετέπειτα μελέτη.

2.2 eConf

Το eConf [12] είναι λογισμικό ανοιχτού κώδικα γραμμένο σε Java και έχει σχεδιαστεί για να καταγράφει παραδόσεις μαθημάτων. Έχει σχεδιαστεί σαν ένα add-one σε HTTP proxy το οποίο έχει τη δυνατότητα να καταγράφει τις σελίδες τις οποίες ο καθηγητής δείχνει κατά τη διάρκεια του μαθήματος και την φωνή του εκφωνητή. Η φωνή και οι σελίδες συγχρονίζονται και οι μαθητές μπορούν να παρακολουθήσουν σε δεύτερο χρόνο τις παραδόσεις. Το eConf έχει υλοποιηθεί πάνω από το W3C Jigsaw proxy.

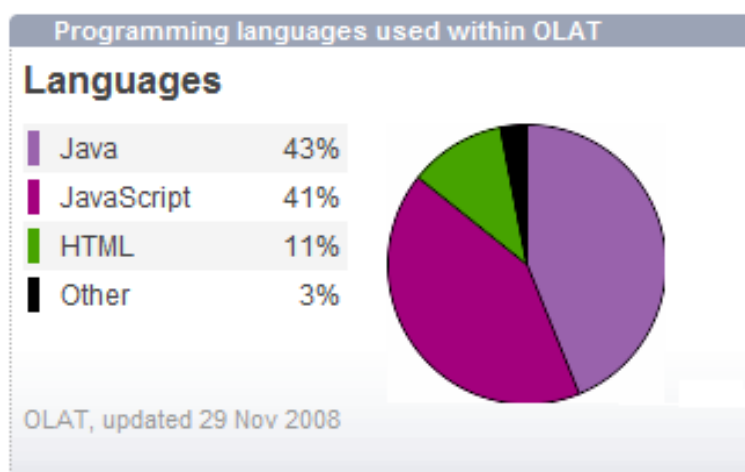
Το eConf παρόλο που είναι υλοποιημένο στην Java, στην γλώσσα προγραμματισμού που θα χρησιμοποιηθεί για την ανάπτυξη της εφαρμογής της πτυχιακής αυτής εργασίας δεν μπορεί να χρησιμοποιηθεί για την παρουσίαση ερευνητικών εργασιών μέσα από δυναμικές ιστοσελίδες. Η χρήση του όμως θα μπορούσε να ήταν η καταγραφή παρουσιάσεων που θα συνόδευαν τα κείμενα των εργασιών. Όταν ένας ερευνητής παρουσίαζε το ερευνητικό του έργο μέσα από κείμενα θα μπορούσε με το λογισμικό αυτό να κάνει και μια εναλλακτική παρουσίαση μέσα από εικόνες, σελίδες αλλά και ήχο. Αυτό μάλιστα θα ήταν ιδανικό για την κατανόηση της έρευνας από άτομα με ειδικές ανάγκες όπως αυτά με πρόβλημα όρασης.

Η χρήση του είναι πραγματικά απλή καθώς το format που χρησιμοποιεί για την καταγραφή των παρουσιάσεων είναι συμβατό με το WebReplay που είναι ένα Java Applet ανοιχτού λογισμικού, βασισμένο στο Java Media Framework (JMF) [13]. Έτσι κάθε σύγχρονος φυλλομετρητής μπορεί να παρουσιάσει χωρίς δυσκολία τις παρουσιάσεις που δημιουργούνται με το eConf.

2.3 OLAT

Το OLAT [14] (Online Learning And Training) αποτελεί μια δικτυοκεντρική υλοποίηση ανοιχτού κώδικα ενός LMS (Learning Management System) καθώς και ενός LCMS (Learning Content Management System) το οποίο χρησιμοποιείται ευρέως στην Σουηδία. Η ανάπτυξη του OLAT ξεκίνησε στο πανεπιστήμιο της Ζυρίχης όπου σήμερα βρίσκεται το μεγαλύτερο installation του και χρησιμοποιείται ενεργά από την ακαδημαϊκή κοινότητα.

Έχει υλοποιηθεί κατά βάση σε Java, σε JavaScript και χρησιμοποιεί και HTML. Η φιλοσοφία του είναι η παροχή ενός framework το οποίο μπορεί να χρησιμοποιηθεί για πολλούς διαφορετικούς σκοπούς που αφορούν το LMS και το LCMS.



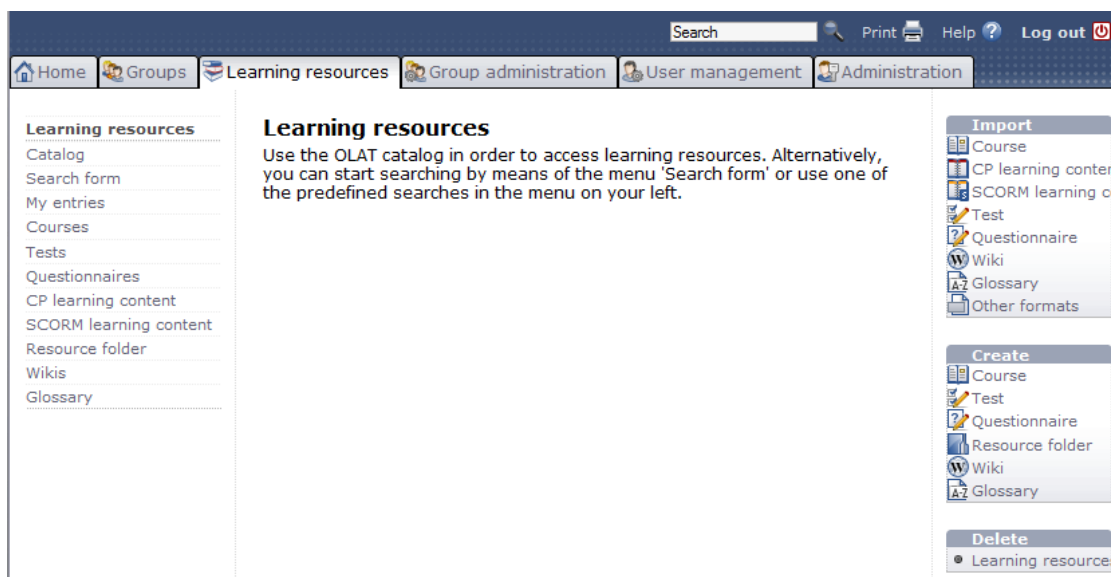
Εικόνα 2.2 Οι γλώσσες προγραμματισμού στις οποίες υλοποιείται το OLAT

Συνολικά 385.577 γραμμές κώδικα έχουν αναπτυχθεί και τη στιγμή αυτή υπάρχουν τουλάχιστον 14 ενεργοί προγραμματιστές που ημερησίως συνεισφέρουν στην περαιτέρω ανάπτυξη της πλατφόρμας.

Το OLAT στηρίζεται για την παρουσίαση των δεδομένων στο YALM [15] (Yet Another Multicolumn Layout) που χρησιμοποιεί XHTML και CSS κώδικα για το presentation layer της εφαρμογής.

Όσον αφορά το scalability, το OLAT είναι σε χρήση σε πανεπιστήμια με 40.000 εγγεγραμμένους χρήστες, περισσότερα από 2.000 ενεργά μαθήματα και μπορεί να εξυπηρετήσει παραπάνω από 900 χρήστες ταυτόχρονα. Τα παραπάνω στοιχεία αφορούν το πανεπιστήμιο της Ζυρίχης όπου είναι εγκατεστημένο σε ένα σύγχρονο υπολογιστικό σύστημα και στηρίζεται σε Apache Tomcat 5 και MySQL.

Το παραπάνω λογισμικό θα μπορούσε να χρησιμοποιηθεί με μερικές τροποποιήσεις για τη δημιουργία του Διαδικτυακού τόπου για τη δημοσίευση επιστημονικών εγγράφων. Ο χρήστης κάνοντας log-in σαν course administrator θα μπορούσε μέσα από την καρτέλα Learning Resources να τοποθετούσε το υλικό (Βιβλία και άρθρα σε μορφή PDF ή Word) και να τα κατηγοριοποιούσε και να τα τοποθετούσε προς προβολή. Κατόπιν μέσω της καρτέλας User Management θα μπορούσε να έδινε δικαίωμα ανάγνωσης για το προαναφερθέν υλικό στους συγκεκριμένους χρήστες που επιθυμούσε.



Εικόνα 2.3 Η σελίδα όπου ο χρήστης επιλέγει τι εγγραφο θα αναζητήσει

Βέβαια οι δυνατότητες του λογισμικού που αναφέρουμε είναι τεράστιες καθώς υλοποιεί σε μεγάλο βαθμό το SCORM. Το SCORM [17] (Sharable Content Object Reference Model) είναι μια συλλογή προδιαγραφών και προτύπων που προέρχονται από πολυετής έρευνα. «Το SCORM παρέχει στους υπεύθυνους για την ανάπτυξη εκπαιδευτικού περιεχομένου ένα πλαίσιο προτύπων που μπορούν να ακολουθήσουν στην ανάπτυξη των μαθημάτων τους έτσι ώστε να είναι επαναχρησιμοποιήσιμα και ανταλλάξιμα σε παγκόσμια κλίμακα». Εφαρμόζει

τις τρέχουσες εξελίξεις στην τεχνολογία κατάρτισης μέσω της χρήσης ενός συγκεκριμένου προτύπου διαχείρισης περιεχομένων (XML), για να εξασφαλίσει σταθερή εφαρμογή της κατάρτισης σε ολόκληρη την e-learning κοινότητα.

Τα οφέλη του SCORM περιληπτικά αφορούν α) την Δια-λειτουργικότητα, τη δυνατότητα χρήσης πολλών πλατφόρμων για πρόσβαση στη γνώση, β) τη Δυνατότητα πρόσβασης από απόσταση γ) την Ικανότητα επαναχρησιμοποίησης του εκπαιδευτικού περιεχομένου (η μέρος του) σε άλλες εφαρμογές δ) τη Συντηρησιμότητα, ώστε το περιεχόμενο να μπορεί να αλλάξει τμηματικά η στο σύνολο του, χωρίς να χρειάζεται δαπανηρός επανασχεδιασμός της δομής και της οργάνωσης του και άλλα πρότυπα.

Στην περίπτωση της πτυχιακής μας, ο χρήστης του συστήματος θα μπορούσε να επισκεφθεί την ιστοσελίδα που θα εγκαθιστούσαμε το OLAT και να έβλεπε τα άρθρα που θα δημοσιοποιούνταν από το περιβάλλον διαχείρισης.

The screenshot shows the OLAT Learning resources interface. At the top, there is a navigation bar with 'Home', 'Groups', and 'Learning resources' tabs. A search bar and 'Print', 'Help', and 'Log out' links are also visible. The main content area is titled 'Learning resources' and shows '14 Entries'. Below this is a table with the following columns: Type, Title of learning resource, Author, Access, and Select. The table lists various course templates and demo courses, all with 'OAUG' access and 'Show content' links. A legend at the bottom explains the access codes: O=Owners of this learning resource (not public), A=All OLAT authors (not public), U=Registered OLAT users (public), and G=Guests (public).

Type	Title of learning resource	Author	Access*	Select
	Course template big	sroth	OAUG	Show content
	Course template small	sroth	OAUG	Show content
	Demo Course	author	OAUG	Show content
	Demo course enrolment	sroth	OAUG	Show content
	Demo course enrol to topics	sroth	OAUG	Show content
	Demo course wiki	sroth	OAUG	Show content
	Demo course with questionnaire	sroth	OAUG	Show content
	Demokurs Einschreibung	sroth	OAUG	Show content
	Demokurs mit Fragebogen	sroth	OAUG	Show content
	Demokurs Themeneinschreibung	sroth	OAUG	Show content
	Demokurs Wiki	sroth	OAUG	Show content
	Kursvorlage gross	sroth	OAUG	Show content
	Kursvorlage klein	sroth	OAUG	Show content
	OLAT - Hilfe, Help, Aide, Aiuto	administrator	OAUG	Show content

*)
O=Owners of this learning resource (not public)
A=All OLAT authors (not public)
U=Registered OLAT users (public)
G=Guests (public)

Εικόνα 2.4 Έγγραφα στο OLAT και δικαιώματα σε αυτά

Το OLAT όπως φαίνεται και στην παραπάνω εικόνα υλοποιεί με έξυπνο τρόπο δικαιώματα στα έγγραφα που δημοσιοποιούνται. Έτσι ο κάθε χρήστης του συστήματος σε ένα

επιστημονικό έγγραφο που ανεβάζει μπορεί να θέσει ένα ή περισσότερα από τα παρακάτω δικαιώματα:

O=Owners of this learning resource (not public)

A=All OLAT authors (not public)

U=Registered OLAT users (public)

G=Guests (public)

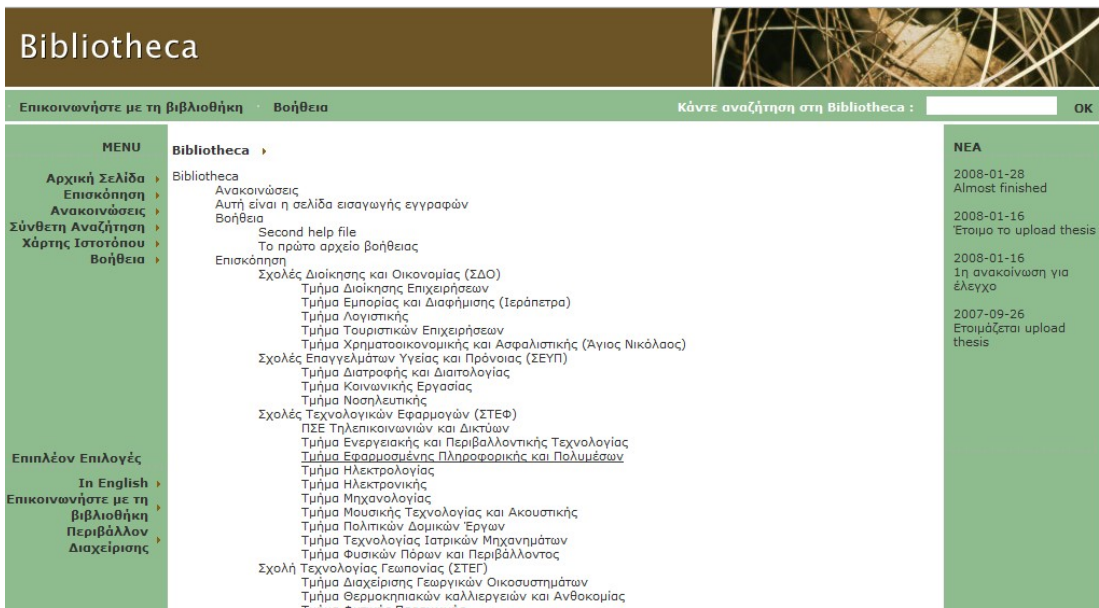
Η φιλοσοφία των παραπάνω δικαιωμάτων μπορεί να χρησιμοποιηθεί και στη δική μας εργασία. Έτσι ένα paper θα μπορεί κατά τη διάρκεια της συγγραφής του να έχει O δικαίωμα, επιτρέποντας μόνο στο δημιουργό του την προβολή του, U δικαίωμα όταν ο χρήστης επιθυμεί μόνο registered χρήστες (μέλη του πανεπιστημίου, καθηγητές και μαθητές) να έχουν πρόσβαση και G για ελεύθερη πρόσβαση στο άρθρο από κάθε χρήστη.

2.4 Bibliotheca



Εικόνα 2.5 Αρχική σελίδα της εφαρμογής bibliotheca

Μια Ελληνική εφαρμογή που παρέχει αντίστοιχες δυνατότητες με αυτές που ζητά η πτυχιακή αυτή εργασία είναι η Bibliotheca [πηγή <http://nefeli.lib.teicrete.gr/>]. Αποτελεί μια εφαρμογή ηλεκτρονικής βιβλιοθήκης του ΤΕΙ Κρήτης όπου τοποθετούνται οι πτυχιακές εργασίες φοιτητών αφού πρώτα κατηγοριοποιηθούν ανά σχολή και τμήμα. Ο χρήστης μπορεί χρησιμοποιώντας το Χάρτη του Ιστοτόπου να επιλέξει τη σχολή και το συγκεκριμένο τμήμα του οποίου τις πτυχιακές εργασίες επιθυμεί να μελετήσει.



Εικόνα 2.6 Χάρτης της εφαρμογής

Κατόπιν παρουσιάζονται στο χρήστη τα εκπαιδευτικά έτη κατά τα οποία κατατέθηκαν οι πτυχιακές εργασίες του συγκεκριμένου τμήματος.

Bibliotheca > **Επισκόπηση** > **Σχολές Τεχνολογικών Εφαρμογών (ΣΤΕΦ)** > **Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων**

- Πτυχιακές 1998
- Πτυχιακές 2003
- Πτυχιακές 2004
- Πτυχιακές 2005
- Πτυχιακές 2006
- Πτυχιακές 2007
- Πτυχιακές 2008
- Πτυχιακές 2009

Εικόνα 2.7 Κατηγοριοποίηση εγγράφων κατά έτος

Κατόπιν επιλέγοντας μια συγκεκριμένη χρονιά, το σύνολο των πτυχιακών εργασιών παρουσιάζονται στο χρήστη. Για κάθε εργασία, ο τίτλος της και το όνομα του φοιτητή που την εκπόνησε παρουσιάζεται.

- A match race sailing dynamometer as a technology and marketing platform
Λιαγγορίδη Ηλέκτρα

- Development of Interactive 3D objects for data visualization
Βγενόπουλος Ιωάννης

- Distributed Logic over Peer to Peer Networks
Τζεφεράκος Renato

- Password cracking
Ορφανουδάκης Φανούριος

- Έλεγχος ρομποτικής πλατφόρμας μέσω κάμερας
Κασιώτης Φίλιππος

- Ανάπτυξη Συστήματος για on line συμπλήρωση ερωτηματολογίων
Γρεβενίτης Κωνσταντίνος

- Ανάπτυξη διαδικτυακής εφαρμογής διαχείρισης προσωπικού και της αντίστοιχης βάσης δεδομένων που απαιτείται για την υποστήριξη της
Σαρμής Δημήτριος

Εικόνα 2.8 Πτυχιακές του ετους 2009

Πατώντας πάνω σε συγκεκριμένη εργασία, περαιτέρω στοιχεία σχετικά με αυτήν παρουσιάζονται στο χρήστη.

Τίτλος	Ανάπτυξη ηλεκτρονικών υπηρεσιών παρουσίασης και πώλησης ακινήτων
Περίληψη	Το ηλεκτρονικό εμπόριο ή αλλιώς e-commerce είναι η παροχή αγαθών και υπηρεσιών, μέσω internet, συνήθως, έναντι αμοιβής. Συνδεόμαστε με μια ιστοσελίδα, η οποία προσφέρει κάποια συγκεκριμένη υπηρεσία, συμβουλευόμαστε τον κατάλογο, επιλέγουμε το προϊόν, που θέλουμε και συμπληρώνουμε την εντολή αγοράς, διευκρινίζοντας τον τρόπο πληρωμής (π.χ. πιστωτική κάρτα, επιταγή ή εξόφληση τοις μετρητοίς, κατά την παραλαβή του προϊόντος).
Λέξεις κλειδιά	Ηλεκτρονικό εμπόριο Αγορές Διαδίκτυο
Δημιουργός	Καλλέργη Ελένη Μαρία Φαναριώτη Σοφία
Χρονιά	2009

Ψηφιακό Τεκμήριο

Εικόνα 2.9 Λεπτομέρειες επιλεγμένης πτυχιακής εργασίας

Για παράδειγμα στην παρακάτω εικόνα παρουσιάζεται τόσο ο τίτλος όσο και η περίληψη, ο δημιουργός, η χρονιά που εκπονήθηκε και ορισμένες λέξεις κλειδιά που μπορούν να βοηθήσουν στην εύρεση της εργασίας μέσα από τη μηχανή αναζήτησης. Αν ο χρήστης

πατήσει την επιλογή «Ψηφιακό τεκμήριο» τότε το κείμενο της εργασίας παρουσιάζεται σε μορφή PDF αρχείου.

Η επιλογή «Σύνθετη αναζήτηση» παρουσιάζει μια φόρμα στο χρήστη για να εισάγει ορισμένα πεδία ή να επιλέξει συγκεκριμένο τμήμα για την αναζήτησή του.

The screenshot shows the search interface for Bibliotheca. At the top, it says 'Bibliotheca > Σύνθετη Αναζήτηση'. Below this, there are two search input fields, each with a dropdown menu set to 'Όλα τα πεδία'. Between the two fields is a dropdown menu set to 'ΚΑΙ'. Below the second input field is a dropdown menu labeled 'Περιορίστε ανάλογα με το τμήμα' with the option 'Όλα τα τμήματα' selected. At the bottom of the form is a button labeled 'Αναζήτηση'.

Εικόνα 2.10 Σύνθετη αναζήτηση στην εφαρμογή bibliotheca

Αναζητώντας με την τιμή «Tomcat» η Bibliotheca επιστρέφει 4 αποτελέσματα τα οποία παρουσιάζονται ανά τμήμα. Υπάρχει βέβαια και η επιλογή «Ταξινόμηση βάση χρονιάς».

Αναζήτηση

Τοπικός κατάλογος: 4 αποτελέσματα

Προβολή αποτελεσμάτων 1-4 **Ταξινόμηση βάση χρονιάς**

Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

▶ Δημιουργία υποδομής τηλεσυνεργασία μέσω διαδικτύου (2009)

▶ Ανάπτυξη εφαρμογής διαχείρισης γραμματείας τμήματος με αντικειμενοστραφή προγραμματισμό σε διαδικτυακό περιβάλλον (2007)

▶ Διαχείρισης Θησαυρών Όρων SIS-TMS (2008)

Τμήμα Ηλεκτρονικής

▶ Σχεδίαση και Υλοποίηση Λογισμικού για τη Δήλωση των Μαθημάτων από τους Σπουδαστές Μέσω Διαδικτύου (2008)

Προβολή αποτελεσμάτων 1-4 **Ταξινόμηση βάση χρονιάς**

Εικόνα 2.11 Αποτελέσματα αναζήτησης

2.5 Συγκρίσεις και συμπεράσματα

Στο κεφάλαιο αυτό παρουσιάστηκαν ορισμένες από τις εφαρμογές από Ελληνικούς και ξένους ιστοτόπους που αφορούν τη διαχείριση επιστημονικών εγγράφων. Πολλές πλατφόρμες έχουν υλοποιηθεί κυρίως από πανεπιστήμια για να εξυπηρετήσουν την εσωτερική τους ανάγκη διαχείρισης εγγράφων και δεδομένων. Μάλιστα σε πανεπιστήμια του εξωτερικού οι δυνατότητες που υπάρχουν στους ακαδημαϊκούς, επιτρέπουν μέχρι και το δυναμικό έλεγχο μιας εργασίας κατά πόσο είναι προϊόν λογοκλοπής ή είναι πρωτότυπη εργασία.

Η πλέον αξιόλογη ελληνική προσπάθεια της Bibliotheca περιέχει πολλές από τις δυνατότητες που πρέπει να υλοποιηθούν και στην εργασία αυτή. Λαμβάνοντας όμως υπόψη το σενάριο όπου μια ομάδα εργασίας μελετά κάποια συγκεκριμένη τεχνολογία και καταγράφει τα αποτελέσματα της έρευνας σε ένα αρχείο, αν το αρχείο αυτό βρεθεί σε ξένα χέρια, τότε κάποιος τρίτος ίσως δημοσιεύσει το άρθρο με το δικό του όνομα ή προλάβει να καταγράψει μια πατέντα με τα δικά του συμφέροντα.

Μιας μορφής προστασίας εγγράφων είναι δηλαδή απαραίτητη. Πηγή έμπνευσης για τη μορφή προστασίας αποτέλεσε το OLAT. Τα έγγραφα που ανεβάζουν στο σύστημα οι χρήστες έχουν ένα πεδίο σχετικά με την ορατότητα:

O=Owners of this learning resource (not public)

A=All OLAT authors (not public)

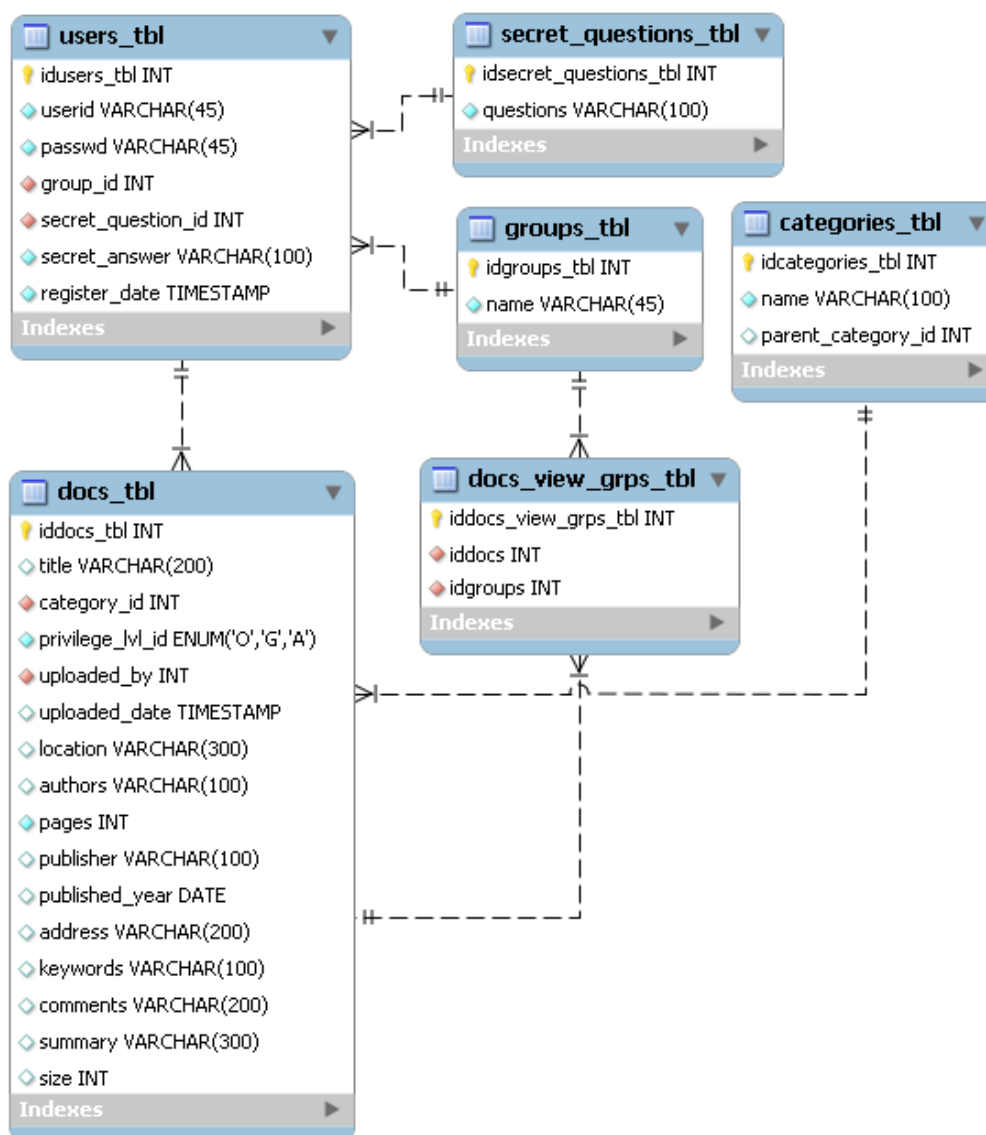
U=Registered OLAT users (public)

G=Guests (public)

Μια παραλλαγή της παραπάνω δυνατότητας θα υλοποιηθεί και στην πτυχιακή αυτή εργασία και θα παρουσιαστεί στη συνέχεια.

ΚΕΦΑΛΑΙΟ 3. Σχεδίαση βάσης δεδομένων

Η βάση δεδομένων που σχεδιάστηκε με τη χρήση του MySQL WorkBench είναι η:



Εικόνα 3.1 Επισκόπηση της βάσης δεδομένων της εφαρμογής

- Ο πίνακας **users_tbl** θα περιέχει του εγγεγραμμένους χρήστες του συστήματος ένας εκ των οποίων θα είναι και ο διαχειριστής του. Τα στοιχεία που θα αποθηκεύονται για

κάθε χρήστη θα είναι: το όνομα χρήστη (userid) που θα είναι μάλιστα το e-mail του χρήστη, ο κωδικός πρόσβασης του χρήστη στο σύστημα (passwd), η ομάδα στην οποία θα ανήκει (group_id), η ερώτηση ασφαλείας που έχει επιλέξει ο χρήστης για την ανάκτηση κωδικού πρόσβασης (secret_question_id), η απάντηση του χρήστη στην ερώτηση ασφαλείας (secret_answer) και τέλος η ημερομηνία δημιουργίας του λογαριασμού του (register_date).

- Ο πίνακας **groups_tbl** θα περιέχει τις ομάδες εργασίας του συστήματος. Τα στοιχεία του πίνακα αυτού θα είναι μόνο το όνομα της ομάδας (name) και τον χαρακτηριστικό αριθμό της ομάδας (idgroups_tbl). Σαν όνομα ομάδας θα μπορούν να χρησιμοποιηθούν χαρακτηριστικά όπως «ομάδα προπτυχιακών», «ομάδα καθηγητών», «ομάδα μεταπτυχιακών» ή «ομάδα καθηγητών». Ο πίνακας αυτός χρησιμοποιείται από τον users_tbl για να δηλωθεί σε ποια ομάδα ανήκει ο κάθε χρήστης. Στην εξέλιξη του λογισμικού αυτού ένας χρήστης θα μπορούσε να ανήκει σε περισσότερο από μια ομάδες (!).
- Ο πίνακας **secret_questions_tbl** θα περιέχει μια σειρά ερωτήσεων ασφαλείας (questions) και τον χαρακτηριστικό αριθμό της ομάδας (idsecret_questions_tbl). Όταν ένας χρήστης θα δημιουργεί λογαριασμό στο σύστημα θα επιλέγει μια από τις υπάρχουσες ερωτήσεις ασφαλείας όπως για παράδειγμα «Ποιο είναι το πατρικό όνομα της μητέρας σου?». Κατόπιν θα δίνει την απάντηση στην ερώτηση αυτή. Αν ξεχάσει το κωδικό πρόσβασης του στο σύστημα, η ερώτηση αυτή θα εμφανίζεται και αν απαντήσει σωστά σε αυτή, θα του αποστέλλεται ο κωδικός πρόσβασης στο e-mail του.
- Ο πίνακας **docs_tbl** θα περιέχει στοιχεία σχετικά με τα αρχεία που ανεβαίνουν στο σύστημα. Για κάθε αρχείο θα αποθηκεύονται στοιχεία όπως: ο τίτλος του αρχείου (title), η κατηγορία στην οποία θα ανήκει (category), η ορατότητα του αρχείου αυτού στους υπόλοιπους χρήστες (privilege_lvl_id), ο ιδιοκτήτης του αρχείου (uploaded_by), η ημερομηνία κατά την οποία το αρχείο αυτό τοποθετήθηκε στο σύστημα (uploaded_date), η τοποθεσία του αρχείου αυτού στον σκληρό δίσκο (location), οι συγγραφείς του εγγράφου αυτού (authors), ο αριθμός των σελίδων (pages), ο εκδότης – σε περίπτωση που έχει δημοσιευθεί το έγγραφο αυτό (publisher), η ημερομηνία έκδοσης (published_year), λέξεις κλειδιά που θα μπορούν να βοηθήσουν κατά την αναζήτηση εγγράφων (keywords), κάποια σχόλια σχετικά με το άρθρο (comments), μια περίληψη (summary) και τέλος το μέγεθος του αρχείου σε bytes (size).

- Ο πίνακας **docs_view_grps_tbl** που είναι το προϊόν της κανονικοποίησης του πίνακα docs με τον groups. Επειδή η σχέση των δύο είναι πολλά προς πολλά ο πίνακας docs_view_grps_tbl ουσιαστικά συγκρατεί σε ποια ομάδα έχει πρόσβαση σε ποια αρχεία.
- Ο πίνακας **categories_tbl** περιέχει τις κατηγορίες και υποκατηγορίες στις οποίες θα εντάσσονται τα αρχεία. Συγκρατεί στοιχεία όπως το όνομα της κατηγορίας (name) και σε περίπτωση που είναι υπο-κατηγορία το id της κύριας κατηγορίας στην οποία ανήκει (parent_category_id). Τέλος οι κύριες κατηγορίες θα έχουν σαν τιμή το NULL ως parent_category_id.

Ο κώδικας δημιουργίας της παραπάνω βάσης δεδομένων μαζί με τα foreign κλειδιά βρίσκεται στο «Παράρτημα - SQL Κώδικας».

ΚΕΦΑΛΑΙΟ 4. Περιβάλλον ανάπτυξης

Οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής είναι οι παρακάτω. Μάλιστα να σημειωθεί ότι όλες οι παρακάτω τεχνολογίες μπορούν να χρησιμοποιηθούν τόσο σε περιβάλλον Windows όσο και Linux:

4.1 Apache Tomcat

Ο Tomcat Application Server είναι ένα servlet container, που υποστηρίζει πλήρως τις τεχνολογίες των Java Servlets και JavaServer Pages, και μπορεί να λειτουργήσει είτε αυτόνομα, είτε σαν plug-in στον Apache HTTP Server. Βασικότατο χαρακτηριστικό του Tomcat αποτελεί το γεγονός ότι είναι open source λογισμικό, δηλαδή μπορεί καθένας να το χρησιμοποιήσει δωρεάν.

4.1.1 Δομή του Tomcat και των Web Applications που φιλοξενεί

Η θέση στο σύστημα στην οποία εγκαθίσταται ο Tomcat ορίζεται με τη μεταβλητή %CATALINA_HOME%. Μέσα στον κατάλογο εγκατάστασης δημιουργούνται κατά την εγκατάσταση του Tomcat κάποιοι υποκατάλογοι με συγκεκριμένες λειτουργικές σημασίες (π.χ. %CATALINA_HOME%\temp). Ο σημαντικότερος από αυτούς για έναν προγραμματιστή μπορεί να θεωρηθεί ο κατάλογος %CATALINA_HOME%\webapps. Μέσα σε αυτόν τον κατάλογο τοποθετούνται οι Web Applications (τα servlets) που θέλουμε να τρέχουν στον Tomcat. Κάθε webapp (web application) τοποθετείται σε έναν δικό της κατάλογο (π.χ. %CATALINA_HOME%\webapps\MyWebApp), ο οποίος με τη σειρά του έχει μια (αρχικά) καθορισμένη δομή από αρχεία και υποκαταλόγους.

Βασικός κατάλογος μιας webapp του Tomcat είναι ο MyWebApp\WEB-INF, στον οποίο βρίσκονται μεταξύ άλλων τα πακέτα με τις κλάσεις που χρησιμοποιούνται από την εφαρμογή (WEB-INF\classes), καθώς και οι βιβλιοθήκες που χρησιμοποιούνται από τις κλάσεις (WEB-INF\lib). Ακόμη, κάθε webapp έχει ένα αρχείο που ονομάζεται web.xml και το οποίο καθορίζει

διάφορες παραμέτρους της εφαρμογής μέσα από μια συγκεκριμένη XML δομή. Πιο συγκεκριμένα, αυτό το αρχείο το διαβάζει ο Tomcat κάθε φορά που ξεκινά η λειτουργία του και γνωρίζει έτσι διάφορα χαρακτηριστικά της webapp, όπως τα Servlet της κ.α.

Η εγκατάσταση του Tomcat σε ένα σύστημα με Windows OS αποτελεί μια αρκετά απλή διαδικασία. Εφόσον έχει κανείς στον υπολογιστή του εγκατεστημένο το JDK (<http://java.sun.com/j2se/1.5.0/download.jsp>), αρκεί να κατεβάσει και να τρέξει το .exe αρχείο της εγκατάστασης του Tomcat από τη διεύθυνση <http://jakarta.apache.org/tomcat/>.



Apache Tomcat

4.2 Java SDK

Το Java SDK αποτελεί το runtime, την πλατφόρμα της Java που είναι απαραίτητη τόσο για την εκτέλεση του Application Server (του Tomcat) όσο και για το compilation του κώδικα σε εκτελέσιμα αρχεία. Η έκδοση που χρησιμοποιήθηκε ήταν η πλέον πρόσφατη, η 6.0

Η Java είναι μια αντικειμενοστραφής (object – oriented) γλώσσα προγραμματισμού υψηλού επιπέδου. Σχεδιάστηκε με στόχο να είναι (α) αντικειμενοστραφής, (β) κατανεμημένη (distributed), (γ) απλή, (δ) πολυνηματική (multi-threaded), (ε) ασφαλής, (στ) ανεξάρτητη υπολογιστικής πλατφόρμας.

Η Java υποστηρίζει εγγενώς τις αρχές του αντικειμενοστραφούς προγραμματισμού. Σε αντιδιαστολή, οι Basic και C είναι διαδικαστικές γλώσσες (procedural languages) βασισμένες στο μοντέλο του δομημένου προγραμματισμού (structured programming).

Η Java είναι μια πλήρης γλώσσα προγραμματισμού. Υποστηρίζει εξαιρέσεις, νηματική εκτέλεση (threaded execution), και άλλα. Επίσης παρέχει τυποποιημένες βιβλιοθήκες κλάσεων (Java class libraries) μέσω των οποίων υποστηρίζονται γραφικά και γραφικές διεπαφές (Graphical User Interface, GUI), πολυμέσα, επικοινωνία πάνω σε δίκτυα, επαναχρησιμοποιούμενα εξαρτήματα (components), σύνδεση με βάσεις δεδομένων, και άλλα. Βασικότατο χαρακτηριστικό της Java είναι το γεγονός ότι δεν εξαρτάται από την πλατφόρμα και το λειτουργικό σύστημα (platform independent). Αυτό σημαίνει ότι μία εφαρμογή

γραμμένη σε Java μπορεί να εκτελείται σε οποιοδήποτε περιβάλλον χωρίς καμία τροποποίηση. Η ανεξαρτησία αυτή βασίζεται στη χρήση μιας εικονικής μηχανής (Java Virtual Machine – JVM), η οποία προσομοιώνει έναν ιδεατό επεξεργαστή, ικανό να εκτελεί προγράμματα γραμμένα σε Java. Η εικονική μηχανή δέχεται μεταγλωττισμένα προγράμματα και μετατρέπει τις εντολές τους σε εντολές εκτελέσιμες από το εκάστοτε λειτουργικό σύστημα.



Με τον τρόπο αυτό, κάθε υπολογιστής, ανεξαρτήτως λειτουργικού συστήματος (Windows, Linux, Solaris, Macintosh, κλπ.), ο οποίος διαθέτει την εικονική μηχανή, μπορεί να εκτελέσει προγράμματα γραμμένα σε Java, ανεξάρτητα από τον μεταγλωττιστή που έχει χρησιμοποιηθεί για την δημιουργία τους. Αυτός ο συνδυασμός ανεξαρτησίας από επεξεργαστές και λειτουργικά συστήματα είναι που κάνει την Java ιδιαίτερα ανταγωνιστική και επιτρέπει τον χαρακτηρισμό της ως «Write-Once-Run-Anywhere» γλώσσα. Ακόμη κι όταν ένα πρόγραμμα Java λειτουργεί σε διαφορετικές υπολογιστικές πλατφόρμες, στην πραγματικότητα εκτελείται πάνω στην ίδια «μηχανή» – την ιδεατή μηχανή Java (Java Virtual Machine, JVM). Η ιδεατή μηχανή Java είναι ένα πρόγραμμα που διερμηνεύει τα εκτελέσιμα προγράμματα Java σε εντολές που μπορεί να εκτελέσει η υπολογιστική πλατφόρμα - στόχος. Η Sun έχει παραχωρήσει σε αρκετές εταιρίες (Oracle, Novell, IBM, και άλλα) την άδεια να κατασκευάσουν ιδεατές μηχανές Java για διάφορες υπολογιστικές πλατφόρμες. Καθώς η ιδεατή μηχανή Java έχει μεταφερθεί σε μεγάλη γκάμα από υπολογιστικές πλατφόρμες και είναι ευρέως εγκατεστημένη, δημιουργούνται συνεχώς νέες γλώσσες προγραμματισμού ή παραλλαγές παλαιότερων γλωσσών προγραμματισμού (Basic, Lisp, Prolog) που μεταγλωττίζονται ώστε το πρόγραμμα – στόχος να εκτελείται στην ιδεατή μηχανή Java. Σύμφωνα με μια πρόσφατη καταμέτρηση (<http://www.robert-tolksdorf.de/vmlanguages.html>), υπάρχουν 200 τέτοια εργαλεία που δίνουν τη δυνατότητα σε πηγαίο κώδικα που δεν είναι Java να μεταφράζεται και να εκτελείται στην ιδεατή μηχανή Java.

Παρά τις διαφορές που έχουν όλες οι υπολογιστικές πλατφόρμες, εντάσσονται στο ίδιο ιδεατό υπολογιστικό μοντέλο: το εκτελέσιμο πρόγραμμα ανακαλείται από τη δευτερεύουσα μνήμη, που είναι συνήθως παραμένουσα (nonvolatile), πχ δίσκος ή μνήμη flash, στην πρωτεύουσα ευμετάβλητη (volatile) μνήμη τυχαίας προσπέλασης, πχ RAM, και από εκεί μεταφέρεται εντολή προς εντολή στην κεντρική μονάδα επεξεργασίας. Παρατηρήστε ότι λεπτομέρειες, όπως το μέγεθος της μνήμης, η ύπαρξη καταχωρητών (registers), τα χαρακτηριστικά των μονάδων εισόδου / εξόδου, σκόπιμα δεν περιλαμβάνονται στις προδιαγραφές του αφαιρετικού μοντέλου

4.3 MySQL εξυπηρετητής

Για την αποθήκευση των δεδομένων πρέπει να εγκατασταθεί και να ρυθμιστεί εξυπηρετητής βάσης δεδομένων. Η MySQL αποτελεί έναν από τους καλύτερους δωρεάν εξυπηρετητές σχεσιακών βάσεων δεδομένων.

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακής βάση ανοικτού κώδικα όπως λέγεται (relational database management system - RDBMS) που χρησιμοποιεί την Structured Query Language (SQL), την πιο γνωστή γλώσσα για την προσθήκη, την πρόσβαση και την επεξεργασία δεδομένων σε μία Βάση Δεδομένων. Επειδή είναι ανοικτού κώδικα (open source), οποιοσδήποτε μπορεί να κατεβάσει την MySQL και να την διαμορφώσει σύμφωνα με τις ανάγκες του σύμφωνα πάντα με την γενική άδεια που υπάρχει. Η MySQL είναι γνωστή κυρίως για την ταχύτητα, την αξιοπιστία, και την ευελιξία που παρέχει. Η MySQL αυτή τη στιγμή μπορεί να λειτουργήσει στα περιβάλλοντα Linux, Unix, και Windows.



4.4 Connector/J (MySQL JDBC driver).

Για την επικοινωνία μιας γλώσσας προγραμματισμού με τον εξυπηρετητή βάσης δεδομένων χρειάζεται κάποια βιβλιοθήκη που παρέχει ένα API επικοινωνίας. Στην περίπτωση της Java, για να επικοινωνήσει με τη MySQL απαιτείται η ύπαρξη του Connector/J driver

[πηγή : <http://www.mysql.com/products/connector/>].

4.5 Struts

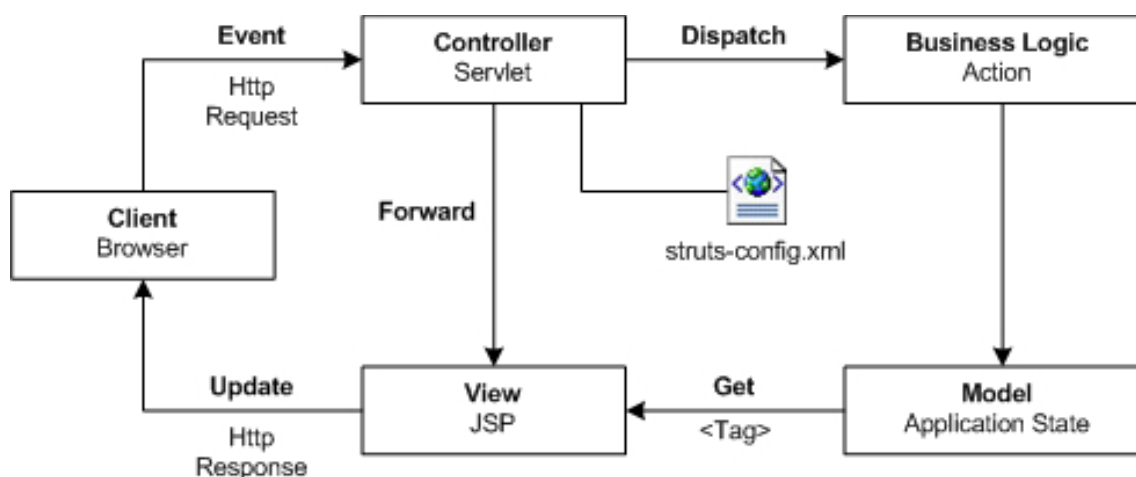
Το Struts αποτελεί ένα Framework που διευκολύνει στην ανάπτυξη web εφαρμογών με java. Χρησιμοποιείται στο web layer και παρέχει διάφορες χρήσιμες λειτουργίες σχετικά με την οργάνωση και εκτέλεση του κώδικα. Στην απλούστερη μορφή του είναι μια συλλογή από κλάσεις και interfaces τα οία συνεργάζονται για να λύσουν ένα συγκεκριμένο τύπο προγραμματιστικού προβλήματος. Ένα framework διακρίνεται από τα εξής χαρακτηριστικά. Αποτελείται από κλάσεις ή components καθένα από τα οποία παρέχει μια αφηρημένη υλοποίηση μίας συγκεκριμένης έννοιας. Ορίζει πώς αυτές οι υλοποιήσεις συνεργάζονται για να λύσουν το πρόβλημα. Τα μέρη από τα οποία αποτελείται είναι επαναχρησιμοποιήσιμα. Επιπλέον παρέχει γενική συμπεριφορά την οποία μπορούν να χρησιμοποιήσουν πολλοί τύποι εφαρμογών. Σε αντίθεση με μία βιβλιοθήκη, της οποίας τις κλάσεις και τις διεπαφές απλά τις καλεί κάποιος για να τις χρησιμοποιήσει, ένα framework προσφέρει μία βάση πάνω στην οποία μπορεί ο προγραμματιστής να χτίσει την εφαρμογή του, απλά επεκτείνοντας τις κλάσεις του framework.



4.5.1 Ειδικότερα για το Struts framework

Το Struts Framework βασίζεται στην τεχνολογία των Java Servlets και σε μικρότερο βαθμό στην τεχνολογία των JSP. Για το λόγο αυτό είναι αναγκαία η ύπαρξη ενός web container, το οποίο φιλοξενεί την εφαρμογή που χρησιμοποιεί το Struts Framework. Με αυτό τον τρόπο η εφαρμογή δεν χρειάζεται να χειρίζεται τις HTTP αιτήσεις, αφού το Struts Framework έχει τη δυνατότητα να χρησιμοποιεί τέτοιου είδους υπηρεσίες που προσφέρονται από το web container.

Η λειτουργία του Struts Framework είναι σύμφωνη με το μοντέλο MVC6 με την διαφορά ότι τα servlets δεν χρειάζεται να απασχολήσουν τον προγραμματιστή αφού προσφέρονται έτοιμα από το Framework.



Εικόνα 4.1 Struts overview

Θα χρησιμοποιηθεί η έκδοση Apache struts 2.1

[πηγή : <http://struts.apache.org/download.cgi#struts218>]

4.6 Βιβλιοθήκες

4.7.1 JSTL.

Το JavaServer Pages Standard Tag Library είναι μια βιβλιοθήκη που περιέχει ορισμένα απλά tags για διαχείριση δομών δεδομένων όπως iterators, sql commands και πολλά άλλα

[πηγή : <http://java.sun.com/products/jsp/jstl/>].

4.7.2 Apache commons DBCP

Το DBCP αποτελεί μια βιβλιοθήκη από την Jakarta που λύνει ένα βασικό πρόβλημα. Όταν ένας χρήστης που χρησιμοποιεί την εφαρμογή πρέπει να αποκτήσει πρόσβαση στη βάση δεδομένων τότε μια χρονοβόρα διαδικασία συμβαίνει (η αρχικοποίηση της σύνδεσης). Χρησιμοποιώντας το DBCP η σύνδεση με τη βάση δεδομένων συμβαίνει μια φορά από τον Application Server (π.χ. Tomcat) και κατόπιν σε κάθε περίπτωση που χρειαστεί η σύνδεση με τη βάση δεδομένων τότε δεν εγκαθιδρύεται μια νέα, αλλά χρησιμοποιείται η προϋπάρχουσα.

[πηγή <http://commons.apache.org/dbcp/>].

commons DBCP

4.7.3 Apache commons Mail.

Για την αποστολή από την εφαρμογή απαιτείται η ύπαρξη μιας βιβλιοθήκης που υλοποιεί τα πρωτόκολλα αποστολής email. Η βιβλιοθήκη που χρησιμοποιήθηκε ήταν η Apache commons Mail [πηγή : <http://commons.apache.org/email/>].

commons email

4.7.4 Apache commons Validator.

Η εφαρμογή που θα υλοποιηθεί θα περιέχει αρκετές φόρμες όπου οι χρήστες θα δημιουργούν π.χ. νέους λογαριασμούς, θα τοποθετούν αρχεία στο σύστημα και άλλα. Πολλά από τα στοιχεία που θα τοποθετηθούν στο σύστημα θα πρέπει να επικυρωθούν. Για παράδειγμα το σύστημα δεν πρέπει να επιτρέπει την τοποθέτηση του «α12» ως ημερομηνίας ή του «α – 2» ως email του χρήστη. Για την επικύρωση των δεδομένων αυτών θα χρησιμοποιηθεί η βιβλιοθήκη Apache Commons Validator [πηγή : <http://commons.apache.org/validator/>].

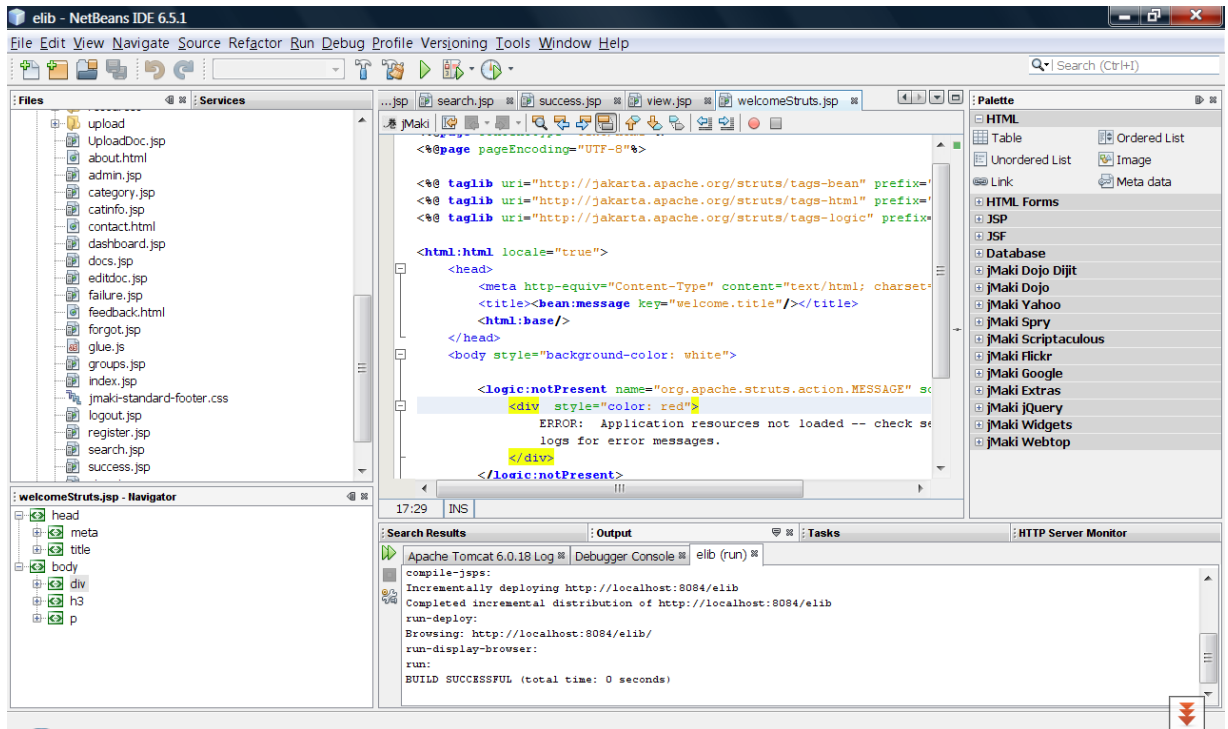
commons *validator*

4.7 Πλατφόρμα NetBeans.

Για την ανάπτυξη του λογισμικού, δηλαδή των σελίδων JSP αλλά και των αρχείων Java θα χρησιμοποιηθεί το NetBeans. Το NetBeans IDE είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών που βασίζεται στην πλατφόρμα NetBeans. Η πλατφόρμα NetBeans IDE επιτρέπει την ανάπτυξη της εφαρμογής NetBeans IDE από ένα σύνολο συστατικών μονάδων λογισμικού που ονομάζονται modules. Ένα module είναι ένα συμπιεσμένο αρχείο Java το οποίο περιέχει κλάσεις, που είναι προορισμένες να αλληλεπιδρούν με NetBeans Open APIs, και ένα αρχείο που το χαρακτηρίζει ως module. Εφαρμογές που έχουν αναπτυχθεί πάνω σε modules μπορούν να επεκταθούν προσθέτοντας νέα modules.

Το NetBeans IDE διανέμεται δωρεάν και είναι λογισμικό ανοιχτού κώδικα. Υποστηρίζει την ανάπτυξη όλων των ειδών εφαρμογών Java (Java SE, web, EJB και mobile εφαρμογές). Επιπλέον, υποστηρίζει την ανάπτυξη εφαρμογών με την χρήση του Ant τον έλεγχο εκδόσεων, με την χρήση των CVS, Subversion, Mercurial και Clear case, και την επεξεργασία κώδικα (refactoring).

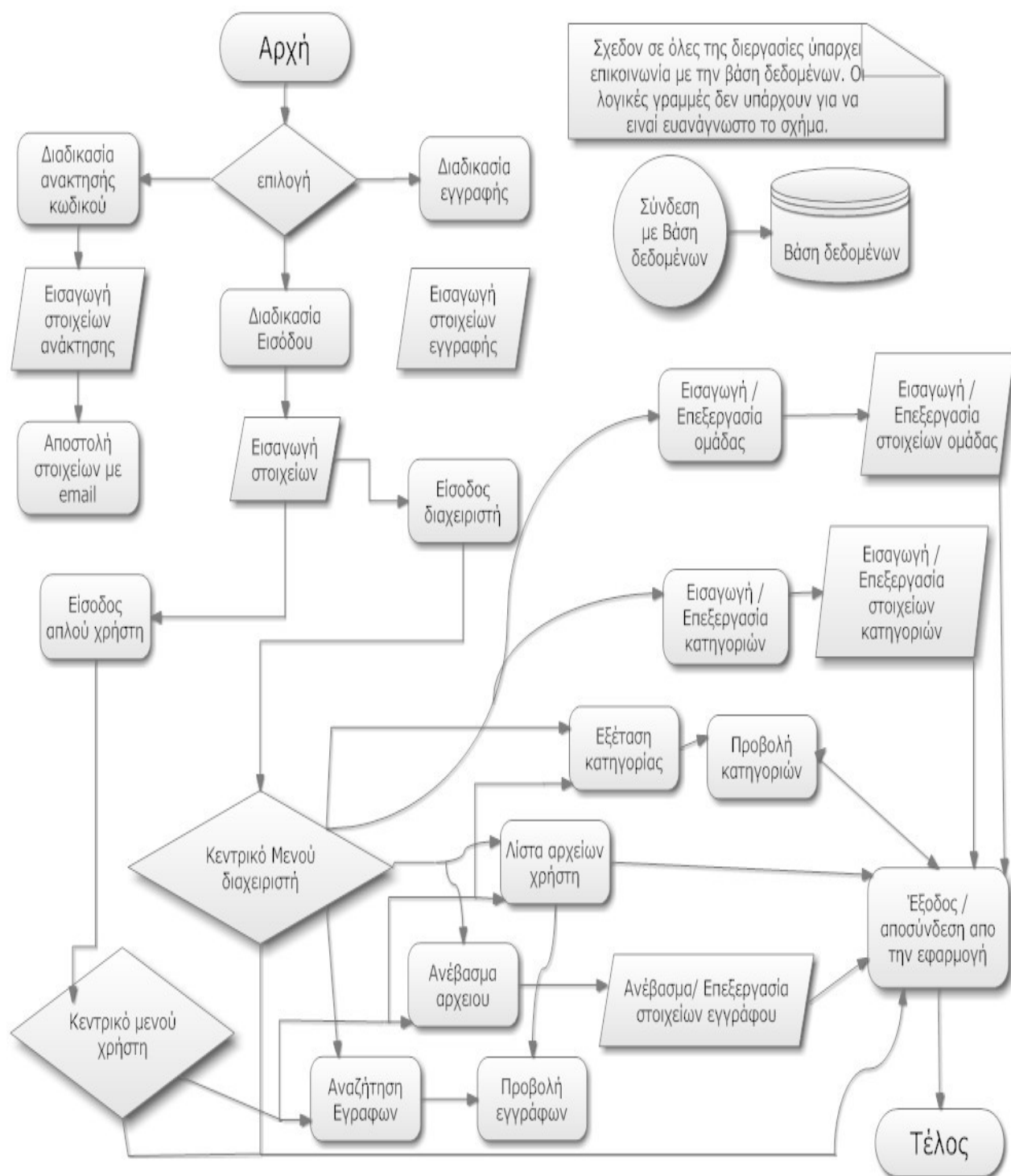
Όλες οι λειτουργίες του IDE παρέχονται από τα modules. Κάθε module παρέχει μία απαιτούμενη λειτουργία όπως υποστήριξη ανάπτυξης εφαρμογών Java, επεξεργασία κώδικα, υποστήριξη συστημάτων διαχείρισης εκδόσεων CVS και SVN. Το NetBeans IDE παρέχει όλα τα απαιτούμενα modules για την ανάπτυξη εφαρμογών Java στο αρχικό πακέτο που κατεβάζει κανείς, το οποίο επιτρέπει στον χρήστη να αρχίσει άμεσα την ανάπτυξη μιας εφαρμογής. Τα modules επιτρέπουν την επέκταση του IDE. Νέες λειτουργίες, όπως η υποστήριξη επιπλέον γλωσσών προγραμματισμού μπορούν να προστεθούν μετά την αρχική εγκατάσταση



Εικόνα 4.2 Άποψη του IDE Netbeans

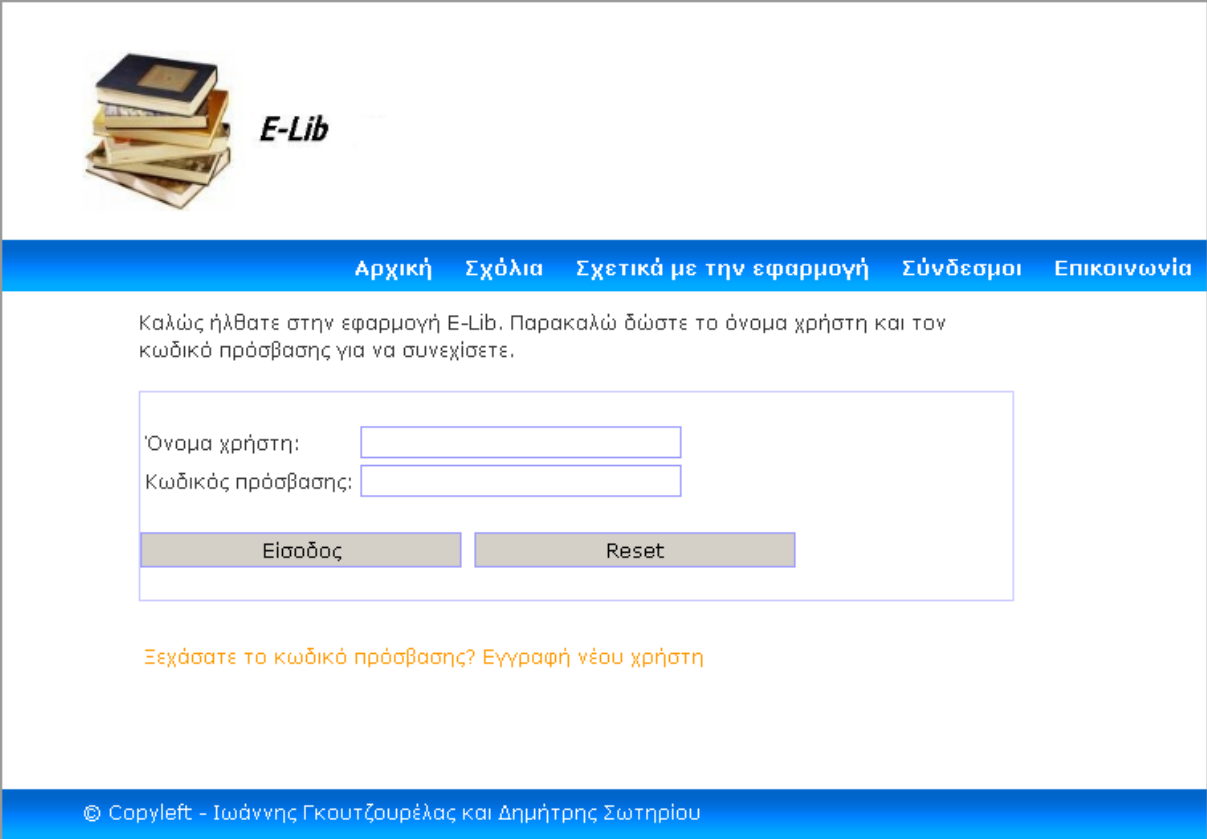
ΚΕΦΑΛΑΙΟ 5. Παρουσίαση εφαρμογής


Σε αυτό το κεφάλαιο θα γίνει μία παρουσίαση των βασικών λειτουργιών της εφαρμογής, αρχικά παραθέτουμε ένα εποπτικό διάγραμμα για την ευκολότερη κατανόησή της.



Διάγραμμα 5.1 Εποπτικό διάγραμμα της εφαρμογής

Ξεκινώντας η εφαρμογή, παρουσιάζει στο χρήστη επιλογές για ανάκτηση κωδικού πρόσβασης στο link «Ξεχάσατε τον κωδικό πρόσβασης» καθώς και την εγγραφή νέου χρήστη. Στο μενού επιλογών στην κορυφή, στο μπλε πλαίσιο ο χρήστης μπορεί να επιλέξει «Αρχική», «Σχόλια», «Σχετικά με την εφαρμογή» «Σύνδεσμοι» και «Επικοινωνία».



 **E-Lib**

Αρχική Σχόλια Σχετικά με την εφαρμογή Σύνδεσμοι Επικοινωνία

Καλώς ήλθατε στην εφαρμογή E-Lib. Παρακαλώ δώστε το όνομα χρήστη και τον κωδικό πρόσβασης για να συνεχίσετε.

Όνομα χρήστη:

Κωδικός πρόσβασης:

Είσοδος Reset

[Ξεχάσατε το κωδικό πρόσβασης? Εγγραφή νέου χρήστη](#)

© Copyleft - Ιωάννης Γκουτζουρέλας και Δημήτρης Σωτηρίου

Εικόνα 5.1: Η κεντρική οθόνη της εφαρμογής

Αν ο χρήστης δεν τοποθετήσει σωστό όνομα χρήστη ή κωδικό πρόσβασης, κατάλληλο μήνυμα του εμφανίζεται «Ο χρήστης δεν υπάρχει!! Παρακαλώ δημιουργήστε νέο λογαριασμό ή βάλτε τα σωστά στοιχεία σας.» με κόκκινα έντονα γράμματα:

Καλώς ήλθατε στην εφαρμογή E-Library. Παρακαλώ δώστε το όνομα χρήστη και τον κωδικό πρόσβασης για να συνεχίσετε.

Ο χρήστης δεν υπάρχει!! Παρακαλώ δημιουργήστε νέο λογαριασμό ή βάλτε τα σωστά στοιχεία σας.

Όνομα χρήστη:

Κωδικός πρόσβασης:

Εικόνα 5.2: Σε περίπτωση λανθασμένης εισόδου κατάλληλο μήνυμα εμφανίζεται

Σε περίπτωση που ο χρήστης δεν συμπληρώσει είτε το όνομα χρήστη είτε τον κωδικό πρόσβασης καθόλου τότε ένα άλλο μήνυμα εμφανίζεται που τον ενημερώνει για το λάθος του.

Καλώς ήλθατε στην εφαρμογή E-Library. Παρακαλώ δώστε το όνομα χρήστη και τον κωδικό πρόσβασης για να συνεχίσετε.

Δεν επιτρέπεται να αφήνεται κενά τα α) όνομα χρήστη και β) κωδικό πρόσβασης!!

Όνομα χρήστη:

Κωδικός πρόσβασης:

Εικόνα 5.3: Η φόρμα εισόδου δεν επιτρέπει να συμπληρωθούν κενά στοιχεία

Σε περίπτωση που ο χρήστης έχει ξεχάσει τον κωδικό πρόσβασης στην εφαρμογή μπορεί να επιλέξει «Ξεχάσατε τον κωδικό πρόσβασης?». Πατώντας την επιλογή αυτή μια φόρμα εμφανίζεται όπου ο χρήστης πρέπει να τοποθετήσει το όνομα χρήστη που διαθέτει (δηλαδή το e-mail του).

Συμπληρώστε τα παρακάτω στοιχεία για επανάκτηση του κωδικού πρόσβασης

Όνομα χρήστη:

Εικόνα 5.4: Το πρώτο βήμα για την ανάκτηση κωδικού πρόσβασης

Αν η διεύθυνση e-mail δεν υπάρχει καταχωρημένη στο σύστημα τότε εμφανίζεται το μήνυμα «Ο χρήστης αυτός δεν είναι εγγεγραμμένος στην εφαρμογή».

Συμπληρώστε τα παρακάτω στοιχεία για επανάκτηση του κωδικού πρόσβασης

Ο χρήστης αυτός δεν είναι εγγεγραμμένος στην εφαρμογή

Όνομα χρήστη:

Εικόνα 5.5: Μήνυμα λάθους σε περίπτωση μη ύπαρξης λογαριασμού

Σε περίπτωση όμως που η διεύθυνση e-mail βρίσκεται στη βάση δεδομένων τότε το επόμενο βήμα για την ανάκτηση του κωδικού είναι η απάντηση σε μια ερώτηση. Όταν ο χρήστης εγγράφεται στο σύστημα για πρώτη φορά τότε αυτό του ζητά να απαντήσει σε μια εκ των:

1. Πατρικό όνομα μητέρας
2. Το όνομα του πρώτου κατοικίδιου
3. Αγαπημένη ταινία
4. Τι αγαπάς περισσότερο
5. Ποιός είναι ο αγαπημένος σου καθηγητής

Η αρχική απάντηση του χρήστη έχει αποθηκευτεί στη βάση δεδομένων και το μόνο που πρέπει να κάνει ο χρήστης για να επαληθεύσει την ταυτότητά του είναι να δώσει τη σωστή απάντηση.

Συμπληρώστε τα παρακάτω στοιχεία για επανάκτηση του κωδικού πρόσβασης

Πατρικό όνομα μητέρας <input type="text" value="Νικολάου"/>	
<input type="button" value="Submit"/>	<input type="button" value="Reset"/>

Εικόνα 5.6: Το δεύτερο βήμα για την ανάκτηση κωδικού πρόσβασης (προσωπική ερώτηση).

Αν η απάντηση που δώσει ο χρήστης είναι σύμφωνη με την αρχική του απάντηση, τότε του αποστέλλεται στο ηλεκτρονικό ταχυδρομείο ο κωδικός πρόσβασης. Αν όμως η απάντηση είναι λάθος τότε κατάλληλο μήνυμα εμφανίζεται.

Ο κωδικός πρόσβασης έχει αποσταλλεί με το ηλεκτρονικό ταχυδρομείο.

Εικόνα 5.7: Επιβεβαίωση αποστολής e-mail.

Η δημιουργία νέου λογαριασμού γίνεται μέσα από την επιλογή «Εγγραφή νέου χρήστη». Ο χρήστης θα πρέπει να τοποθετήσει ορισμένα στοιχεία του, όπως διεύθυνση e-mail, κωδικό πρόσβασης, να επιλέξει την προσωπική ερώτηση και να δώσει την προσωπική του απάντηση. Επίσης μπορεί να επιλέξει την ομάδα στην οποία θα ανήκει, ανάλογα με τα ενδιαφέροντά του. Ο κωδικός πρόσβασης πρέπει να είναι από 6 μέχρι 7 χαρακτήρες. Μόλις ο χρήστης ολοκληρώσει την διαδικασία εγγραφής του με επιτυχία, τότε θα αποσταλεί στο email που έδωσε σαν όνομα χρήστη ένα μνημα που θα τον καλωσορίζει στην εφαρμογή. Η εφαρμογή

έχει την δυνατότητα να ελέγχει αν όντως ο χρήστης κατά την εγγραφή του έδωσε διεύθυνση email ή όχι.



[Αρχική](#) [Σχόλια](#) [Σχετικά με την εφαρμογή](#) [Επικοινωνία](#)

Δημιουργήστε λογαριασμό στην εφαρμογή E-Library. Το όνομα χρήστη πρέπει να είναι η πραγματική σας διεύθυνση e-μails.

Όνομα χρήστη(Σημείωση:Το πραγματικό σας email):	<input type="text" value="john@john.com"/>
Κωδικός πρόσβασης(Σημείωση:Πρέπει να είναι από 6-7 χαρακτήρες):	<input type="password" value="••••••"/>
Επανάληψη κωδικού πρόσβασης:	<input type="password" value="••••••"/>
Επιλέξτε μια ερώτηση ασφαλείας:	<input type="text" value="Πατρικό όνομα μητέρας"/>
Δώστε την απάντηση:	<input type="text" value="maria"/>
Επιλέξτε τον κύριο κλάδο ενδιαφέροντος:	<input type="text" value="Ομάδα Τηλεπικοινωνιών"/>
<input type="button" value="Δημιουργία λογαριασμού"/> <input type="button" value="Reset"/>	

© Copyright 2009 - Ιωάννης Γκουτζουρέλας και Δημήτρης Σωτηρίου

Εικόνα 5.8: Ένας νέος χρήστης δημιουργεί λογαριασμό στην εφαρμογή.

Ο λογαριασμός σας δημιουργήθηκε. Μπορείτε να χρησιμοποιήσετε την εφαρμογή.


Εικόνα 5.9: Μήνυμα επιτυχούς δημιουργίας λογαριασμού.

Αφού λοιπόν έχει δημιουργηθεί ο νέος χρήστης, στην συνέχεια θα παρουσιαστούν οι δυνατότητες που παρέχονται στους χρήστες του συστήματος από την εφαρμογή. Οι επιλογές του χρήστη βρίσκονται πάντα στην αριστερή πλευρά της σελίδας. Οι επιλογές αυτές είναι οι

«Εξέταση κατηγορίας», «Τα αρχεία μου», «Ανέβασε αρχείο», «Αναζήτηση» και «Αποσύνδεση».

Πλοήγηση Εφαρμογής
Εξέταση κατηγορίας
Τα αρχεία μου
Ανέβασε αρχείο
Αναζήτηση
Αποσύνδεση

Εικόνα 5.10: Οι επιλογές των χρηστών.



Archive Σχόλια Σχετικά με την εφαρμογή Σύνδεσμοι Επικοινωνία

Πλοήγηση Εφαρμογής	Καλώς ήλθετε goutz2000@yahoo.com
Εξέταση κατηγορίας	Χρησιμοποιήστε τις επιλογές αριστερά για να πλοηγηθείτε στην εφαρμογή.
Τα αρχεία μου	
Ανέβασε αρχείο	
Αναζήτηση	
Αποσύνδεση	

© Copyleft - Ιωάννης Γκουτζουρέλας και Δημήτρης Σωτηρίου

Εικόνα 5.11: Η κεντρική σελίδα του χρήστη goutz2000 μετά την είσοδό του.

Στην εξέταση κατηγορίας, ο χρήστης μπορεί να δει τις υπάρχουσες κατηγορίες και υποκατηγορίες στις οποίες μπορεί να εντάξει τα έγγραφα που ανεβάζει στο σύστημα. Αρχικά βλέπει τις κατηγορίες γονέας και στη συνέχεια επιλέγοντας μία από αυτές μπορεί να δει τις περιεχόμενες υποκατηγορίες.



E-Lib

[Αρχική](#) [Σχόλια](#) [Σχετικά με την εφαρμογή](#) [Σύνδεσμοι](#) [Επικοινωνία](#)

**Πλοήγηση
Εφαρμογής**

**Εξέταση
κατηγορίας**

Τα αρχεία μου

Ανέβασε αρχείο

Αναζήτηση

Αποσύνδεση

Παρακάτω ακολουθούν οι υπάρχουσες κατηγορίες και υποκατηγορίες:
Σημείωση: Οι κύριες κατηγορίες (Top Level) δεν έχουν άλλες κατηγορίες ως γονείς

Πατήστε πάνω στο όνομα μιας κατηγορίας για να την επεξεργαστείτε.

Όνομα κατηγορίας	Κατηγορία γονέας
Προπτυχιακές εργασίες	Top Level
Μεταπτυχιακές εργασίες	Top Level
Δικατορικές εργασίες	Top Level
Ερευνητικές εργασίες	Top Level
Διαλέξεις	Top Level

© Copyleft - Ιωάννης Γκουτζουρέλας και Δημήτρης Σωτηρίου

Εικόνα 5.12. Εξέταση κατηγορίας. Παρουσιάζονται οι κατηγορίες που μπορούν να αρχειοθετηθούν τα έγγραφα στο σύστημα.

Αν για παράδειγμα επιλέξει την κατηγορία Προπτυχιακές εργασίες τότε θα του εμφανιστεί η παρακάτω εικόνα



E-Lib

[Αρχική](#) [Σχόλια](#) [Σχετικά με την εφαρμογή](#) [Σύνδεσμοι](#) [Επικοινωνία](#)

Πλοήγηση Εφαρμογής
Εξέταση κατηγορίας
Τα αρχεία μου
Ανέβασε αρχείο
Αναζήτηση
Αποσύνδεση

Παρακάτω ακολουθούν οι υπάρχουσες κατηγορίες και υποκατηγορίες:
Σημείωση: Οι κύριες κατηγορίες (Top Level) δεν έχουν άλλες κατηγορίες ως γονείς

Πατήστε πάνω στο όνομα μιας κατηγορίας για να την επεξεργαστείτε.

Όνομα κατηγορίας	Κατηγορία γονέας
---> ΠΕ-Αρχιτεκτονικής Υπολογιστικών Συστημάτων	Προπτυχιακές εργασίες
---> ΠΕ-Υποδομής Υπολογιστικών Συστημάτων	Προπτυχιακές εργασίες
---> ΠΕ-Προγραμματισμού	Προπτυχιακές εργασίες
---> ΠΕ-Τεχνολογίας Λογισμικού	Προπτυχιακές εργασίες
---> ΠΕ-Δικτύων	Προπτυχιακές εργασίες
---> ΠΕ-Τηλεπικοινωνιών	Προπτυχιακές εργασίες
---> ΠΕ-Διοίκ. Οικ. Νομ. και Ανθρωπ. σπουδών (ΔΟΝΑ)	Προπτυχιακές εργασίες

© Coryleft - Ιωάννης Γκουτζουρέλας και Δημήτρης Σωτηρίου

Εικόνα 5.13 Υποκατηγορίες της κατηγορίας Προπτυχιακές εργασίες

Στη συνέχεια μπορεί να πιάσει την προτιμώμενη υποκατηγορία και θα εμφανιστεί ένας κατάλογος με τα έγγραφα τα οποία ανήκουν στη συγκεκριμένη υποκατηγορία και ο χρήστης έχει την εξουσιοδότηση να εξετάσει..



E-Lib

[Αρχική](#) [Σχόλια](#) [Σχετικά με την εφαρμογή](#) [Σύνδεσμοι](#) [Επικοινωνία](#)

Πλοήγηση Εφαρμογής
Εξέταση κατηγορίας
Τα αρχεία μου
Ανέβασε αρχείο
Αναζήτηση
Αποσύνδεση

Εδώ βρίσκονται τα αρχεία που έχετε ανεβάσει στην εφαρμογή E-Lib :

Τίτλος	Κατηγορία	Επιλογές
Load Balancing Servers, Firewalls, and Caches	ΠΕ-Δικτύων	Προβολή Επεξεργασία Προβολή Στοιχείων

© Coryleft - Ιωάννης Γκουτζουρέλας και Δημήτρης Σωτηρίου

Εικόνα 5.14 Αρχεία της υποκατηγορίας ΠΕ-Δίκτυα

Στη παραπάνω εικόνα φαίνεται ότι ο χρήστης επέλεξε να δει τα διαθέσιμα έγγραφα στην υποκατηγορία δίκτυα των προπτυχιακών εργασιών

Η επόμενη επιλογή του χρήστη είναι να ανεβάσει κάποιο αρχείο (π.χ. ένα paper που έχει ετοιμάσει) στην εφαρμογή. Τα στοιχεία που πρέπει να συμπληρώσει ο χρήστης είναι τα παρακάτω:

- **Τίτλος.** Ο τίτλος που θα έχει το αρχείο στο σύστημα
- **Προβολή.** Υπάρχει δυνατότητα ορισμού ορατότητας. Δηλαδή ποιοι άλλοι χρήστες της εφαρμογής θα έχουν πρόσβαση στο αρχείο που ανεβαίνει στο σύστημα; Θα έχει πρόσβαση μόνο ο ιδιοκτήτης του (κάτι χρήσιμο όταν χρησιμοποιεί το σύστημα για να αποθηκεύει draft εκδόσεις της μελέτης); Θα έχουν πρόσβαση μόνο μέλη που βρίσκονται στην ίδια ομάδα εργασίας με τον ιδιοκτήτη του αρχείου (χρήσιμο όταν μια ομάδα εργασίες συμμετέχει συλλογικά στην συγκεκριμένη έρευνα) ή θα έχουν πρόσβαση όλοι οι χρήστες της εφαρμογής;
- **Κατηγορία.** Σε ποια από τις προαναφερθείσες κατηγορίες θα προστεθεί το έγγραφο.
- **Συγγραφείς.** Το όνομα ή τα ονόματα των συγγραφέων του άρθρου.
- **Σελίδες.** Ο αριθμός των σελίδων του άρθρου.
- **Εκδότης.** Τα στοιχεία του εκδότη αν πρόκειται για δημοσιευμένο κείμενο (σε ερευνητικό περιοδικό ή βιβλίο).
- **Έτος έκδοσης.** Το έτος έκδοσης του άρθρου.
- **Έτος/Μήνας/Ημέρα:** Η ακριβής ημερομηνία τελευταίας αναθεώρησης.
- **Tags:** Keywords που θα χρησιμοποιηθούν ώστε το συγκεκριμένο άρθρο να επιστρέφει σε αντίστοιχες αναζητήσεις χρηστών του συστήματος.
- **Σχόλια.** Εδώ τοποθετούνται τυχόν σχόλια σχετικά με το άρθρο.
- **Περιγραφή.** Μια σύντομη περιγραφή του άρθρου.
- **Αρχείο.** Το επισυναπτόμενο αρχείο που θα ανέβει στον εξυπηρετητή και πρέπει να είναι είτε με μορφή PDF ή σε DOC.

Πλοήγηση Εφαρμογής	Εξέταση κατηγορίας
Τα αρχεία μου	
Ανέβασε αρχείο	
Αναζήτηση	
Αποσύνδεση	

Ανέβασε αρχείο στο E-Lib !

Τίτλος:

Προβολή: ▼

Κατηγορία: ▼

Συγγραφείς:

Σελίδες:

Εκδότης:

Έτος έκδοσης Έτος/Μήνας/Ημέρα: ▼ ▼ ▼

Tags:

Σχόλια:

Περιγραφή:

Αρχείο:

Εικόνα 5.15 Φόρμα τοποθέτησης νέου αρχείου.

Προβολή:	<input type="text" value="Μόνο σε εμένα"/> ▼
Κατηγορία:	<input type="text" value="Μόνο σε εμένα"/> ▼
Συγγραφείς:	<input type="text" value="Μόνο σε ομάδες που ανήκω"/> ▼
	<input type="text" value="Σε όλους"/> ▼

Εικόνα 5.16. Οι δυνατότητες επιλογής σχετικά με την ορατότητα του αρχείου.

Κατηγορία:	ΠΕ-Αρχιτεκτονικής Υπολογισ
Συγγραφείς:	Ερευνητικές εργασίες
Σελίδες:	Διαλέξεις
Εκδότης:	ΠΕ-Αρχιτεκτονικής Υπολογιστικών Συστημάτων
Έτος έκδοσης Έτος/Μήνας/Ημέρα:	ΠΕ-Υποδομής Υπολογιστικών Συστημάτων
Tags:	ΠΕ-Προγραμματισμού
Σχόλια:	ΠΕ-Τεχνολογίας Λογισμικού
Περιγραφή:	ΠΕ-Δικτύων
	ΠΕ-Τηλεπικοινωνιών
	ΠΕ-Διοίκ. Οικ. Νομ. και Ανθρωπ. σπουδών (ΔΟΝΑ)
	ΜΕ-Αρχιτεκτονικής Υπολογιστικών Συστημάτων
	ΜΕ-Υποδομής Υπολογιστικών Συστημάτων
	ΜΕ-Προγραμματισμού
	ΜΕ-Τεχνολογίας Λογισμικού
	ΜΕ-Δικτύων
	ΜΕ-Τηλεπικοινωνιών
	ΜΕ-Διοίκ. Οικ. Νομ. και Ανθρωπ. σπουδών (ΔΟΝΑ)
	ΔΕ-Αρχιτεκτονικής Υπολογιστικών Συστημάτων
	ΔΕ-Υποδομής Υπολογιστικών Συστημάτων
	ΔΕ-Προγραμματισμού
	ΔΕ-Τεχνολογίας Λογισμικού

Εικόνα 5.17 Οι επιλογές του χρήστη σχετικά με την κατηγορία στην οποία θα εντάξει το αρχείο έρχονται από τη βάση δεδομένων.

Ανέβασε αρχείο στο E-Lib !

Τίτλος:	<input type="text" value="Create dynamic sites with PHP"/>
Προβολή:	<input type="text" value="Μόνο σε εμένα"/>
Κατηγορία:	<input type="text" value="ΠΕ-Προγραμματισμού"/>
Συγγραφείς:	<input type="text" value="Md. Ashraful Anam"/>
Σελίδες:	<input type="text" value="21"/>
Εκδότης:	<input type="text" value="developerWorks"/>
Έτος έκδοσης Έτος/Μήνας/Ημέρα:	<input type="text" value="2005"/> <input type="text" value="1"/> <input type="text" value="1"/>
Tags:	<input type="text" value="PHP SQL"/>
Σχόλια:	<input type="text" value="Create dynamic sites with PHP & MySQL"/>
Περιγραφή:	<input type="text" value="Create dynamic sites with PHP & MySQL"/>
Αρχείο:	<input type="text" value="\\iris\E-BOOKS\le-book"/> <input type="button" value="Browse..."/>
<input type="button" value="Ανέβασε το αρχείο"/>	

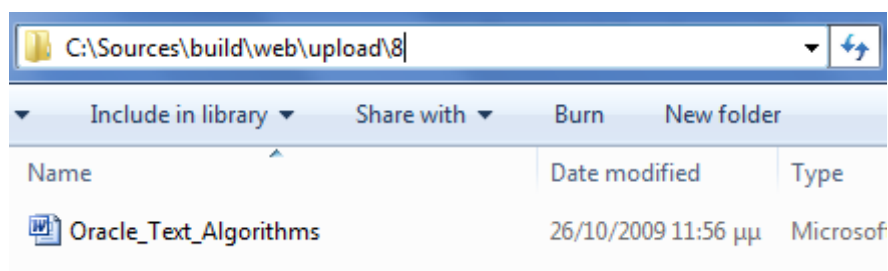
Εικόνα 5.18 Ο χρήστης ανεβάζει το πρώτο του αρχείο.

Document has been uploaded successfully

Εικόνα 5.19 Η εφαρμογή εμφανίζει μήνυμα ότι το αρχείο ανέβηκε με επιτυχία στον εξυπηρετητή.

Το παραπάνω αρχείο τοποθετείται σε φάκελο *build/web/upload/user_id/ονομα_αρχείου*. Στη συγκεκριμένη περίπτωση επειδή το *user_id* του συγκεκριμένου χρήστη στον πίνακα *users_tbl* είναι το **3** και η εφαρμογή εκτελείται από το φάκελο *C:\Sources\build\web\upload* το αρχείο

τοποθετείται σε νέο φάκελο που δημιουργείται αυτομάτως από την εφαρμογή για να συγκρατεί όλα τα αρχεία του συγκεκριμένου χρήστη.



Εικόνα 5.19. Το αρχείο τοποθετείται στον κατάλληλο φάκελο.

Στην επιλογή τα αρχεία μου, ο χρήστης μπορεί να δει τα αρχεία που ήδη έχει ανεβάσει στην εφαρμογή. Ο χρήστης goutz2000 έχει ανεβάσει δύο αρχεία και για αυτό ο παρακάτω πίνακας παρουσιάζει τα δυο αυτά αρχεία.



[Αρχική](#) [Σχόλια](#) [Σχετικά με την εφαρμογή](#) [Σύνδεσμοι](#) [Επικοινωνία](#)

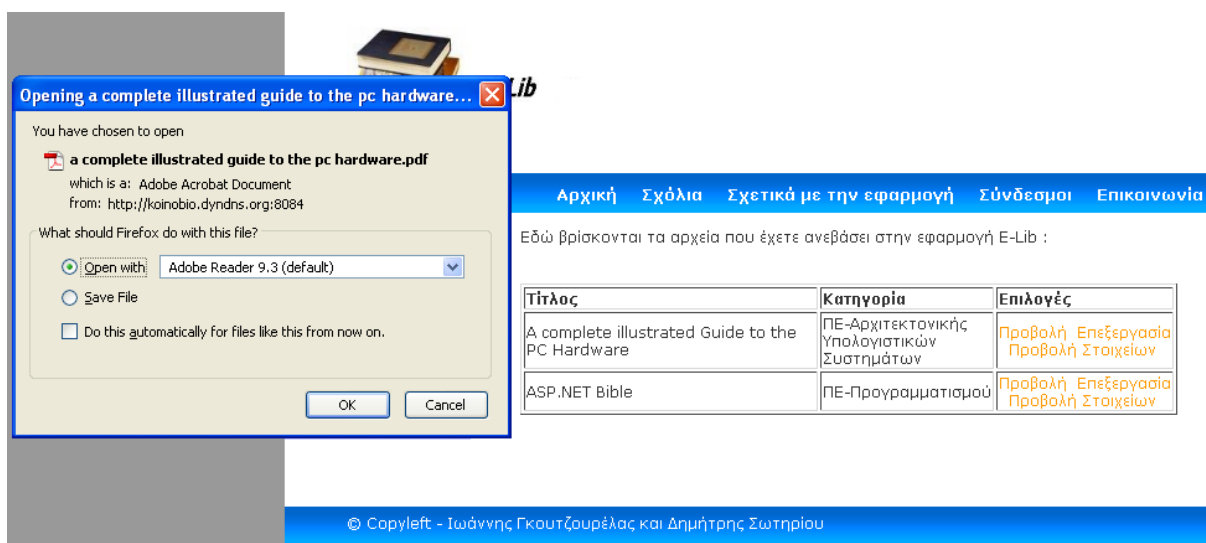
Πλοήγηση Εφαρμογής	Εδώ βρίσκονται τα αρχεία που έχετε ανεβάσει στην εφαρμογή E-Lib :
Εξέταση κατηγορίας	
Τα αρχεία μου	
Ανέβασε αρχείο	
Αναζήτηση	
Αποσύνδεση	

Τίτλος	Κατηγορία	Επιλογές
A complete illustrated Guide to the PC Hardware	ΠΕ-Αρχιτεκτονικής Υπολογιστικών Συστημάτων	Προβολή Επεξεργασία Προβολή Στοιχείων
ASP.NET Bible	ΠΕ-Προγραμματισμού	Προβολή Επεξεργασία Προβολή Στοιχείων

© Copyleft - Ιωάννης Γκουτζουρέλας και Δημήτρης Σωτηρίου

Εικόνα 5.20 Εμφάνιση αρχείων που ο χρήστης έχει ανεβάσει στην εφαρμογή

Πατώντας «Προβολή» ένα νέο παράθυρο εμφανίζεται που επιτρέπει στο χρήστη να κατεβάσει το αρχείο στον τοπικό του δίσκο.



Εικόνα 5.21. Η επιλογή «Προβολή» αυτομάτως δίνει τη δυνατότητα στο χρήστη να κατεβάσει το αρχείο αυτό.

Επίσης πρέπει να επισημανθεί ότι για να μη χρησιμοποιηθεί η εφαρμογή E-Lib για διακίνηση μουσικής, ταινιών ή γενικότερα άλλου τύπου αρχείων, μόνο αρχεία PDF ή DOC επιτρέπεται να ανεβούν στον εξυπηρετητή. Σε περίπτωση που κάποιος χρήστης προσπαθήσει να ανεβάσει άλλου τύπου αρχεία τότε κατάλληλο μήνυμα λάθους εμφανίζεται.

Ανέβασε αρχείο στο E-Lib !

Μόνο PDF και DOC αρχεία επιτρέπονται.

Εικόνα 5.22. Εμφάνιση μηνύματος λάθους όταν άλλος τύπος αρχείων ανεβαίνει στον εξυπηρετητή.

Στη συνέχεια θα παρουσιάσουμε την αναζήτηση. Ο χρήστης επιλέγοντας «Αναζήτηση αρχείου» μεταφέρεται στη σελίδα όπου μπορεί να τοποθετήσει κάποια λέξη βάση της οποίας θα γίνει η αναζήτηση στα αρχεία που διαθέτει η εφαρμογή.

Η αναζήτηση μπορεί να σας επιστρέψει αρχεία τύπου PDF ή MS Word format.

Τοποθετήστε τον τίτλο ή κάποιο key-word σχετικά με το είδος των εγγράφων που αναζητείτε:

Εικόνα 5.23. Η φόρμα αναζήτησης αρχείων.

Τοποθετώντας την λέξη που μας ενδιαφέρει στο παραπάνω πεδίο και πατώντας «Αναζήτηση» ένα σύνθετο SQL Query ψάχνει ανάμεσα στα αρχεία που έχουν τοποθετηθεί στην εφαρμογή ώστε να επιτρέψει μόνο εκείνα τα οποία σχετίζονται με τη λέξη αυτή.

Ο τίτλος, η κατηγορία στην οποία ανήκει, η ορατότητα, ο ιδιοκτήτης, η ημερομηνία που ανέβηκε στον εξυπηρετητή για κάθε αρχείο που ανακτάται εμφανίζονται στο χρήστη. Στο πεδίο ορατότητα οι τιμές που μπορεί να δει είναι:

- A** (All Users) Ορατό σε όλους τους χρήστες
- G** (Group) Ορατό μόνο σε χρήστες της ίδιας ομάδας
- O** (Owner) Ορατό μόνο στον ιδιοκτήτη

Για να παρουσιάσουμε τις δυνατότητες της εφαρμογής θα δημιουργηθούν τρεις χρήστες goutz2000@yahoo.com και dimsot@gmail.com που θα ανήκουν στην ίδια ομάδα και ένας τρίτος sotos82@yahoo.com ο οποίος θα ανηκει σε διαφορετική ομάδα Έχουμε λοιπόν τον χρήστη sotos82 ο οποίος όπως βλέπουμε και παρακάτω στην επισκόπηση του πίνακα των χρηστών από το πρόγραμμα MySQL-Front έχει group_id 4 και είναι ο χρήστης με id_users 6 και

idusers_tbl	userid	passwd	group_id	secret_question_id	secret_answer	register_date
1	admin	admin123	1	1	admin	2008-12-18 16:22:45
2	dimsot@gmail.com	dimsot123	2	2	karolos	2010-06-20 23:05:06
3	goutz2000@yahoo.com	goutz2000	2	2	kaiser	2010-06-20 23:05:17
4	swmaras@teilar.gr	swmaras	5	3	yes	2010-06-20 23:05:16
5	dsotiriou@smalls.gr	dsotiriou	3	2	karolos	2010-06-20 23:05:16
6	sotos82@yahoo.com	sotos82	4	2	karolos	2010-06-20 23:05:16

Εικόνα 5.23. Ο πίνακας των χρηστών της βάσης δεδομένων

έχει ανεβάσει τρία έγγραφα όπως φαίνεται στον πίνακα παρακάτω και πιο συγκεκριμένα τα έγγραφα με iddocs 4, 5 και 6. Το καθένα από αυτά όμως έχει διαφορετική ορατότητα. Το αρχείο “Load balancing servers” είναι ορατό από όλους αφού έχει privilege_lvl_id ίσο με A (ALL), το Wi-fi toys είναι ορατό μόνο από την ομάδα του καθώς έχει privilege_lvl_id G (GROUP) και το “Internet & computers terms” είναι ορατό μόνο από τον ίδιο με privilege_lvl_id ίσο με O (Owner ή αλλιώς private)..

iddocs_tbl	title	category_id	privilege_lvl_id	uploaded_by
1	Create dynamic sites with PHP & MySQL	8	O	2
2	MySQL/PHP Database Applications	8	G	2
3	MySQL	8	A	2
4	Load Balancing Servers, Firewalls, and Caches	10	A	6
5	Wi-Fi Toys 15 Cool Wireless Projects for Home, Office	10	G	6
6	INTERNET & COMPUTER TERMS	10	O	6
7	A complete illustrated Guide to the PC Hardware	6	G	3
8	ASP.NET Bible	8	A	3

Εικόνα 5.24. Τρία αρχεία με διαφορετικό περιεχόμενο και διαφορετική ορατότητα από τον ίδιο χρήστη.

Αφού δημιουργήσαμε τους παραπάνω χρήστες θα παρουσιάσουμε την αναζήτηση με τα ίδια κριτήρια που αυτή τη φορά όμως θα επιστρέφουν μόνο τα αρχεία στα οποία έχουν ορατότητα.

Έτσι όταν ο χρήστης goutz2000@yahoo.com κάνει αναζήτηση με τη λέξη mysql θα του εμφανίζονται τα αρχεία MySql και MySQL/PHP database Applications αλλά όχι και το αρχείο Create dynamic sites with PHP & MySql. Αυτό οφείλεται στο ότι το πρώτο άρχείο το βλέπει γιατί είναι ορατό από όλους ενώ το δεύτερο το βλέπει γιατί είναι στην ίδια κατηγορία με τον χρήστη που το ανέβασε, στην συγκεκριμένη περίπτωση ο dimsot@gmail.com.

Η αναζήτηση μπορεί να σας επιστρέψει αρχεία τύπου PDF ή MS Word format.

Τοποθετήστε τον τίτλο ή κάποιο key-word σχετικά με το είδος των εγγράφων που αναζητείτε:

Τα παρακάτω αρχεία ανακτήθηκαν κατά την αναζήτηση:

Συνολικά αποτελέσματα:2

Τίτλος	Κατηγορία	Ορατότητα	Ιδιοκτήτης	Ανέβηκε στις	Ενέργειες
MySQL	ΠΕ-Προγραμματισμού	A	dimsot@gmail.com	2010-06-24 01:21:50.0	Προβολή Επεξεργασία Προβολή Στοιχείων
MySQL/PHP Database Applications	ΠΕ-Προγραμματισμού	G	dimsot@gmail.com	2010-06-24 01:13:58.0	Προβολή Επεξεργασία Προβολή Στοιχείων

Εικόνα 5.25. Η φόρμα ανάκτησης αρχείων από την αναζήτηση.

Αντιθέτως ο χρήστης sotos82@yahoo.com εάν κάνει την ίδια αναζήτηση θα του εμφανιστεί μόνο ,οπώς φαίνεται παρακατω, το αρχείο MySql καθώς καθώς δεν ανήκει στην ίδια ομάδα με τον ιδιοκτήτη του αρχείου. Το συγκεκριμένο αρχείο θα εμφανίζεται σε οποιοδήποτε χρήστη εξάλλου καθώς έτσι έχει οριστεί από το χρήστη που το ανεβασε.

Τα παρακάτω αρχεία ανακτήθηκαν κατά την αναζήτηση:

Συνολικά αποτελέσματα:1

Τίτλος	Κατηγορία	Ορατότητα	Ιδιοκτήτης	Ανέβηκε στις	Ενέργειες
MySQL	ΠΕ-Προγραμματισμού	A	dimsot@gmail.com	2010-06-24 01:21:50.0	Προβολή Επεξεργασία Προβολή Στοιχείων

Εικόνα 5.26. Η φόρμα ανάκτησης αρχείων από την αναζήτηση.

Το σύνολο και των τριών αρχείων μπορεί να το δει μόνο dimsot@gmail.com ο οποίος και τα ανέβασε στην εφαρμογή.

Η αναζήτηση μπορεί να σας επιστρέψει αρχεία τύπου PDF ή MS Word format.

Τοποθετήστε τον τίτλο ή κάποιο key-word σχετικά με το είδος των εγγράφων που αναζητείτε:

Τα παρακάτω αρχεία ανακτήθηκαν κατά την αναζήτηση:

Συνολικά αποτελέσματα:3

Τίτλος	Κατηγορία	Ορατότητα	Ιδιοκτήτης	Ανέβηκε στις	Ενέργειες
Create dynamic sites with PHP & MySQL	ΠΕ- Προγραμματισμού	O	dimsot@gmail.com	2010-06-24 01:11:02.0	Προβολή Επεξεργασία Προβολή Στοιχείων
MySQL/PHP Database Applications	ΠΕ- Προγραμματισμού	G	dimsot@gmail.com	2010-06-24 01:13:58.0	Προβολή Επεξεργασία Προβολή Στοιχείων
MySQL	ΠΕ- Προγραμματισμού	A	dimsot@gmail.com	2010-06-24 01:21:50.0	Προβολή Επεξεργασία Προβολή Στοιχείων

Εικόνα 5.27. Η φόρμα ανάκτησης αρχείων από την αναζήτηση.

Έτσι βλέπουμε ότι η εφαρμογή προσφέρει την διακριτικότητα και τον διαχωρισμό μεταξύ των χρηστών της.

Πιο έντονος γίνεται αυτός ο διαχωρισμός και αντιληπτός από την πρώτη στιγμή εισαγωγής στην εφαρμογή όταν εισέλθει ο διαχειριστής. Στο αριστερό menu της εμφανίζεται ακόμα μια επιλογή οπου δεν έχει ένας απλός χρήστης. Η επιλογή αυτή είναι η δημιουργία / προβολή ομάδας Έτσι ο μπορεί επιπλέον να δημιουργήσει μία ομάδα στην οποία θεωρεί ότι μπορεί να ενταχθεί ένας αριθμός χρηστών με κοινά ενδιαφέροντα ή ιδιότητες.

Πλοήγηση Εφαρμογής
Εξέταση κατηγορίας
Δημιουργία/Προβολή ομάδας
Τα αρχεία μου
Ανέβασε αρχείο
Αναζήτηση
Αποσύνδεση

Εικόνα 5.28 Κεντρικό menu του διαχειριστή

Δημιουργία/Προβολή ομάδας

Ομάδες:

Δημιουργία ομάδας:

Εικόνα 5.29 Δημιουργία / προβολή ομάδας

Στη συνέχεια παρατηρείται ότι η διαφοροποίηση συνεχίζει αφού πατώντας στην εξέταση κατηγορίας εμφανίζεται μια επιλογή οπου μπορεί να δημιουργήσει μια καινούργια κατηγορία ή υποκατηγορία.

Πατήστε εδώ για δημιουργία νέα (υπο)κατηγορίας

Πατήστε πάνω στο όνομα μιας κατηγορίας για να την επεξεργαστείτε.

Εικόνα 5.30 Επιλογές του διαχειριστή

Επιλέγοντας το μεταφέρεται στην παρακάτω τοποθεσία όπως φαίνεται παρακάτω οπου δίνεται η επιλογή της δημιουργίας.

Όνομασία κατηγορίας:

Κάντην κεντρική κατηγορία:

Επέλεξε κεντρική κατηγορία:

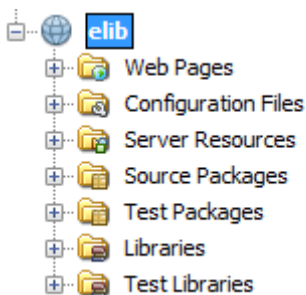
Δημιουργία κατηγορίας

Εικόνα 5.31 Επιλογές του διαχειριστή για την δημιουργία κατηγορίας

Αυτή ήταν μία συνοπτική παρουσίαση της εφαρμογής που δημιουργήθηκε στα πλαίσια αυτής της πτυχιακής εργασίας. Θα θέλαμε να αναφέρουμε ότι πέραν των λειτουργιών που παρουσιάστηκαν η εφαρμογή προβάλλει στο εκάστοτε tag δυναμικά τον τίτλο κάθε σελίδας και παρέχει και άλλες δυναμικές λειτουργίες οι οποίες γίνονται διακριτές μόνο με την χρήση αυτής. Τέλος για περαιτέρω εξέταση της εργασίας μπορεί να πλοηγηθεί στην διεύθυνση www.e-lib.gr όπου και βρίσκεται ανεβασμένη η εφαρμογή για ερευνητικούς σκοπούς.

ΚΕΦΑΛΑΙΟ 6. Επεξήγηση υλοποίηση

Ο κώδικας της εφαρμογής μπορεί να κατηγοριοποιηθεί ως εξής:



Στην εικόνα αριστερά φαίνονται οι φάκελοι στους οποίους έχουν τοποθετηθεί τα αρχεία που απαρτίζουν την εφαρμογή. Στο φάκελο **Web Pages** βρίσκεται ο **JSP** κώδικας. Στο φάκελο **Configuration Files** βρίσκονται αρχεία που είναι απαραίτητα από τον Application Server (Tomcat) αλλά και από το Struts Framework. Στο φάκελο **Source Packages** βρίσκονται τα αρχεία Java που

υλοποιούν το business logic της εφαρμογής.

Ως **Java Servlet κώδικας**, συνολικά 23 αρχεία Java υλοποιήθηκαν. Τα αρχεία αυτά μπορούν να χωριστούν σε 3 κατηγορίες αρχείων, τα Action, τα Bean και τα Form:

Αρχεία Action	Αρχεία Bean	Αρχεία Form
CatAction.java	CatBean.java	CatForm.java
DocAction.java	DocBean.java	DocForm.java
ForgotAction.java	ForgotBean.java	ForgotForm.java
GroupAction.java	GroupBean.java	GroupForm.java
LoginAction.java	LoginBean.java	LoginForm.java
RegisterAction.java	RegisterBean.java	RegisterForm.java
SearchAction.java	SearchBean.java	SearchForm.java
AdminAction.java		
LogoutAction.java		

Οι **σελίδες JSP** που δημιουργήθηκαν είναι οι παρακάτω:

Όνομα αρχείου	Σύντομη περιγραφή
UploadDoc.jsp	Το αρχείο αυτό περιέχει τη φόρμα που συμπληρώνει ο χρήστης της εφαρμογής όταν θέλει να ανεβάσει κάποιο αρχείο. (εικόνα 17).

admin.jsp	Αυτή η σελίδα επιτρέπει την είσοδο μόνο στο διαχειριστή της εφαρμογής με χρήση του κατάλληλου κωδικού πρόσβασης και ονόματος χρήστη (διαχειριστή).
category.jsp	Η σελίδα αυτή παρουσιάζει τις κατηγορίες και υποκατηγορίες που υπάρχουν στο σύστημα και επιτρέπει την διαχείρισή τους (εικόνα 12).
catinfo.jsp	Το catinfo επιτρέπει την επεξεργασία κατηγοριών του συστήματος.
subcatinfo.jsp	Το subcatinfo επιτρέπει την επεξεργασία υποκατηγοριών του συστήματος.
dashboard.jsp	Περιέχει το χαιρετιστήριο μήνυμα που εμφανίζεται στους χρήστες όταν αυτοί κάνουν με επιτυχία είσοδο στην εφαρμογή
docs.jsp	Η σελίδα αυτή εμφανίζει στον χρήστη τα αρχεία που έχει ανεβάσει στην εφαρμογή και του επιτρέπει να τα κατεβάσει ή να τα επεξεργαστεί.
editdoc.jsp	Η σελίδα αυτή επιτρέπει την επεξεργασία των στοιχείων των σχετικών με ένα αρχείο που έχει ανέβει στο σύστημα.
footer.jsp	Το footer είναι το τελευταίο μέρος που τοποθετείται στο κάτω μέρος κάθε σελίδας και δείχνει το χαιρετιστήριο μήνυμα «Copyright 2009...».
forgot.jsp	Όταν ο χρήστης έχει ξεχάσει τον κωδικό πρόσβασης στην εφαρμογή μπορεί να τον ανακτήσει μέσα από τη σελίδα αυτή.
groups.jsp	Η σελίδα αυτή αφορά τη δημιουργία και τη διαχείριση ομάδων χρηστών.
header.inc	Αυτή η σελίδα τοποθετείται στο πάνω μέρος κάθε άλλης σελίδας της εφαρμογής και περιέχει τις επιλογές των χρηστών που εμφανίζονται στο πάνω μέρος.
index.jsp	Αυτή είναι η κεντρική σελίδα. Όταν ο χρήστης πρώτο-μπαίνει, ζητά κωδικό πρόσβασης. Αν ο χρήστης αυθεντικοποιηθεί, τότε τους εμφανίζει το dashboard.
logout.jsp	Με αυτή τη σελίδα ο χρήστης κάνει sign-out.
menu.jsp	Αυτό το αρχείο περιέχει το μενού επιλογών. Αναλόγως αν ο χρήστης είναι απλός χρήστης ή διαχειριστής του εμφανίζει διαφορετικές επιλογές.
navigation.jsp	Περιέχει το κεντρικό navigation με επιλογές «Αρχική», «Σχόλια», «Σχετικά με την εφαρμογή» και «Επικοινωνία».
register.jsp	Από τη σελίδα αυτή οι χρήστες μπορούν να δημιουργούν νέους λογαριασμούς.

search.jsp	Αυτή η σελίδα περιέχει κώδικα για την αναζήτηση αρχείων.
view.jsp	Η σελίδα αυτή όταν ο χρήστης επιλέγει να ανοίξει ένα αρχείο αναλαμβάνει να το διαβάσει από το δίσκο και να το μεταφέρει στο φυλλομετρητή του πελάτη.

Όλα τα παραπάνω όπως έχει ειπωθεί προϋποθέτουν την ύπαρξη μιας MySQL βάσης δεδομένων και έναν εξυπηρετητή Java Application Server.

6.1 Αναλυτική επεξήγηση των αρχείων JSP

header.inc

Το **header.inc** περιέχει κώδικα αρχικοποίησης. Από μόνο του δεν εκτελείται αλλά όλα τα υπόλοιπα .jsp αρχεία το καλούν και τοποθετούν το περιεχόμενό του στο πάνω μέρος τους. Το καταφέρνουν αυτό χρησιμοποιώντας την εντολή `include`. Ο τρόπος με τον οποίο το κάνουν αυτό είναι πρώτα να τοποθετούν τον επιθυμητό τίτλο της σελίδας σε μια μεταβλητή με την ονομασία `title` και κατόπιν με την `include`. Για παράδειγμα η σελίδα **catinfo.jsp** έχει σαν πρώτες εντολές τις:

```
<% String title = "Categories@E-lib"; %>
<%@ include file="header.inc" %>
```

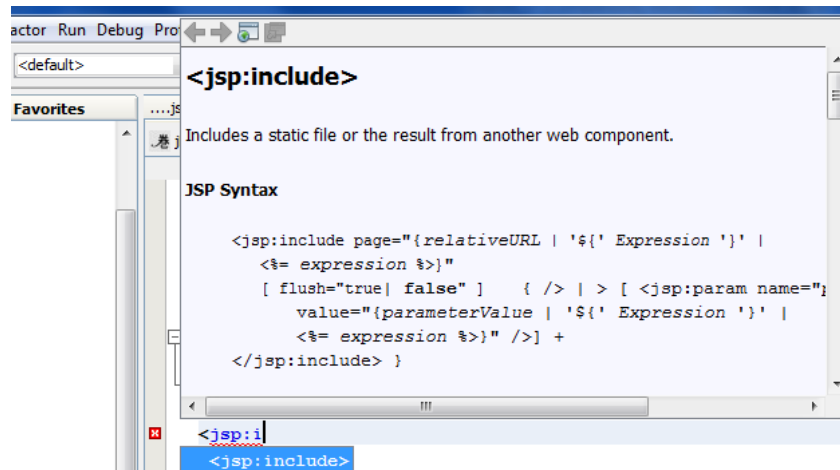
Το αρχείο **header.inc** αρχικά περιέχει τις παρακάτω εντολές:

```
<%@page language="java" pageEncoding="utf8" contentType="text/html;
charset=UTF8"%>

<%@ taglib uri="http://struts.apache.org/tags-bean" prefix="bean" %>
<%@ taglib uri="http://struts.apache.org/tags-html" prefix="html" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

Η πρώτη εντολή ορίζει το encoding του αρχείου. Κάνοντας το **utf8** (δηλαδή Unicode) επιτρέπει την τοποθέτηση και αποθήκευση ελληνικών και ξένων χαρακτήρων. Οι επόμενες τέσσερις εντολές αφορούν την ανάγνωση των κατάλληλων βιβλιοθηκών. Οι βιβλιοθήκη **sql** αφορά τη διασύνδεση με τη βάση δεδομένων και η **core** ενεργοποιεί αρκετές δυνατότητες όπως: υποστήριξη μεταβλητών, Flow control, URL management κ.α. Τέλος τα **tags-beans** και **tags-html** είναι απαραίτητα από το Struts. Μια ακόμα από τις δυνατότητες που δίνουν τα tags στο εργαλείο ανάπτυξης της εφαρμογής (Netbeans) είναι η *δυναμική συμπλήρωση κώδικα*.



Στο παραπάνω παράδειγμα, μέσα από το Netbeans προγραμματίζεται η εφαρμογή. Ο προγραμματιστής έχει πληκτρολογήσει `<jsp:i` και με την υποστήριξη των tags το Netbeans προτείνει τη χρήση της εντολής `<jsp:include>` και παραλλήλως εμφανίζει ένα pop up που εξηγεί τη χρήση και τη σύνταξη της εντολής.

Συνεχίζοντας να εξηγούμε το `header.inc` οι επόμενες εντολές που περιέχει είναι κυρίως HTML κώδικας:

```
<html>
  <head>
    <style type="text/css" media="screen">
      @import url("css/tools.css");
      @import url("css/typo.css");
      @import url("css/forms.css");
      @import url("css/layout-navtop-localleft.css");
      @import url("css/layout.css");
    </style>

    <title><%=title%></title>
```

Ο παραπάνω κώδικας έχει τις απαραίτητες εντολές HTML για έναρξη σελίδας και κατόπιν ακολουθούν τα `import` των CSS. Τα αρχεία CSS ή αλλιώς Cascading Style Sheets αφορούν την οπτική μορφοποίηση των σελίδων. Δηλαδή ποιο Font θα χρησιμοποιηθεί που, ποιο Font-Size, ποιο χρώμα και ποια μορφοποίηση. Η εντολή στο `<title> </title>` τοποθετεί το περιεχόμενο της μεταβλητής `title` σαν τίτλο της σελίδας.

Η συνέχεια του header.inc είναι η παρακάτω. Ουσιαστικά τοποθετεί την εικόνα – λογότυπο της εφαρμογής και ορισμένα άλλα βασικά συστατικά της σελίδας.

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body id="page-home">
  <div id="page">

    <div id="header" class="clearfix">
      <div id="branding">
        
      </div><!-- end branding -->
      <hr />
    </div><!-- end header -->
```

Το αρχείο header.inc είναι 32 γραμμών. Ο κώδικας αυτός αν δε τοποθετούταν στο αρχείο αυτό θα έπρεπε να βρισκόταν **σε κάθε αρχείο** με κατάληξη jsp. Υπήρχε δηλαδή αρχικά ο ίδιος κώδικας σε περίπου 20 αρχεία. Κατόπιν για να τακτοποιηθεί λίγο ο κώδικας, το κομμάτι που επαναλαμβανόταν τοποθετήθηκε στο header.inc και κατόπιν και τα 20 αρχεία το καλούν με την εντολή include.

UploadDoc.jsp

Το UploadDoc.jsp όπως και τα περισσότερα άλλα αρχεία JSP εμπεριέχουν το παραπάνω αρχείο header.jsp με την εντολή

```
<%@ include file="header.inc" %>
```

Κατόπιν δημιουργεί τη σύνδεση με τη βάση δεδομένων και αποθηκεύει ένα από τα SQL Queries που θα εκτελέσει.

```
<sql:setDataSource var="elib" dataSource="jdbc/elib"/>
```

```

        <sql:query
        dataSource="{elib}"
        var="dt"
        sql="SELECT idcategories_tbl,name FROM categories_tbl" />

```

Η πρώτη γραμμή του παραπάνω κώδικα δημιουργεί τη σύνδεση με το datasource. Η ερώτηση που έρχεται στο νου μας είναι «Πώς συνδέεται με τη βάση δεδομένων? Ποιο όνομα χρήστη χρησιμοποιεί και ποιον κωδικό πρόσβασης?»

Η απάντηση είναι ότι χρησιμοποιήθηκε μια μεθοδολογία η οποία επιτρέπει να μην τοποθετούμε τα παραπάνω στοιχεία σε κάθε JSP αρχείο της εφαρμογής. Αντιθέτως τα τοποθετούμε σε ένα αρχείο, αυτό με την ονομασία context.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<Context antiJARLocking="true" path="/elib">
  <Resource auth="Container"
    driverClassName="com.mysql.jdbc.Driver"
    maxActive="8"
    maxIdle="4"
    name="jdbc/elib"
    password=""
    type="javax.sql.DataSource"
    url="jdbc:mysql://localhost:3306/elib"
    username="root"/>
</Context>

```

Ορίζουμε δηλαδή μια φορά το DataSource σε ένα αρχείο XML και κατόπιν το καλούμε σε κάθε αρχείο JSP που το χρειαζόμαστε με μια και μόνο εντολή.

Η συνέχεια του αρχείου UploadDoc.jsp είναι φόρμες που χρησιμοποιούν ένα table για να τοποθετήσουν τα στοιχεία που θέλουμε να εμφανιστούν στην οθόνη του χρήστη

```

<div id="content" class="clearfix">
  <div id="main">
    <c:choose>
      <c:when test="{requestScope.status eq 'success'}">

```

```

        <bean:write name="DocForm" property="mesg" filter="false"/>
    </c:when>
    <c:otherwise>
        <c:choose>
            <c:when test="{param.ro eq 'ronly'}"><strong>Προβολή
Στοιχείων</strong></c:when>
            <c:otherwise>
                <c:choose>
                    <c:when test="{param.v eq 'invalid'}"><strong>Επεξεργασία
αρχείου</strong><br></c:when>
                    <c:otherwise><strong>Ανέβασε αρχείο στο E-Lib !</strong><br></c:otherwise>
                </c:choose>
            </c:otherwise>
        </c:choose>

        <%boolean ro = false;%>
        <c:if test="{param.ro eq 'ronly'}">
            <% ro = true;%>
        </c:if>

        <html:form action="/createdoc" enctype="multipart/form-data" onsubmit="docu-
ment.DocForm.v.value='validate'" readonly="<%=ro%>">

        <table >
            <thead>
                <tr>
                    <td colspan="4">
                        <bean:write name="DocForm" property="error" filter="false"/>
                        &nbsp;</td>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Τίτλος:</td>
                    <td><html:text property="doctitle" /></td>
                </tr>
            </tbody>
        </table>

```

```

<tr>
  <td>
    Προβολή: </td>
    <td><html:select property="visibility" >
      <html:option value="O">Μόνο σε εμένα</html:option>
      <html:option value="G">Μόνο σε ομάδες που ανήκω</html:option>
      <html:option value="A">Σε όλους</html:option>
    </html:select></td>

</tr>
<tr>
  <td>Κατηγορία:</td>
  <td><html:select property="category">
    <c:forEach items="{dt.rowsByIndex}" var="cats">
      <html:option value="{cats[0]}">${cats[1]}</html:option>
    </c:forEach>

    </html:select></td>
</tr>
<tr>
  <td>Συγγραφείς:</td>
  <td><html:text property="authors" /></td>

</tr>
<tr>
  <td>Σελίδες:</td>
  <td><html:text property="pages" /></td>
</tr>
<tr>
  <td>Εκδότης:</td>
  <td><html:text property="publisher" /></td>

</tr>
<tr>
  <td>Έτος έκδοσης Έτος/Μήνας/Ημέρα:
  </td>
  <td><html:select property="published_year" style="width:60px">

```

```

        <c:forEach begin="1900" end="2100" step="1" var="year">
            <html:option value="\${year}">\${year}</html:option>
        </c:forEach>
    </html:select>&nbsp;&nbsp;&nbsp;<html:select property="published_month"
style="width:45px">
        <c:forEach begin="1" end="12" step="1" var="month">
            <html:option value="\${month}">\${month}</html:option>
        </c:forEach>
    </html:select>&nbsp;&nbsp;&nbsp;<html:select property="published_day"
style="width:45px">
        <c:forEach begin="1" end="31" step="1" var="day">
            <html:option value="\${day}">\${day}</html:option>
        </c:forEach>
    </html:select></td>
</tr>
<tr>
    <td>Tags:</td>
    <td><html:text property="keywords" /></td>
</tr>
<tr>
    <td>Σχόλια:</td>
    <td><html:textarea property="comments" /></td>
</tr>
<tr>
    <td>Περιγραφή:</td>
    <td><html:textarea property="summary" /></td>
</tr>
<tr>
    <td>Αρχείο:</td>
    <td><html:file property="thefile" /></td>
</tr>
    <html:hidden property="docid"/>
</tbody>
</table>
<html:hidden property="v"/>
<c:choose>
    <c:when test="\${param.a eq 'u'}"><html:hidden property="a"

```

```

value="u"/></c:when>
    <c:otherwise><html:hidden property="a" value="c"/>
    </c:otherwise>
</c:choose>
<c:choose>
    <c:when test="{param.rolne 'ronly'}">
        <html:submit value="Ανέβασε το αρχείο" />
    </c:when>
</c:choose>
</html:form>
</c:otherwise>
</c:choose>

</div>
<%@include file="menu.jsp" %>

```

Όμως αφού αξιολογήθει η σύνδεση που έχει ο χρήστης εάν θα μπορέσει να ανεβάσει-τροποποιήσει το αρχείο ή όχι. Εκτός από τα πεδία που είναι ελεύθερος ο χρήστης να συμπληρώσει με την βοήθεια της html δημιουργείτε ένα μενού από επιλογές που μπορεί να συμπληρώσει διαλέξει. Υπάρχουν τα μενού τα οποία είναι στατικά όπως το ορατότητα και οι τιμές τους ορίζονται άμεσα από την html όπου οι φράσεις «μόνο σε εμένα» , «μόνο σε ομάδες που ανήκω» και «σε όλους» αντιστοιχούν στο “O, G και A” που είναι οι τιμές που τελικά θα μεταφερθούν και θα χρησιμοποιηθούν στην βάση δεδομένων. Αντίστοιχα θα μπορούσε να ονομαστεί στατικό και το πεδίο «έτος έκδοσης » γιατί δημιουργείται από την αρχική και τελική τιμή που ορίζουμε εμείς μέσα στο αρχείο. Στην συγκεκριμένη περίπτωση από 1/1/1900 μέχρι 31/12/2100. Δυναμικό όμως μπορεί να χαρακτηριστεί το πεδίο της κατηγορίας καθώς οι επιλογές που εμφανίζονται είναι οι κατηγορίες που υπάρχουν μέχρι εκείνη την στιγμή στην βάση δεδομένων περασμένες.

Ουσιαστικά αν θυμηθούμε στην αρχή του αρχείου ορίστηκε μια μεταβλητή dt για το SQL query. Στις παραπάνω γραμμές χρησιμοποιούνται τα στοιχεία που ανακτήθηκαν από τη βάση δεδομένων για να παρουσιαστούν στο χρήστη.

footer.jsp

Το footer.jsp περιέχει το μέρος της σελίδας που θα εμφανίζεται σε κάθε σελίδα της εφαρμογής. Περιέχει τον παρακάτω κώδικα:

```
<%@page language="java" pageEncoding="utf8" contentType="text/html;
charset=UTF8"%>

<div id="footer" class="clearfix">
  <p>&copy; Copyleft - Ιωάννης Γκουτζουρέλας και Δημήτρης Σωτηρίου</p>
</div><!-- end footer -->

</div><!-- end page -->

<div id="extra1">&nbsp;</div>
<div id="extra2">&nbsp;</div>
</body>
</html>
```

Ουσιαστικά όλα τα αρχεία JSP είναι κατά το μεγαλύτερο τους HTML κώδικας που δημιουργεί φόρμες όπου ο χρήστης μπορεί να τοποθετήσει δεδομένα ή να δει υπάρχοντα δεδομένα.

groups.jsp

Στο groups.jsp ο διαχειριστής δημιουργεί τις διάφορες ομάδες εργασίας. Καθώς θέλουμε μόνο αυτός να έχει το συγκεκριμένο δικαίωμα γίνεται στην αρχή του αρχείου έλεγχος εάν το username αντιστοιχεί στο admin και μόνο τότε ο χρήστης μπορεί να συνεχίσει. Σε αντίθετη περίπτωση τον επιστρέφει στο index.jsp.

```
<%@page language="java" pageEncoding="utf8" contentType="text/html; charset=UTF8"%>

<% String title = "Groups@E-lib"; %>
<%@ include file="header.inc" %>

<c:if test="${sessionScope.username ne 'admin'}">
    <jsp:forward page="/index.jsp"/>
</c:if>
```

Στη συνέχεια βλέπουμε ένα sql ερώτημα όπου ζητείται να δημιουργηθεί ένας πίνακας με δύο στήλες. Στην πρώτη στήλη θα περιέχεται το id της ομάδας και στη δεύτερη το όνομα αυτής. Το αποτέλεσμα αυτού του ερωτήματος θα χρησιμοποιηθεί στην συνέχεια για την προβολή των ομάδων. Το success που ακολουθεί δεν αφορά στο mapping στο struts config αλλά είναι ένα session attribute και αποθηκεύεται στο session.

```
<sql:setDataSource var="elib" dataSource="jdbc/elib"/>

<sql:query
    dataSource="${elib}"
    var="dt"
    sql="SELECT idgroups_tbl,name FROM groups_tbl" />

<div id="content" class="clearfix">
    <div id="main">
        <c:choose>
            <c:when test="${requestScope.status eq 'success'}">
                <bean:write name="GroupForm" property="mesg" filter="false"/>
            </c:when>
            <c:otherwise
```

Τέλος έχουμε html κώδικα και βλέπουμε ότι το Δημιουργία/Προβολή ομάδας που πατάει ο διαχειριστής τρέχει το action creategroup από το struts-config.xml. Βλέπουμε επίσης το select box που εμφανίζεται στην εφαρμογή δίπλα στο ομάδες και από όπου μπορεί ο διαχειριστής να δει τις υπάρχουσες ομάδες οι οποίες επιστρέφονται μέσω της μεταβλητής dt που δηλώθηκε παραπάνω,

```
<b>Δημιουργία/Προβολή ομάδας</b>
<br>
<br>
<html:form action="/creategroup">

  <table border="0">
    <thead>
      <tr>
        <td colspan="2">
          <bean:write name="GroupForm" property="error" filter="false"/>
          &nbsp;</td>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Ομάδες:</td>
          <td align="left"><html:select property="groups">
            <option value="" selected></option>
            <c:forEach items="{dt.rowsByIndex}" var="groups">
              <option value="{groups[0]}">{groups[1]}</option>
            </c:forEach>
          </html:select></td>
        </tr>
        <tr>
          <td>Δημιουργία ομάδας:</td>
          <td align="left"><html:text property="groupname" /></td>
        </tr>
      </tbody>
    </table>
  </form>
```

```
</tbody>
</table>

<html:submit value="Δημιουργία" />

</html:form>
</c:otherwise>
</c:choose>

</div>
<%@include file="menu.jsp" %>
```

6.2 Αναλυτική επεξήγηση κώδικα Java

Ο τρόπος με τον οποίο λειτουργεί η εφαρμογή είναι ο εξής. Οι σελίδες JSP, ορισμένες από τις οποίες επικοινωνούν και με τη βάση δεδομένων παρουσιάζουν το γραφικό περιβάλλον του χρήστη, καθώς και ορισμένα δεδομένα. Το κομμάτι αυτό ονομάζεται View.

Η αρχιτεκτονική που χρησιμοποιείται από το Struts είναι το MVC ή αλλιώς Model – View – Controller. Γίνεται δηλαδή separation του κώδικα που ασχολείται με την παρουσίαση της εφαρμογής (View) , το μοντέλο (Model) και την αλληλεπίδραση του χρήστη (Controller).

Όταν ο χρήστης λοιπόν τοποθετήσει κάποια δεδομένα σε μια φόρμα και πατήσει κάποιο κουμπί, ο τρόπος με τον οποίο θα αποκριθεί το σύστημα εξαρτάται από τον κώδικα Java. Για παράδειγμα όταν ο χρήστης θέλει να δημιουργήσει έναν νέο λογαριασμό στην εφαρμογή θα κάνει την αντίστοιχη επιλογή στο site και θα του παρουσιαστεί το περιεχόμενο της σελίδας register.jsp. Το αρχείο αυτό είναι το View (περιέχει το γραφικό περιβάλλον δηλαδή).

Τρία αρχεία Java υπάρχουν για την υποστήριξη του γραφικού αυτού περιβάλλοντος. Τα αρχεία αυτά είναι τα RegisterAction.java, RegisterBean.java και RegisterForm.java

Το αρχείο RegisterBean περιέχει τις μεθόδους που είναι απαραίτητες για την αλληλεπίδραση με τη βάση δεδομένων.

```
public int insertUser(String UserID, String Passwd, int PassQuestion, String PassAnswer, int
groupId)
public boolean chkUserIDInDB(String UserID)
```

Το αρχείο RegisterForm περιέχει μεταβλητές σε Java αντίστοιχες με αυτές που θα χρησιμοποιηθούν στο γραφικό περιβάλλον της εφαρμογής. Στη σελίδα JSP υπάρχουν φόρμες για να συμπληρώσει ο χρήστη στοιχεία όπως το όνομα χρήστη και τον κωδικό πρόσβασης. Τα στοιχεία αυτά για να χρησιμοποιηθούν στο backend θα πρέπει να τοποθετηθούν σε μεταβλητές πρώτα. Το αρχείο RegisterForm ασχολείται δηλαδή αποκλειστικά με τη φόρμα.

```
private String userid;
private String passwd;
private int passquestion;
private String passanswer;
```

```
private String repasswd;  
private String error;  
private String mesg;  
private int groupId;
```

Το τρίτο αρχείο RegisterAction περιέχει κώδικα για τα διάφορα Actions που μπορεί να κάνει ο χρήστης, όπως να πατήσει το κουμπί «**Δημιουργία λογαριασμού**». Αν ο χρήστης δεν έχει συμπληρώσει τα απαραίτητα στοιχεία, θα πρέπει να του εμφανιστεί κάποιο μήνυμα λάθους. Αν προσπαθήσει να χρησιμοποιήσει ένα όνομα χρήστη που ήδη χρησιμοποιείται τότε θα πρέπει πάλι να του εμφανιστεί ένα μήνυμα λάθους. Αν τα στοιχεία του όμως είναι σωστά τότε θα πρέπει να τοποθετηθούν στη βάση δεδομένων. Όλα τα παραπάνω υλοποιούνται στο αρχείο αυτό.

Η παραπάνω λογική χρησιμοποιείται σε όλες τις σελίδες της εφαρμογής. Κάθε τι που βλέπει ο χρήστης είναι ουσιαστικά:

Αρχείο JSP + Αρχείο Action + Αρχείο Form + Αρχείο Bean

Συνοπτικά, το JSP περιέχει το γραφικό περιβάλλον. Το αρχείο Form τοποθετεί τις διάφορες τιμές σε μεταβλητές Java, το αρχείο Bean περιέχει την λογική (τοποθέτηση δεδομένων στη βάση δεδομένων) και το αρχείο Action αναλαμβάνει τον έλεγχο των δεδομένων και την εμφάνιση των μηνυμάτων λάθους.

Ας δούμε μερικά τμήματα αρχείων κώδικα java που παρουσιάζουν ενδιαφέρον. Στο searchBean.java υπάρχει το ερώτημα για αναζήτηση κειμένου στη βάση δεδομένων. Η αναζήτηση γίνεται στο όνομα του τίτλου αλλά και στα tags που έβαλε ο χρήστης που ανέβασε το έγγραφο.

Όπως φαίνεται παρακάτω μετά την σύνδεση με τη βάση δεδομένων γίνεται έλεγχος εάν το user id ισούται με 1 δηλαδή εάν ο χρήστης είναι ο διαχειριστής και τότε προβάλλονται όλα τα αρχεία ανεξαρτήτως δικαιωμάτων αρκεί φυσικά να ταιριάζει στον τίτλο ή στα tags με την λέξη που έβαλε ο διαχειριστής ως κριτήριο για την αναζήτηση.

```
public ArrayList<HashMap> getData(String title,String uid){  
    Connection con = null;  
    Statement stmt = null;  
    ResultSet rs =null;
```

```

ArrayList<HashMap> results = new ArrayList<HashMap>();
try {
    con = dataSource.getConnection();
    stmt = con.createStatement();
    if(uid.equals("1")){
        String query = "SELECT iddocs_tbl,title,c.name,privilege_lvl_id,users.userid,uploaded_date "
+
            " FROM docs_tbl, categories_tbl c,users_tbl users" +
            " WHERE (title LIKE '%" + title + "%' OR keywords LIKE '%" + title + "%')"+
            " AND docs_tbl.uploaded_by=users.idusers_tbl " +
            " AND category_id=idcategories_tbl ";
        rs = stmt.executeQuery(query);
    }
}

```

Στη περίπτωση που ο χρήστης δεν είναι ο διαχειριστής τότε τρέχει ένα ερώτημα που αρχικά παρουσιάζει όλα τα αρχεία που ανέβασε ο ίδιος ο χρήστης και ταιριάζουν με το κριτήριο αναζήτησης

```

else {
    String query2 = "SELECT iddocs_tbl,title,c.name,privilege_lvl_id,users.userid,uploaded_date "
+
        " FROM docs_tbl, categories_tbl c,users_tbl users" +
        " WHERE (title LIKE '%" + title + "%' OR keywords LIKE '%" + title + "%')"+
        " AND uploaded_by="+uid+
        " AND docs_tbl.uploaded_by=users.idusers_tbl " +
        " AND category_id=idcategories_tbl union "
}

```

Στη συνέχεια το αποτέλεσμα του πρώτου ερωτήματος ενώνεται με union με το αποτέλεσμα του άλλου ερωτήματος που εξάγει τα αρχεία που έχουν πρόσβαση όλοι οι χρήστες και ταιριάζουν φυσικά με το κριτήριο αναζήτησης.

```

"SELECT iddocs_tbl,title,c.name,privilege_lvl_id,users.userid,uploaded_date " +
    " FROM docs_tbl, categories_tbl c,users_tbl users" +
    " WHERE (title LIKE '%" + title + "%' OR keywords LIKE '%" + title + "%')"+
    " AND privilege_lvl_id = 'A'+
    " AND docs_tbl.uploaded_by=users.idusers_tbl " +

```

```
" AND category_id=idcategories_tbl union "
```

Και στο τελευταίο ερώτημα βρίσκουμε σε ποια ομάδα ανήκει ο χρήστης και προβάλλονται τα αρχεία που έχουν ανεβάσει οι χρήστες που ανήκουν στην ίδια ομάδα.

```
" SELECT iddocs_tbl,title,c.name,privilege_lvl_id,users.userid,uploaded_date " +  
    " FROM docs_tbl,categories_tbl c, users_tbl users " +  
    " WHERE (title LIKE '%" + title + "%' OR keywords LIKE '%" + title + "%')"+  
    " AND privilege_lvl_id = 'G'" +  
    " AND docs_tbl.uploaded_by=users.idusers_tbl "+  
    " AND category_id=idcategories_tbl " +  
    " AND iddocs_tbl IN "+  
    "(SELECT iddocs_tbl "+  
    "FROM docs_tbl "+  
    " WHERE uploaded_by IN (select idusers_tbl "+  
    " FROM users_tbl "+  
    " WHERE group_id=(SELECT group_id "+  
    "FROM users_tbl  
    "WHERE idusers_tbl="+uid+")));
```

ενώ στο τέλος το hashmap κάνει δυναμικούς πίνακες με 6 στήλες όπου η καθεμία εμπεριέχει τα στοιχεία που φαίνονται και τα προσθέτει στο ArrayList.

```
rs = stmt.executeQuery(query2);  
    }  
    while(rs.next()){  
        HashMap tmap = new HashMap();  
        tmap.put("iddocs", rs.getInt(1));  
        tmap.put("title",rs.getString(2));  
        tmap.put("cid",rs.getString(3));  
        tmap.put("pid", rs.getString(4));  
        tmap.put("uploaded_by",rs.getString(5));  
        tmap.put("uploaded_date",rs.getString(6));  
        results.add(tmap);
```

Ένα άλλο ενδιαφέρον σημείο του κώδικα είναι στο αρχείο GroupAction.java το οποίο δημιουργεί τις διάφορες ομάδες στις οποίες ανήκει κάθε χρήστης. Έτσι βλέπουμε αρχικά να

γίνεται ο έλεγχος για το αν υπάρχει το όνομα της ομάδας στο group bean και να επιστρέφει FAILURE στο struts-config αν όντως υπάρχει.

```
GroupBean gb = new GroupBean(ds);
String gname = gf.getGroupname();
if (gb.chkGrp(gname)) {
    gf.setError("Το όνομα του group υπάρχει ήδη!");
    return mapping.findForward(FAILURE);
} else {
    if (gb.insertGrp(gname) >= 1) {
        gf.setMesg("Το group δημιουργήθηκε με επιτυχία.");
        request.setAttribute("status", "success");
        return mapping.findForward(SUCCESS);
    } else {
        gf.setError("Πρόβλημα κατά την εισαγωγή του group στη βάση δεδομένων.");
        return mapping.findForward(FAILURE);
    }
}
```

Σε αντίθετη περίπτωση που το όνομα δεν υπάρχει τότε γίνεται εισαγωγή της ομάδας που υπάρχει εντωμεταξύ ως το αντικείμενο rs στο GroupBean.java. Στο αρχείο αυτό που φαίνεται παρακάτω, η executeUpdate επιστρέφει τον αριθμό των εγγράφων που επηρεάζονται από την εντολή SQL και στη συγκεκριμένη περίπτωση επιστρέφει πόσες προστέθηκαν από την INSERT για αυτό και γίνεται ο έλεγχος `gb.insertGrp(gname) >= 1` στο GroupAction.java.

```
try {
    con = dataSource.getConnection();
    stmt = con.createStatement();
    rs = stmt.executeUpdate("INSERT INTO groups_tbl(name) " +
        "VALUES('" + gname + "')");
```

Όταν η ομάδα τελικά δημιουργηθεί με επιτυχία εκτός από το μήνυμα προς το χρήστη έχουμε success session attribute στο groups.jsp αλλά και ένα SUCCESS στο struts-config.

Παρακάτω θα εξετάσουμε το αρχείο LoginBean.java το οποίο τοποθετεί τα δεδομένα που εισάγει ο χρήστης στη βάση δεδομένων ώστε να γίνει ο έλεγχος από το LoginAction.java για το αν υπάρχει ή όχι ο συγκεκριμένος χρήστης και αναλόγως να γίνει η είσοδος στην εφαρμογή. Ξεκινάει λοιπόν με την αρχικοποίηση της επικοινωνίας με την βάση δεδομένων καθώς και την αρχικοποίηση του uid.

```
Connection con = null;
Statement stmt = null;
ResultSet rs = null;
int uid=0;
try {
    con = dataSource.getConnection();
    stmt = con.createStatement();
```

Στη συνέχεια έχουμε την .executeQuery η οποία τρέχει ένα ερώτημα και επιστρέφει ένα αντικείμενο result set με αποτελέσματα. Στη συγκεκριμένη περίπτωση παίρνει τιμή το uid το οποίο περιέχει την πρώτη στήλη στον πίνακα της βάσης με το id του χρήστη, μέσω του αντικειμένου rs.

```
rs = stmt.executeQuery("SELECT idusers_tbl FROM users_tbl " +
    "WHERE userid = '" + username + "'" +
    "AND passwd = '" + password + "'");
if (rs.next()) {
    uid = rs.getInt(1);
```

Παρακάτω γίνεται το κλείσιμο της σύνδεσης με τη βάση δεδομένων και επιστρέφεται η τιμή του uid.

```
} finally {
    try {
        if (rs != null) {
            rs.close();
        }
    } catch (SQLException e) {
```

```

    }
    try {
        if (stmt != null) {
            stmt.close();
        }
    } catch (SQLException e) {
    }
    try {
        if (con != null) {
            con.close();
        }
    } catch (SQLException e) {
    }
    }
    return uid;
}
}

```

Η τιμή του uid θα ισούται με το μηδέν εάν δεν γίνει η επιβεβαίωση του χρήστη αλλιώς θα πάρει την τιμή που ισούται με το id του. Ο συγκεκριμένος έλεγχος γίνεται στο LoginAction.-java όπως φαίνεται παρακάτω.

```

LoginBean lb = new LoginBean(ds);//getDataSource(request,"elib");
String passwd = formBean.getPasswd();
String userid = formBean.getUserid();
passwd = passwd.trim();
userid = userid.trim();
if (passwd.equals("") || userid.equals("")) {
    formBean.setError("Δεν επιτρέπεται να αφήνεται κενά τα α) όνομα χρήστη και β) κωδικό πρόσβασης!!");
    return mapping.findForward(Failure);
}
int uid = lb.validateUser(userid, passwd);
if (uid >= 1) {
    HttpSession sess = request.getSession(true);
    sess.setAttribute("uname", userid);
    sess.setAttribute("uid", uid);
}
}

```

```

    sess.setMaxInactiveInterval(600);
    return mapping.findForward(SUCCESS);
} else {
    formBean.setError("Ο χρήστης δεν υπάρχει!! Παρακαλώ δημιουργήστε νέο λογαριασμό ή
βάλτε τα σωστά στοιχεία σας.");
    return mapping.findForward(FAILURE);
}
}
}
}

```

Τέλος βλέπουμε στο RegisterAction.java το πώς δημιουργείται και στέλνεται ένα html email με ένα μήνυμα καλωσορίσματος αφού επιβεβαιωθεί ότι έγινε η εγγραφή ενός νέου χρήστη στην εφαρμογή και στη συνέχεια όπως σε κάθε action έχουμε ένα status success ή failure που αφορά στο struts-config.

```

if (rb.insertUser(userid, passwd, passquestion, passanswer, groupId) >= 1) {
    HtmlEmail email = new HtmlEmail();
    email.setHostName("mail.smalls.gr");
    email.setAuthentication("e-lib@vpn.gr", "ptixiaki123");
    email.setCharset("UTF-8");
    email.addTo(userid, userid);
    email.setFrom("admin@elib.org", "Ομάδα E-Lib");
    email.setSubject("Δημιουργία νέου λογαριασμού");
    email.setMsg("Καλώς ήλθατε στη πλατφόρμα E-Library...");
    email.send();
    request.setAttribute("status", "success");
    formBean.setMesg("Ο λογαριασμός σας δημιουργήθηκε. Μπορείτε να χρησιμοποιήσετε την
εφαρμογή.");
    return (mapping.findForward(SUCCESS));
} else {
    formBean.setError("Πρόβλημα με τη σύνδεση με τη βάση δεδομένων.");
    return (mapping.findForward(FAILURE));
}
}

```

Όμοιος έχουμε αποστολή email με τον κωδικό πρόσβασης του χρήστη σε περίπτωση που ο χρήστης τον ξεχάσει, εφόσον απαντήσει σωστά στην ερώτηση ασφαλείας.

```

else if (fuid.equals("passa")) {
    passanswer = passanswer.trim();
    String passwd = fb.chkPassPhrase(passanswer, uid);
    if (!passwd.equals("")) {
        SimpleEmail email = new SimpleEmail();
        email.setHostName("");
        email.setAuthentication("e-lib@vpn.gr", "ptixiaki123");
        email.addTo(uid, uid);
        email.setFrom("admin@elib.org", "E-Lib.gr");
        email.setSubject("Ο κωδικός πρόσβασης σας είναι : ");
        email.setMsg(passwd);
        email.send();

        request.setAttribute("status", "success");
        ff.setMsg("Ο κωδικός πρόσβασης έχει αποσταλλεί με το ηλεκτρονικό ταχυδρομείο.");
        return mapping.findForward(SUCCESS);
    } else {
        ff.setError("Η απάντηση ήταν λανθασμένη. Προσπαθήστε ξανά.");
        return mapping.findForward(FAILURE);
    }
}

```

Ενδιαφέρον έχει εδώ ότι το email δημιουργείται ως SimpleEmail ενώ στο RegisterAction.java προτιμήσαμε τη δημιουργία αντικειμένου HtmlEmail ώστε να έχουμε την δυνατότητα να κωδικοποιήσουμε το μήνυμα, εν προκειμένω ως UTF-8 για να στέλνονται σωστά και οι ελληνικοί χαρακτήρες.

6.3 Αναλυτική επεξήγηση Struts configuration

Μέχρι τώρα παρουσιάστηκαν όλα τα μέρη της εφαρμογής που ουσιαστικά είναι:

Αρχείο JSP + Αρχείο Action + Αρχείο Form + Αρχείο Bean

Αυτό που λείπει είναι ο συνδετικός κρίκος που ενοποιεί τα παραπάνω και επιτρέπει την πλοήγηση από τη μία σελίδα στην επόμενη. Δηλαδή όταν ο χρήστης δημιουργήσει νέο λογαριασμό, τότε πως μεταφέρεται στην επόμενη σελίδα;

Τα struts είναι μια υλοποίηση ανοιχτού κώδικα που ακολουθεί το πρότυπο <<μοντέλο – παρουσίαση – ελεγκτής>> (Model View Controller - MVC), για την κατασκευή διαδικτυακών εφαρμογών βασιζόμενες σε servlets, jsp, xml και java. Βασικό χαρακτηριστικό του struts είναι η λογική της εφαρμογής, η οποία αλληλεπιδρά με την βάση δεδομένων, διαχωρίζεται πλήρως από τις σελίδες που εμφανίζονται στον χρήστη καθώς και τον ελεγκτή που προωθεί τις πληροφορίες μεταξύ της λογικής της εφαρμογής και της σελίδας. Ο ελεγκτής αυτός, ο οποίος καλείται ActionServlet, παρέχεται από τα Struts. Επιπλέον τα struts διευκολύνουν τη συγγραφή των προτύπων για τις σελίδες που θα παρουσιαστούν στους χρήστες. Παρ' όλο που είναι δυνατόν να χρησιμοποιηθεί XML/XSLT καθώς και Velocity για τη συγγραφή των αρχείων, συνήθως αυτά γράφονται σε JSP. Τα .jsp αρχεία συνδέονται με τη λογική εφαρμογής και τον ελεγκτή μέσω ενός κεντρικού αρχείου διαμόρφωσης με το όνομα struts-config.xml. Το αρχείο struts-config.xml όπως και η λογική της εφαρμογής πρέπει να γραφτούν από τον προγραμματιστή.

Όταν ένας χρήστης θέλει να προσπελάσει μια σελίδα, στέλνει μία αίτηση στον ελεγκτή με τη μορφή μιας ενέργειας (action). Τότε ο ελεγκτής καλεί την java κλάση που αντιστοιχεί στην action που έλαβε και σύμφωνα με τα αποτελέσματα που λαμβάνει, στέλνει στον χρήστη την αντίστοιχη σελίδα.

Την δουλειά αυτή την αναλαμβάνει αποκλειστικά το Struts Framework και για αυτό χρησιμοποιήθηκε στην πτυχιακή αυτή εργασία. Αντί να τοποθετηθεί συγκεκριμένος κώδικας που να κάνει “redirect στην σελίδαA.jsp” όλες οι συνδετικές πληροφορίες τοποθετούνται σε ένα αρχείο XML που αναλαμβάνει όλες αυτές τις αρμοδιότητες. Το αρχείο αυτό **struts-con-**

fig.xml ξεκινά περιέχοντας όλες τις φόρμες (αρχεία Java που τοποθετούν τιμές σε μεταβλητές).

```
<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE struts-config PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 1.3//EN"
    "http://jakarta.apache.org/struts/dtds/struts-config_1_3.dtd">

<form-beans>
    <form-bean name="GroupForm" type="com.myapp.struts.GroupForm"/>
    <form-bean name="SearchForm" type="com.myapp.struts.SearchForm"/>
    <form-bean name="DocForm" type="com.myapp.struts.DocForm"/>
    <form-bean name="CatForm" type="com.myapp.struts.CatForm"/>
    <form-bean name="RegisterForm" type="com.myapp.struts.RegisterForm"/>
    <form-bean name="ForgotForm" type="com.myapp.struts.ForgotForm"/>
    <form-bean name="LoginForm" type="com.myapp.struts.LoginForm"/>
</form-beans>
```

Στο παραπάνω φαίνεται ποιο αρχείο JSP αντιστοιχεί σε ποιο αρχείο Java (Form).

Στη συνέχεια το XML αυτό αρχείο περιέχει ορισμένα Forward rules

```
<global-forwards>
    <forward name="home" path="/home.do"/>
    <forward name="register" path="/register.do"/>
    <forward name="admin" path="/admin.do"/>
    <forward name="forgot" path="/forgot.do"/>
    <forward name="category" path="/category.do"/>
    <forward name="upload" path="/upload.do"/>
    <forward name="search" path="/search.do"/>
    <forward name="groups" path="/groups.do"/>
</global-forwards>
```

Στο παραπάνω section τοποθετούνται κάποια mappings. Ο τρόπος με τον οποίο λειτουργεί εξηγείται με το παρακάτω παράδειγμα. Αν είχαμε

```
<global-forwards>
```

```
<forward name="error" path="/WEB-INF/jsp/error.jsp"/>
</global-forwards>
```

Τότε μέσα σε μια σελίδα JSP τοποθετώντας

```
<html:link forward="error">Εδώ</html:link>
```

Θα μεταφερόμασταν στην σελίδα `error.jsp`. Το παραπάνω έχει δύο οφέλη. Πρώτον κατά τον προγραμματισμό χρειάζεται μόνο να θυμόμαστε το «`error`» και όχι το full path “`/WEB-INF/jsp/error.jsp`”. Το δεύτερο όφελος είναι ότι αν επιλέξουμε να αλλάξουμε τη θέση ή το όνομα του αρχείου “`error.jsp`” τότε κατόπιν θα πρέπει να αλλάχθει μόνο το `mapping`. Σε διαφορετική περίπτωση θα έπρεπε να ψάχνουμε δεκάδες αρχεία κώδικα για να αντικατασταθεί κάθε εμφάνιση του “`/WEB-INF/jsp/error.jsp`” με τη νέα ονομασία και θέση του αρχείου.

Στη συνέχεια του αρχείου **`struts-config.xml`** ακολουθούν τα `mappings` των `Actions`.

Λέγοντας `Action-Mappings` ας παρουσιάσουμε το παράδειγμα που ο χρήστης προσπαθεί να κάνει είσοδο στην εφαρμογή:

```
<action input="/"
  name="LoginForm"
  path="/login"
  scope="session"
  type="com.myapp.struts.LoginAction">
  <forward name="success" path="/dashboard.jsp"/>
  <forward name="failure" path="/index.jsp"/>
</action>
```

Στο παραπάνω καθορίζεται ότι το αρχείο `Form` είναι το `LoginForm` και ότι το `path` είναι το `/login`. Αν ο χρήστης κάνει με επιτυχία (`success`) είσοδο στο σύστημα, θα μεταφερθεί στη σελίδα `/dashboard.jsp` διαφορετικά στη σελίδα `/index.jsp` (ώστε να του ζητηθεί ξανά είσοδος).

Φαίνεται ξεκάθαρα ότι αποτελεί το συνδεδετικό κρίκο το αρχείο αυτό, του οποίου μάλιστα μεγάλο μέρος δημιουργείται από το `NetBeans`.

Τα Action-Mappings είναι λοιπόν τα παρακάτω

```
<action-mappings>
  <action input="/admin"
    name="LoginForm"
    path="/adminchk"
    scope="session"
    type="com.myapp.struts.AdminAction">
    <forward name="success" path="/dashboard.jsp"/>
    <forward name="failure" path="/admin.jsp"/>
  </action>
  <action input="/"
    name="LoginForm"
    path="/login"
    scope="session"
    type="com.myapp.struts.LoginAction">
    <forward name="success" path="/dashboard.jsp"/>
    <forward name="failure" path="/index.jsp"/>
  </action>
  <action name="CatForm"
    path="/createcat"
    scope="session"
    type="com.myapp.struts.CatAction">
    <forward name="success" path="/category.jsp"/>
    <forward name="failure" path="/category.jsp"/>
  </action>
  <action input="/upload"
    name="DocForm"
    path="/createdoc"
    scope="session"
    type="com.myapp.struts.DocAction">
    <forward name="success" path="/UploadDoc.jsp"/>
    <forward name="failure" path="/UploadDoc.jsp"/>
```



```

</action>
<action input="/register"
    name="RegisterForm"
    path="/registeract"
    scope="session"
    type="com.myapp.struts.RegisterAction">
    <forward name="success" path="/register.jsp"/>
    <forward name="failure" path="/register.jsp"/>

</action>
<action path="/logout" type="com.myapp.struts.LogoutAction">
    <forward name="success" path="/logout.jsp"/>
</action>
<action input="/"
    name="SearchForm"
    path="/searchdoc"
    scope="session"
    type="com.myapp.struts.SearchAction">
    <forward name="success" path="/search.jsp"/>
    <forward name="failure" path="/search.jsp"/>

</action>
<action input="/"
    name="GroupForm"
    path="/creategroup"
    scope="session"
    type="com.myapp.struts.GroupAction">
    <forward name="success" path="/groups.jsp"/>
    <forward name="failure" path="/groups.jsp"/>

</action>
<action input="/forgot"
    path="/forgotact"
    name="ForgotForm"
    scope="session"
    type="com.myapp.struts.ForgotPasswdAction">
    <forward name="success" path="/forgot.jsp"/>

```

```

    <forward name="failure" path="/forgot.jsp"/>

    </action>

<action forward="/register.jsp" path="/register"/>
    <action forward="/UploadDoc.jsp" path="/upload"/>
    <action forward="/forgot.jsp" path="/forgot"/>
    <action forward="/index.jsp" path="/home"/>
    <action forward="/admin.jsp" path="/admin"/>
    <action forward="/category.jsp" path="/category"/>
    <action forward="/search.jsp" path="/search"/>
    <action forward="/groups.jsp" path="/groups"/>

</action-mappings>

```

Τέλος το αρχείο struts-config.xml ολοκληρώνεται με το παρακάτω

```

<controller processorClass="org.apache.struts.tiles.TilesRequestProcessor"/>
    <message-resources parameter="com/myapp/struts/ApplicationResource"/> <plug-in class-
Name="org.apache.struts.tiles.TilesPlugin" >
        <set-property property="definitions-config" value="/WEB-INF/tiles-defs.xml" />
        <set-property property="moduleAware" value="true" />
    </plug-in>

```

Το τελευταίο αυτό κομμάτι του αρχείου ενημερώνει τον Application Server ότι γίνεται χρήση του Tiles. Αυτό σημαίνει απλά ότι θα αναγνωστεί το αρχείο web.xml. Στο αρχείο αυτό υπάρχουν πληροφορίες σχετικά με το επίπεδο debug που θα χρησιμοποιηθεί, πιθανές χρήσεις proxy, ποια είναι η αρχική σελίδα της εφαρμογής και ποια είναι η σύνδεση με τη βάση δεδομένων.

Η χρήση του Struts Framework απαιτεί αρκετή προσπάθεια για να κατανοηθεί αρχικά και να χρησιμοποιηθεί για πρώτη φορά. Αφού όμως καταλάβει κανείς την λογική του που μπορεί να ανακεφαλαιωθεί στα

Για κάθε σελίδα της εφαρμογής θα υλοποιούνται τα

Αρχείο JSP + Αρχείο Action + Αρχείο Form + Αρχείο Bean

και κατόπιν θα ενημερώνεται το struts-config.xml και το web.xml»

τότε μπορεί κανείς να επωφεληθεί από τα πολλά πλεονεκτήματα που προσφέρει το Struts σε εφαρμογές που βασίζονται σε Java + JSP.

ΚΕΦΑΛΑΙΟ 7. Επίλογος – Συμπεράσματα

Μετά το πέρας αυτής της εργασίας αναδείξαμε κάποιες από τις δυνατότητες τεχνολογιών όπως η java και η SQL τις οποίες γνωρίσαμε και οι ίδιοι καλύτερα ώστε να υπάρχει πλέον το υπόβαθρο για μελλοντική ενασχόληση με αυτές. Δε θεωρούμε ότι εξαντλήσαμε όλες τις δυνατότητες των συγκεκριμένων τεχνολογιών και πιστεύουμε ότι υπάρχει χώρος και για περαιτέρω ανάπτυξη και παραμετροποίηση της συγκεκριμένης εφαρμογής.

Η προσπάθεια μας εστιάστηκε στο να δημιουργήσουμε μία απλή και λειτουργική εφαρμογή προς τον χρήστη και αναπτύξαμε λιγότερο το αισθητικό κομμάτι της. Θεωρούμε ότι η εφαρμογή μπορεί να χρησιμοποιηθεί και σε πραγματικές συνθήκες και να αποτελέσει εργαλείο στην επιστημονική κοινότητα εφόσον διορθωθούν και οι ατέλειες που θα προκύψουν κατά την πραγματική χρήση της. Επιπλέον σημαντικό θα ήταν να εντοπιστούν οι επιπλέον ανάγκες των χρηστών και να παραμετροποιηθεί αναλόγως η εφαρμογή.

Μπορούμε να παρατηρήσουμε σαν παράβλεψη της εφαρμογής την έλλειψη πιο σύνθετων κριτηρίων αναζήτησης τα οποία θα κάνανε πιο εύκολο τον εντοπισμό των ζητούμενων εγγράφων από τον χρήστη. Επίσης χρήσιμο θα ήταν ένα καλύτερο και πιο ανεπτυγμένο σύστημα από διαχείριση δικαιωμάτων που θα επέτρεπε για παράδειγμα ένα χρήστη να ανήκει σε περισσότερες των μία ομάδες ενδιαφέροντος ή και την ύπαρξη διαχειριστών ανά ομάδα.

Στα πλεονεκτήματα της εφαρμογής πιστώνεται ότι είναι ανεξάρτητη από λειτουργικά συστήματα αφού τρέχει σε Java virtual machine. Επιπλέον έχει την δυνατότητα να αποστείλει μέσω του ηλεκτρονικού ταχυδρομείου τον κωδικό πρόσβασης του χρήστη εάν τον ξεχάσει. Επίσης στα πλεονεκτήματα μπορεί να καταμετρηθεί ότι ο χρήστης μπορεί να επιλέξει από ποιόν θα είναι προσπελάσιμο το έγγραφο του. Τέλος θεωρούμε ότι η εφαρμογή καθότι απλή και δυναμική επιτρέπει την κάλυψη των αδυναμιών της εάν κριθεί αναγκαίο.

Βιβλιογραφία - Πηγές

[1] PHP

<http://www.php.net>

[2] ASP - Active Server Pages

<http://www.asp.net/>

[3] JSP - Java Server Pages

<http://www.asp.net/>

[4] The Apache HTTP Server Project

<http://httpd.apache.org/>

[5] [Apache 2 and PHP 5 \(mod_php\)](#)

<http://dan.drydog.com/apache2php.html>

[6] *Scripts* in HTML documents

<http://www.w3.org/TR/REC-html40/interact/scripts.html>

[7] History of PHP and related projects

<http://gr2.php.net/history>

[8] Active Server Pages - ASP

http://en.wikipedia.org/wiki/Active_Server_Pages

[9] Java Servlet Technology

<http://java.sun.com/products/servlet/>

[10] Προγραμματισμός Internet Με Την Τεχνολογία Των Java Server Pages (Jsp)

(Α' Μέρος) - Κακαρόντζας Γιώργος ΤΕΙ Λάρισας Τμήμα Τεχνολογίας Πληροφορικής & Τηλ/νιών Δεκέμβριος 2003

[11] Sakai,

<https://collab.sakaiproject.org/portal/>

[12] eConf,

<http://econf.sourceforge.net/>

[13] Java Media FrameWork

<http://java.sun.com/javase/technologies/desktop/media/jmf/>

[14] OLAT Web Site,

<http://www.olat.org/website/en/html/index.html>

[15] YAML, Yet Another MultiColumn Layout

<http://www.yaml.de/en/home.html>

[16] Apache Tomcat 5.0

<http://tomcat.apache.org/>

[17] Advanced Distributed Learning - SCORM

<http://www.adlnet.gov/scorm/>

[18] Downes, S. (2007). Learning networks in practice, Emerging Technologies for Learning.

Coventry, U.K.: Becta. διαθέσιμο στη διεύθυνση:

http://partners.becta.org.uk/page_documents/research/emerging_technologies07_chapter2.pdf

[19] Adobe AIR

<http://www.adobe.com/products/air/>