



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Σχεδιασμός και Υλοποίηση Συστήματος Διαχείρισης Εγγράφων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Κελεσίδη Νικολάου

Επιβλέπουσα :

Θεοδώρα Βαρβαρίγου
Καθηγήτρια

Αθήνα, Οκτώβριος 2014



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Σχεδιασμός και Υλοποίηση Συστήματος Διαχείρισης Εγγράφων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
του
Κελεσίδη Νικολάου

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 2 Μαρτίου 2015.

.....
Βασίλειος Λούμος
Καθηγητής Ε.Μ.Π.

.....
Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

.....
Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014

.....
Κελεσίδης Νικόλαος

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

Copyright © Κελεσίδης Νικόλαος, 2014

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία είναι αποτέλεσμα έρευνας στον τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής.

Ο στόχος της εργασίας είναι η υλοποίηση ενός συστήματος διαχείρισης εγγράφων (document management system) εξειδικευμένου για τις ανάγκες ενός ευρωπαϊκού ερευνητικού έργου και της κοινοπραξίας που το υλοποιεί.

Θα ήθελα να ευχαριστήσω ιδιαίτερα την υπεύθυνη στην εκπόνηση της διπλωματικής την κ. Θ. Βαρβαρίγου όπως επίσης και τον επιβλέποντα κ. Κ. Τσερπέ για τη δυνατότητα που μου δόθηκε μέσω αυτής της εργασίας να αποκτήσω ποικίλες εμπειρίες πάνω στα θέματα με τα οποία ασχολείται ο Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής.

Κελεσίδης Νικόλαος

ΠΕΡΙΛΗΨΗ

Ο κύριος στόχος της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός συστήματος διαχείρισης εγγράφων. Ο στόχος ενός τέτοιου συστήματος είναι η διαχείριση των εγγράφων συμπεριλαμβανομένων λειτουργιών όπως είναι η αποθήκευση αρχείων, ανάκτηση, κοινοποίηση, διαγραφή. Ωστόσο, η κύρια λειτουργία που υποστηρίζεται είναι η ασύγχρονη ομαδική επεξεργασία ενός αρχείου. Το σύστημα διατηρεί μια μονάδα ελέγχου ταυτότητας, εξουσιοδότησης και λογιστικής που μεταξύ άλλων παρέχει δικαιώματα πρόσβασης σε συγκεκριμένες ομάδες χρηστών. Η συνοδευτική αναφορά δίνει έμφαση στις διάφορες αρχιτεκτονικές που χρησιμοποιούνται για το χαρακτηρισμό των υπηρεσιών, καθώς και το επίπεδο των βασικών τεχνολογιών με το οποίο το σύστημα είναι χτισμένο.

Το συγκεκριμένο σύστημα έχει σχεδιαστεί έχοντας κατά νου τις απαιτήσεις ενός ερευνητικού έργου και της κοινοπραξίας που το υλοποιεί. Σε μια τέτοια περίπτωση, η συνεργατική παραγωγή και διαμοιρασμός περιεχομένου, καθώς και η διάκριση των ρόλων με βάση τη διαχειριστική δομή είναι σημαντική για την υλοποίηση ενός τέτοιου συστήματος.

Λέξεις κλειδιά:

Σύστημα Διαχείρισης Εγγράφων, αποθήκευση εγγράφων, ανάκτηση εγγράφων, διαγραφή εγγράφων, χαρακτηρισμός εγγράφων.

ABSTRACT

The main objective of the present diplomatic thesis is the design and the implementation of a document management system. The objective of such a system is to manage documents allowing operations such as file saving, recovering, notifying, deleting. However the main operation supported is the asynchronous collaborative editing of a file. The system maintains an authentication, authorization and accounting module which –amongst others- provides access rights to specific user groups. The accompanying report emphasizes on the various architectures employed for the featured services as well as the baseline technologies upon which the system is built.

The current system was designed having in mind the requirements of a project and the consortium which runs it. In such a case, the collaborative generation of content as well as the sharing and finally the distinction of roles based on the management structure are important driving needs.

Keywords:

Document Management System, Document Saving, Document Recovery, Document Deleting, Document Characterization

Πίνακας Περιεχομένων

ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ	11
1.1 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ	11
1.2 ΟΡΓΑΝΩΣΗ ΤΟΜΟΥ.	12
ΚΕΦΑΛΑΙΟ 2 - ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΓΓΡΑΦΩΝ	14
2.1 ΕΙΣΑΓΩΓΗ	14
2.2 ΓΕΝΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ	15
2.2.1 Συστήματα αποθήκευσης εγγράφων.....	15
2.2.2 Ανάκτηση εγγράφων	16
2.2.3 Δεικτοδότησης εγγράφων.....	16
2.2.4 Συστήματα ασφάλειας εγγράφων	19
2.3 ΥΠΑΡΧΟΝΤΑ ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΓΓΡΑΦΩΝ	20
2.4 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ.....	22
2.5 ΔΙΑΧΕΙΡΙΣΗ ΕΓΓΡΑΦΩΝ ΚΑΙ ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ.....	24
ΚΕΦΑΛΑΙΟ 3 - ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	27
3.1 ΡΟΛΟΙ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	27
3.1.1 Περιγραφή	27
3.1.2 Ρόλοι και η ανάθεση εργασίας.....	29
3.2 ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΕΣ	33
3.2.1 Διαχείριση αρχείων και φακέλων	33
3.2.2 Διαμοιρασμός Αρχείων.....	37
3.2.3 Επιλογές εγγραφής για αποστολή ειδοποιήσεων	38
3.2.4 Πρόσβαση χρηστών.	39
3.2.5 Chat.....	39
3.2.6 Ανάλυση των Απαιτήσεων Ασφάλειας.....	41
3.3 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	42
ΚΕΦΑΛΑΙΟ 4 - ΤΕΧΝΟΛΟΓΙΕΣ	45
4.1 ΔΥΝΑΤΟΤΗΤΕΣ SPRING FRAMEWORK	46
4.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ SPRING FRAMEWORK.....	47
4.3 SPRING MVC	49
4.3.1 Ο «κύκλος ζωής» μίας αίτησης στο Spring MVC.....	51
4.4 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ SPRING FRAMEWORK.....	53
4.5 JAVASERVER PAGES STANDARD TAG LIBRARY (JSTL).....	54
4.6 HIBERNATE ORM.....	55
4.7 JAVASCRIPT	59
4.8 JAVA	60
4.9 MYSQL	62
4.10 ΛΟΙΠΕΣ ΤΕΧΝΟΛΟΓΙΕΣ.....	62
ΚΕΦΑΛΑΙΟ 5 - ΒΙΒΛΙΟΓΡΑΦΙΑ	64

1

Εισαγωγή

Κεφάλαιο 1 - Εισαγωγή

1.1 Περιγραφή της διπλωματικής

Καθώς ο όγκος των -πάσης φύσεως- εγγράφων αυξάνεται δραματικά στις επιχειρήσεις, η διαχείρισή τους αποτελεί μία ιδιαίτερα πιεστική πρόκληση. Αν η επιχείρησή ανήκει στην πλειονότητα των εταιρειών όπου έγγραφα, e-mails, φαξ, αρχεία PDF, CAD και άλλου τύπου συγκεντρώνονται σε καθημερινή βάση και αποθηκεύονται -στην καλύτερη περίπτωση- σε file servers ή υπολογιστές διάσπαρτους στην εταιρεία η απουσία συστήματος διαχείρισης εγγράφων εκθέτει την επιχείρησή σε διάφορες προκλήσεις που, μεταξύ άλλων, περιλαμβάνουν καθυστερήσεις στην εξυπηρέτηση των πελατών, υπερβάλλον κόστος και προβλήματα συμμόρφωσης με διάφορους κανονισμούς.

Τα σημερινά συστήματα λογισμικού για τη διαχείριση εγγράφων δεν έχουν καμία σχέση με τις παλιότερες λύσεις και συνδυάζουν δυνατότητες διαχείρισης εγγράφων και αρχείων σε μία ενιαία πρόταση. Οι προτάσεις αυτές επιτρέπουν στην επιχείρηση να ελαττώνει σημαντικά το χρόνο, το κόστος και την πολυπλοκότητα της αποθήκευσης εγγράφων και της πρόσβασης σε αυτά καθ' όλη τη διάρκεια ζωής της πληροφορίας. Επίσης, της παρέχουν τη δυνατότητα να βελτιώνει την πρόσβαση στα κρίσιμα έγγραφα, αλλά και να εμπλουτίζει την εξυπηρέτηση των πελατών και τη λειτουργική αποδοτικότητα. Και όλα αυτά μειώνοντας τους κινδύνους και συμβαδίζοντας με τις απαιτήσεις συμμόρφωσης των όποιων κανονισμών.

Στόχος αυτής της διπλωματικής είναι ο σχεδιασμός, ανάλυση και η υλοποίηση ενός συστήματος διαχείρισης εγγράφων εξειδικευμένου για τις ανάγκες ενός ευρωπαϊκού ερευνητικού έργου και της κοινοπραξίας που το υλοποιεί. Ένα σύστημα διαχείρισης εγγράφων οργανώνει, συγκεντρώνει, χαρακτηρίζει και τακτοποιεί τα έγγραφα που υπάρχουν σε μια επιχείρηση και αναλαμβάνει την εύκολη διακίνησή τους μεταξύ των μελών της κοινοπραξίας, επιτρέπει στις επιχειρήσεις-μέλη να επιλέγουν από το πλήθος των λειτουργιών που προσφέρει (ταξινόμηση, αρχειοθέτηση και ανάκτηση, διαχείριση εγγράφων, διαχείριση αρχείων-καταλόγων, ενοποίηση, πρόσβαση

στον Web και διανομή μέσω αυτού), ώστε να ικανοποιούν τις ανάγκες τους και να επιτυγχάνουν τους στόχους της. Για τις ανάγκες του συγκεκριμένου ερευνητικού έργου δημιουργήθηκε εφαρμογή η οποία προσφέρει τις εξής λειτουργίες:

- Ανάλυση των απαιτήσεων του συστήματος
- Ρόλοι των χρηστών (admin, project coordinator, technical manager, workpackage leader, task leader, task participant)
- Δημιουργία φακέλων
- Αποθήκευση (ανέβασμα) αρχείων σε φακέλους
- Κατέβασμα αρχείων από χρήστες
- Υποσημείωση αρχείων (έκδοση, σχόλια, ημερομηνία αποθήκευσης, ημερομηνία τροποποίησης, ιδιοκτήτης)
- Διαχείριση δικαιωμάτων αρχείων
- Διαχείριση εκδόσεων (δυνατότητα επαναφοράς της προηγούμενης έκδοσης)
- Δοιαμιρασμός αρχείων
- Επιλογές εγγραφής για αποστολή ειδοποιήσεων για αλλαγές σε σχέση με ένα αρχείο με αποστολή email
- Προβολή χρηστών οι οποίοι είναι logged in
- Chat μεταξύ χρηστών.

Ο κώδικας της εφαρμογής βρίσκεται στην διεύθυνση:

<https://github.com/kelas-spirit/dms>

1.2 Οργάνωση τόμου.

Ο τόμος της διπλωματικής είναι οργανωμένος σε 5 βασικά κεφάλαια. Στο κεφάλαιο 2 γίνεται μια περιγραφή ενός σύγχρονου συστήματος διαχείρισης εγγράφων, περιγράφονται τα χαρακτηριστικά και η δομή του. Επίσης, γίνεται μια ιστορική αναδρομή και παρουσιάζονται κάποια απ' τα πιο δημοφιλή συστήματα διαχείρισης εγγράφων που κυκλοφορούν στην αγορά.

Στο κεφάλαιο 3 παρουσιάζεται ο σχεδιασμός και η υλοποίηση του συστήματος που υλοποιήθηκε. Ιδιαίτερη έμφαση θα δοθεί στην αρχιτεκτονική του συστήματος με use case diagrams. Περιγράφονται οι ρόλοι πρόσβασης που χρησιμοποιούνται στο σύστημα.

Στο κεφάλαιο 4 γίνεται εισαγωγή στις τεχνολογίες που έχουν χρησιμοποιηθεί για την υλοποίηση του συστήματος.

Στο κεφάλαιο 5 δίνεται βιβλιογραφία.

2

Σύστημα Διαχείρισης Εγγράφων

Κεφάλαιο 2 - Σύστημα Διαχείρισης Εγγράφων

2.1 Εισαγωγή

Το **Σύστημα διαχείρισης εγγράφων** (DMS) είναι ένα υπολογιστικό σύστημα που χρησιμοποιείται για την παρακολούθηση και αποθήκευση ηλεκτρονικών εγγράφων ή φυσικών εντύπων. Παρέχει κάποια εργαλεία όπως αποθήκευση, ιστορικό παλαιότερων εκδόσεων, μεταδιδόμενα, ασφάλεια, καθώς και δημιουργία ευρετηρίου και ανάκτηση εγγράφων.

Η αποδοτική διαχείριση των εγγράφων είναι κρίσιμου σημασίας για την επιτυχία και την εξέλιξη μιας επιχείρησης. Η αυξανόμενη ανάγκη για αποδοτικότερη διαχείριση των εγγράφων μιας εταιρίας έδωσε ώθηση στην ανάπτυξη των Συστημάτων Διαχείρισης Εγγράφων τα τελευταία χρόνια. Στο σύγχρονο απαιτητικό επιχειρησιακό περιβάλλον, είναι απαραίτητο ο κάθε οργανισμός να εκμεταλλεύεται κάθε δυνατότητα οργάνωσης και ανάπτυξης που προσφέρει ο τομέας της πληροφορικής.

Η ηλεκτρονικά διαθέσιμη γνώση σε πληροφοριακά συστήματα και δίκτυα είναι δυνατό να αποτελέσει πηγή ανταγωνιστικού πλεονεκτήματος αλλά και μοχλό αύξησης της παραγωγικότητας των εργαζομένων γραφείου για τους περισσότερους οργανισμούς σήμερα. Ο συνεχώς αυξανόμενος όγκος της όμως, όπως και η κατανομή της σε πολλά, διάσπαρτα συστήματα μπορούν να αποτελέσουν εμπόδιο στην αφομοίωση και την αξιοποίησή της. Στην περιοχή αυτή, λύση έρχονται να δώσουν τα Συστήματα Διαχείρισης Εγγράφων, που υποστηρίζουν τον εντοπισμό και την αναζήτηση των διαθέσιμων πόρων γνώσης. Τα συστήματα αυτά, εφόσον συνοδεύονται από κατάλληλες διαδικασίες είναι δυνατό να ενισχύσουν σημαντικά την «εταιρική» μνήμη, να επιταχύνουν τον κύκλο εξοικείωσης των εργαζομένων με νέα συστήματα

και προϊόντα και να βελτιώσουν τη συνεργασία ομάδων.

Κάποια από τα βασικά χαρακτηριστικά του DMS:

- Αποθηκευτικός χώρος
- Διαδικασία προσθήκης εγγράφου
- Διαδικασία χαρακτηρισμού και αναζήτησης εγγράφου μέσα στο αποθηκευτικό χώρο
- Διαδικασία ανάκτησης/αφαίρεσης εγγράφου

2.2 Γενικές απαιτήσεις

Ως βασικές απαιτήσεις του συστήματος διαχείρισης αρχείων θα μπορούσαμε να αναφέρουμε τα εξής:

- αποθήκευση εγγράφων
- ανάκτηση εγγράφων
- δεικτοδότηση εγγράφων
- ασφάλεια εγγράφων

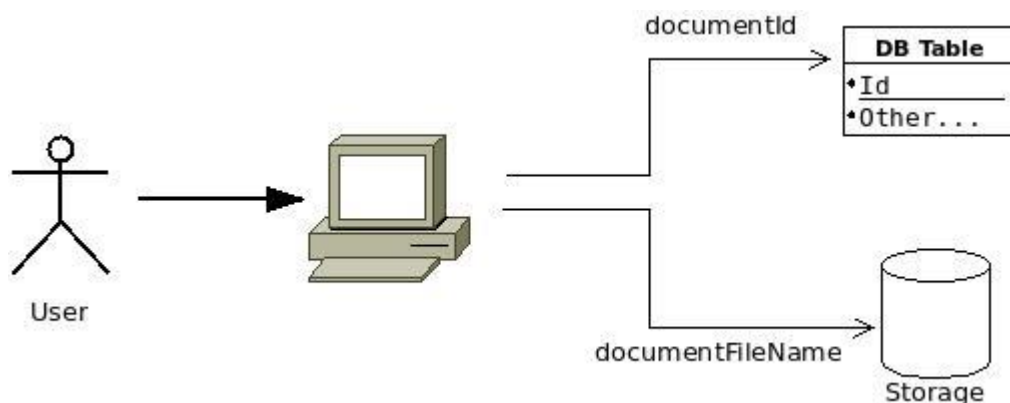
2.2.1 Συστήματα αποθήκευσης εγγράφων

Η λειτουργία αποθήκευσης εγγράφων περιλαμβάνει τη διαχείριση των εγγράφων όπου είναι αποθηκευμένα για όσο χρονικό διάστημα πραγματοποιείται η μεταφορά των εγγράφων από το ένα μέσο αποθήκευσης στο άλλο καθώς και για την ενδεχόμενη καταστροφή του εγγράφου.

Τα σημαντικότερα μέρη από τα οποία αποτελείται ένα σύστημα αποθήκευσης εγγράφων είναι:

Σύστημα Αρχείων: χώρος όπου γίνεται η φυσική αποθήκευση των εγγράφων

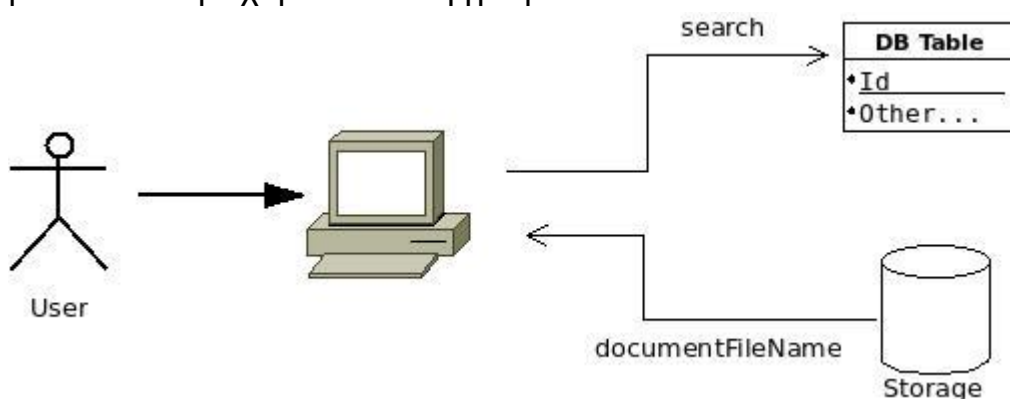
Βάσεις δεδομένων: κάποια χαρακτηριστικά που έχουν δοθεί στα αποθηκευμένα έγγραφα αποθηκεύονται στην βάση δεδομένων για ευκολία στην διαχείριση εγγράφων με εφαρμογές.



Σχήμα 2.1 Αποθήκευση εγγράφων

2.2.2 Ανάκτηση εγγράφων

Η ανάκτηση των ηλεκτρονικών εγγράφων από τον αποθηκευτικό χώρο είναι μια διαδικασία που μπορεί να είναι αρκετά περίπλοκη και δύσκολη. Μια απλή ανάκτηση των μεμονωμένων εγγράφων μπορεί να υποστηριχθεί, επιτρέποντας στο χρήστη να καθορίσει το μοναδικό αναγνωριστικό έγγραφο, και το σύστημα θα χρησιμοποιήσει το βασικό δείκτη για να ανακτήσει το έγγραφο. Μια πιο ευέλικτη ανάκτηση επιτρέπει στο χρήστη να καθορίσει τους επιμέρους όρους αναζήτησης που αφορούν το αναγνωριστικό έγγραφο. Έτσι, επιστρέφεται μια λίστα που ανταποκρίνεται στους όρους αναζήτησης του χρήστη. Υπάρχουν συστήματα που δίνουν την δυνατότητα να καθοριστεί μια έκφραση που περιέχει πολλές λέξεις-κλειδιά ή φράσεις που αναμένεται να υπάρχουν μέσα στα περιεχόμενα των εγγράφων.



Σχήμα 2.2 Ανάκτηση εγγράφων

2.2.3 Δεικτοδότησης εγγράφων

Η λειτουργία αυτή αφορά στην οργάνωση, ταξινόμηση, αρχειοθέτηση

των εγγράφων για μελλοντική χρήση. Η ταχύτητα εισαγωγής και ανάκτησης εγγράφων, η αξιοπιστία των αναζητήσεων εξαρτώνται σε μεγάλο βαθμό απ' το πόσο καλά έχει δημιουργηθεί το σύστημα δεικτοδότησης, οπότε το σύστημα δεικτοδότησης παίζει καθοριστικό ρόλο για την ποιότητα του συστήματος διαχείρισης εγγράφων.

#	METADATA_ID	COMMENTS	CREATED_USER	CREATED_DATE	DOCUMENT_FILE_NAME	SIZE	DOCUMENT_TYPE	OWNER	SUBJECT	UPDATED_DATE
1	1	pdf	admin	0000-00-00 00:00:00	read_me.pdf	200		admin	doc	0000-00-00 00:00
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Υποθέτοντας ότι χρησιμοποιούμε μια σχεσιακή βάση δεδομένων τα γνωρίσματα των εγγράφων θα μπορούσαν να έχουν την εξής μορφή:

Id: Αριθμός-αναγνωριστικό κάθε εγγράφου το οποίο είναι μοναδικό. Η μοναδικότητα του αριθμού μας εξασφαλίζει ότι με το συγκεκριμένο id υπάρχει ένα μοναδικό έγγραφο.

Comments: Κάποια επιπλέον σχόλια για συγκεκριμένο έγγραφο που καθοδηγούν τον χρήστη σε ευκολότερη κατανόηση του περιεχομένου του εγγράφου ή ακόμα και σε ευκολότερη αναζήτηση του.

Create User: Όνομα(ή nickname/username) του χρήστη ο οποίος δημιούργησε (έκανε upload) το συγκεκριμένο ηλεκτρονικό έγγραφο.

Created Date: Ημερομηνία δημιουργίας του εγγράφου.

Document File Name: Όνομα του αρχείου.

Size: Μέγεθος του εγγράφου. Το συγκεκριμένο γνώρισμα συνήθως χρησιμοποιείται όταν περιορίζεται το μέγεθος των εγγράφων που θα δημιουργηθούν.

Document Type: Τύπος εγγράφου που δημιουργήθηκε.

Owner: Ιδιοκτήτης(-ες) του εγγράφου, το συγκεκριμένο γνώρισμα είναι πολύ χρήσιμο ειδικά όταν γίνεται διαμοιραζόμενο (shared) το έγγραφο μεταξύ χρηστών.

Subject: επικεφαλίδα του εγγράφου.

Updated Date: Ημερομηνία της τελευταίας τροποποίησης.

Parrent Id: αριθμός-αναγνωριστικό του φακέλου στο οποίο ανήκει το έγγραφο.

Για αποδοτικότερη μέθοδο οργάνωσης εγγράφων θα μπορούσαν αυτά επίσης να είναι οργανωμένα σε ηλεκτρονικούς φακέλους.

#	FOLDER_ID	CREATED_USER	CREATED_DATE	FOLDER_FILE_NAME	FOLDER_PATH	PARRENT_ID	PATH_ID
1	272	b	2014-05-12 06:43:09	kelas1	/	0	0 0
2	288	b	2014-05-12 15:43:00	maria1	/	0	0 0
3	330	b	2014-05-12 20:56:27	maria2	/maria1	288	0 0 288
4	331	b	2014-05-13 19:42:21	kelas2	/kelas1	272	0 0 272
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Έτσι, τα γνωρίσματα των φακέλων θα μπορούσαν να έχουν την εξής μορφή:

Folder Id: Αριθμός-αναγνωριστικό κάθε φακέλου το οποίο είναι μοναδικό. Η μοναδικότητα του αριθμού μας εξασφαλίζει ότι με το συγκεκριμένο id υπάρχει ένα μοναδικό φάκελο.

Created User: Όνομα(nickname/username) του χρήστη ο οποίος δημιούργησε (έκανε upload) το συγκεκριμένο ηλεκτρονικό φάκελο.

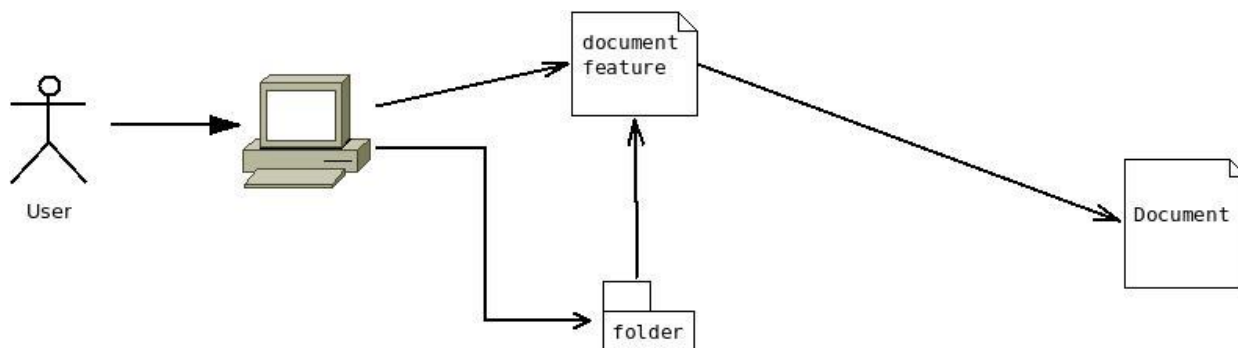
Created Date: Ημερομηνία δημιουργίας του φακέλου.

Folder Name: Όνομα του φακέλου.

Folder Path: Η τοποθεσία του φακέλου.

Parrent Id: Το γνώρισμα αυτό μας δείχνει το id του φακέλου μέσα στο οποίο βρίσκετε ο συγκεκριμένος φάκελος .

Path Id: Το γνώρισμα αυτό αποτελείται από τα id των φακέλων και δείχνει την τοποθεσία του συγκεκριμένο φακέλου. Πολύ χρήσιμο για διαγραφεί των φακέλων.

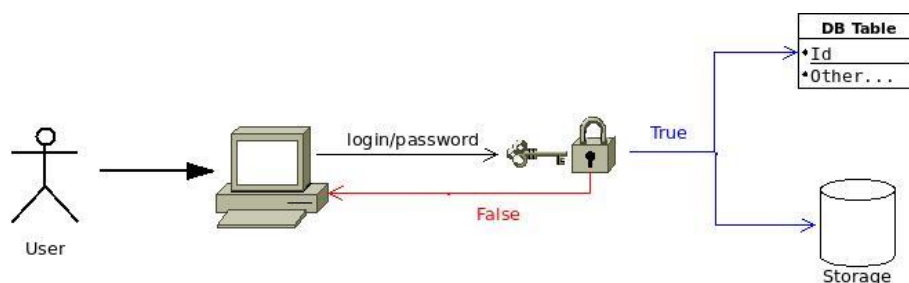


Σχήμα 2.3 Δεικτοδότηση εγγράφων

2.2.4 Συστήματα ασφάλειας εγγράφων

Η ασφάλεια των εγγράφων είναι ζωτικής σημασίας σε πολλές εφαρμογές διαχείρισης εγγράφων. Σκοπός του συστήματος είναι πρόσβαση στο σύστημα, διαχείριση δικαιωμάτων των εγγράφων (ανάγνωση/τροποποίηση), διαχείριση των ρόλων(admin, project coordinator, technical manager, workpackage leader, task participant) των εγγεγραμμένων χρηστών καθώς και επιτρεπτές λειτουργίες που τους αναλογούν.

Τα περισσότερα ζητήματα ασφάλειας που προκύπτουν είναι αποτέλεσμα της έλλειψης του ελέγχου πάνω στη δομή από τη μεριά του χρήστη ή της επιχείρησης. Οι περισσότερες επιχείρησης δεν γνωρίζουν που αποθηκεύονται τα δεδομένα και τι είδος μηχανισμοί λαμβάνουν χώρα για να τα προστατέψουν(π.χ. αν τα δεδομένα είναι κρυπτογραφημένα).



Σχήμα 2.4 Ασφάλεια εγγράφων

2.3 Υπάρχοντα Συστήματα Διαχείρισης Εγγράφων

Παρακάτω παρουσιάζονται και αναλύονται κάποια από τα ήδη υπάρχοντα Συστήματα Διαχείρισης Εγγράφων:

Dropbox: Το Dropbox είναι μια υπηρεσία που επιτρέπει την αποθήκευση, τον συγχρονισμό και την κοινή χρήση αρχείων μεταξύ διαφορετικών συσκευών που ο χρήστης έχει επιλέξει να συνδέσει στο λογαριασμό του. Η υπηρεσία προσφέρει από 2GB έως 16GB δωρεάν αποθηκευτικού χώρου ενώ είναι διαθέσιμη σε Windows, Mac, Linux, iPhone, iPad, Android και BlackBerry.

Ιδρύθηκε το 2007 στο San Francisco από τους Drew Houston και Arash Ferdowsi, φοιτητές του MIT. Η εφαρμογή κυκλοφόρησε για πρώτη φορά το Σεπτέμβριο του 2008 ενώ τον Οκτώβριο του 2011 έγινε διαθέσιμο στην αγορά το Dropbox for Business, μια υπηρεσία που παρέχει τόσο διοικητικό έλεγχο στο εργασιακό περιβάλλον όσο και διαθέσιμο χώρο για τη συνεργασία των εργαζομένων. Σήμερα, το Dropbox προσμετρά 200 εκατομμύρια χρήστες παγκοσμίως ενώ υπολογίζεται ότι ένα δισεκατομμύριο αρχεία αποθηκεύεται σε αυτό καθημερινά.

Η αυξημένη αποδοχή που γνώρισε η υπηρεσία Dropbox πυροδότησε την δημιουργία πολυάριθμων εφαρμογών οι οποίες χρησιμεύουν ως εργαλεία για τη διευκόλυνση των χρηστών του Dropbox, με την προσέλκυση των οποίων προσδοκούσαν να αποκτήσουν μια θέση στην αγορά. Οι πιο δημοφιλείς σήμερα είναι οι:

DropItToMe: Το Dropbox επιτρέπει την κοινοποίηση αρχείων από το χρήστη για να τα κατεβάζουν άλλα άτομα όμως το ανέβασμα αρχείων από άλλα άτομα στο λογαριασμό του χρήστη είναι αυξημένης δυσκολίας. Με το DropItToMe το ανωτέρω καθίσταται εφικτό ενώ η υπηρεσία παρέχει επίσης έναν εύκολο τρόπο για τον ίδιο το χρήστη να ανεβάζει αρχεία στο λογαριασμό του στο Dropbox.

BoxCryptor: Η υπηρεσία αυτή κρυπτογραφεί αυτόματα τα αρχεία του χρήστη προτού αυτά ανέβουν στο λογαριασμό του στο Dropbox χρησιμοποιώντας κρυπτογράφηση AES 256 bit.

Droptunes : Το Dropbox δεν παρέχει δυνατότητα αναπαραγωγής των αρχείων ήχου που έχουν αποθηκευτεί στον διακομιστή της υπηρεσίας επομένως για να επιτευχθεί η αναπαραγωγή ο χρήστης καλείται να τα κατεβάσει ξανά στη συσκευή του. Το Droptunes μεταδίδει τα αρχεία ήχου στο διαδίκτυο, παρακάμπτοντας τη χρονοβόρα αυτή διαδικασία.

IFTTT: Δημιουργεί συνδέσμους μεταξύ υπηρεσιών διαδικτύου με σκοπό να απλουστεύσει και να αυτοματοποιήσει τις ενέργειες του χρήστη. Με άλλα λόγια αποθηκεύει συνταγές στις οποίες εφόσον μια ενέργεια πραγματοποιηθεί, πυροδοτείτε μια αυτόματη αντίδραση.

Για παράδειγμα, ο χρήστης μπορεί να ζητήσει όλα τα συνημμένα που λαμβάνει στο Gmail να αποθηκεύονται αυτόματα στο Dropbox.

Send to Dropbox: Δημιουργεί μια διεύθυνση ηλεκτρονικού ταχυδρομείου, στην οποία ό,τι συνημμένο αποστέλλεται αποθηκεύεται αυτόματα στο Dropbox.

Google Drive: Το Google Drive είναι υπηρεσία αποθήκευσης και συγχρονισμού αρχείων που παρέχεται από την Google, κυκλοφόρησε στις 24 Απριλίου του 2012, και επιτρέπει την χρήση αποθηκευτικού νέφους τον διαμερισμό αρχείων και την συνεργατική επεξεργασία από τον χρήστη. Τα αρχεία που μοιράζονται δημόσια στο Google Drive μπορούν να αναζητηθούν με μηχανές αναζήτησης.

Το Google Drive προσφέρει σε όλους τους χρήστες του έναν αρχικό online χώρο αποθήκευσης 15 GB (αρχικά ήταν 5GB), που μπορεί να χρησιμοποιηθεί από τις τρεις πιο διαδεδομένες υπηρεσίες: το Google Drive, το Gmail και της Φωτογραφίες του Google+. Ο χρήστης μπορεί να λάβει επιπλέον χώρο αποθήκευσης, ο οποίος μοιράζεται μεταξύ Picasa και Google Drive, από 100 GB μέχρι και 16TB μέσω της καταβολής μηνιαίας συνδρομής (US4.99\$ το μήνα για 100GB). Ο χρήστης με τον πληρωμένο αποθηκευτικό χώρο δεν λαμβάνει δωρεάν χώρο μαζί με τον πληρωμένο χώρο.

SkyDrive: Το SkyDrive επιτρέπει στους χρήστες να αποθηκεύουν αρχεία στο Διαδίκτυο και να αποκτούν πρόσβαση σε αυτά από οποιονδήποτε υπολογιστή. Η υπηρεσία SkyDrive χρησιμοποιεί το Windows Live ID περιορίζοντας την πρόσβαση στα ιδιωτικά αρχεία ενός χρήστη, τον διαμοιρασμό τους με επαφές ή τη δημοσιοποίησή τους. Η υπηρεσία προσφέρει χώρο 25 GB, με μέγιστο μέγεθος αρχείου 50 MB ανά αποστολή. Κάθε φορά μπορούν να αποθηκευτούν μέχρι και πέντε αρχεία. Οι τροφοδοσίες RSS επιτρέπουν στους χρήστες να ενημερώνονται για ενημερωμένο περιεχόμενο που οι άλλοι τυχόν τοποθέτησαν στους δημόσιους φακέλους τους. Οι χρήστες μπορούν να εγγραφούν στις τροφοδοσίες RSS με την περιήγησή τους στον δημόσιο φάκελο στο Windows Live SkyDrive και κάνοντας κλικ στο εικονίδιο των τροφοδοσιών RSS στην πάνω δεξιά γωνία του παραθύρου. Η τροφοδοσία RSS περιέχει μια προεπισκόπηση των αρχείων που προστέθηκαν – είτε μέσω μιας μικρογραφίας μιας εικόνας είτε μέσω ενός εικονιδίου ενός αρχείου – και συνδέσεις για τη λήψη του αρχείου από το Windows Live SkyDrive.

Digicube: Η εταιρεία Digicube εξειδικεύεται στις υπηρεσίες ψηφιοποιήσεις

αρχείων και συλλογών καθώς και την, επεξεργασία, διαχείριση και ανάδειξή τους στο εταιρικό δίκτυο ή στο διαδίκτυο, διαθέτοντας συστήματα διαχείρισης περιεχομένου προσανατολισμένα σε κάθε τομέα δραστηριότητας.

Noetron: Το σύστημα Document καλύπτει την διαχείριση επιχειρησιακών εγγράφων, την κεντρική και ασφαλή τους αποθήκευση, και την αποτελεσματική αναζήτηση και ανάκλησή τους. Παράλληλα παρέχει μεγάλες δυνατότητες προσαρμογής των λύσεων στις εκάστοτε ανάγκες της επιχείρησης, μέσω εργαλείων ανάπτυξης προσαρμοσμένων εφαρμογών διαχείρισης εγγράφων.

2.4 Ιστορική Αναδρομή

Το πρώτο εργαλείο διαχείρισης εγγράφων που κατασκευάστηκε είναι ,πιθανότατα , ο συνδετήρας. Στην συνέχεια έκαναν την εμφάνιση τους το συρραπτικό , ο φάκελος και τελικά τα ντουλάπια αρχειοθέτησης. Έπειτα, στις αρχές του 1980, ξέσπασε η επανάσταση των προσωπικών υπολογιστών. Καθώς η υπολογιστική ισχύς αποκεντρώθηκε, το ίδιο συνέβη και με την ικανότητα συλλογής και δημιουργίας πληροφορίας, ιδίως με την μορφή των εγγράφων. Ωστόσο, η επιμέλεια των εγγράφων ακολουθούσε την παλιά νοοτροπία. Οι δισκέτες που περιείχαν τα έγγραφα σε ηλεκτρονική μορφή, τοποθετούνταν μαζί με τα αυθεντικά έγγραφα σε έναν κεντρικό αποθηκευτικό χώρο. Η χωρητικότητα αποθήκευσης των δισκετών ,ωστόσο, ήταν περιορισμένη, με αποτέλεσμα να αποθηκεύονται μόνο τα πιο σημαντικά αρχεία στην δισκέτα.

Στα τέλη του 1980, ο όγκος της πληροφορίας που παραγόταν έγινε απαγορευτικά μεγάλος, γεγονός που προβλημάτισε σοβαρά τις επιχειρήσεις. Με τον ερχομό, όμως, των τοπικών δικτύων (LAN) ,τα πράγματα άλλαξαν. Στην αρχή, τα τοπικά δίκτυα χρησιμοποιούνταν κυρίως για υπηρεσίες εκτύπωσης. Η χωρητικότητα των δίσκων ήταν μικρή και η αγορά τους πανάκριβη, με αποτέλεσμα να αποθηκεύονται μόνο τα πιο σημαντικά αρχεία στον κύριο αποθηκευτικό χώρο. Αργότερα, οι τιμές των συσκευών αποθήκευσης έπεσαν , με αποτέλεσμα να γίνει δυνατή η μαζική αποθήκευση των αρχείων σε έναν κεντρικό υπολογιστή και η κοινή τους χρήση από άλλους σταθμούς εργασίας στο τοπικό δίκτυο. Το πρόβλημα της αποθήκευσης και ανταλλαγής εγγράφων είχε ξεπεραστεί και αυτό θεωρήθηκε μεγάλο κατόρθωμα.

Ωστόσο, γρήγορα εμφανίστηκε ένα άλλο πρόβλημα. Δεν υπήρχε κανένας απολύτως μηχανισμός για την διαχείριση των εγγράφων, με αποτέλεσμα σύντομα να δημιουργηθεί ένα χάος μεταξύ τους. Δεν υπήρχε καμία μέθοδος απ' την στιγμή της δημιουργίας των εγγράφων, που θα επέτρεπε να

ξεχωρίσεις ποια έγγραφα είναι σημαντικά και ποια όχι. Η επανάσταση των υπολογιστών είχε μειώσει τον απαιτούμενο χρόνο παραγωγής, αλλά παράλληλα δημιουργήσε νέα προβλήματα. Ο τεράστιος όγκος από δεδομένα που δεν είχαν φιλτραριστεί ούτε είχαν και κάποια δομή, έκανε την επαναχρησιμοποίηση της πληροφορίας πολύ δύσκολη καθώς δεν υπήρχαν αποτελεσματικοί μέθοδοι για την αναζήτηση συγκεκριμένων εγγράφων.

Στα πλαίσια ανάπτυξης της τεχνολογίας δικτύων εμφανίστηκε το Διαδίκτυο (Internet) μαζί με την τεχνολογία ηλεκτρονικού ταχυδρομείου. Το γεγονός αυτό άρχισε να αλλάζει τον τρόπο με τον οποίο οι άνθρωποι χειρίζονται τα έγγραφα. Η έννοια της διακινούμενης πληροφορίας διευρύνθηκε, και η επίσημη ή μη επικοινωνία μέσω ηλεκτρονικού ταχυδρομείου έγινε δημοφιλής. Αυτό αν και αποτελούσε σημαντικό επίτευγμα για την γρήγορη διακίνηση πληροφορίας, ωστόσο αποτελούσε παράλληλα και ένα παραπάνω βήμα προς το χάος των εγγράφων. Ήταν πλέον φανερό. Υπήρχε η άμεση ανάγκη για την δημιουργία συστημάτων που θα είναι σε θέση να διαχειριστούν και να ανακτήσουν τα έγγραφα αποδοτικά.

Στα μέσα του 1990, η διαχείριση εγγράφων αναπτύχθηκε σημαντικά. Η σάρωση, η δεικτοδότηση, η επεξεργασία, η αποθήκευση και η ανάκτηση ηλεκτρονικών εγγράφων αποτέλεσε πυρήνας των συστημάτων διαχείρισης εγγράφων. Αρχικά τα συστήματα διαχείρισης εγγράφων αφορούσαν την διαχείριση εγγράφων που υπήρχαν σε χαρτί.

Ακολούθησαν τα πρώτα συστήματα διαχείρισης ηλεκτρονικών εγγράφων (EDMS) , που περιλάμβαναν την επεξεργασία ηλεκτρονικών εγγράφων και την διακίνηση τους στα τοπικά και τα μητροπολιτικά δίκτυα, και φυσικά στο Διαδίκτυο. Η μορφή των ηλεκτρονικών εγγράφων σιγά σιγά άρχισε να ποικίλει , με αποτέλεσμα να δημιουργούνται σύνθετα ηλεκτρονικά έγγραφα που να περιλαμβάνουν εκτός από κείμενο και εικόνες, ήχο και βίντεο.

Σήμερα, έχουν αναπτυχθεί αρκετά συστήματα διαχείρισης εγγράφων που εκμεταλλεύονται την σύγχρονη ανάπτυξη της τεχνολογίας σε όλους τους τομείς. Ωστόσο θα μπορούσε να πει κανείς πως η ανάπτυξη των συστημάτων διαχείρισης εγγράφων δεν έχει καλύψει ικανοποιητικά τις σύγχρονες ανάγκες διαχείρισης της πληροφορίας. Κάποιοι μελέτες είχαν κάνει την εκτίμηση ότι το 1985 ο όγκος των εγγράφων διπλασιαζόταν , με ρυθμό μια φορά κάθε πέντε χρόνια.

Το 2003 ο ρυθμός αυτός υπολογίστηκε περίπου ότι είναι μια φορά κάθε 6 μήνες. Αυτός είναι και ο κύριος λόγος για τον οποίο υπάρχει περιορισμένη ανάπτυξη στον τομέα της διαχείρισης εγγράφων. Δημιουργούμε πληροφορία πιο γρήγορα από ποτέ, αλλά δεν είμαστε σε θέση να την ανακτήσουμε με

έναν αποδοτικό τρόπο.

2.5 Διαχείριση εγγράφων και μηχανές αναζήτησης

Η αναζήτηση των πληροφοριών στο επιχειρηματικό περιβάλλον αποτελεί έννοια πολύ στενά συνδεδεμένη με τη διαχείριση εγγράφων. Στο συγκεκριμένο χώρο υπάρχει σημαντική κινητικότητα, η οποία αναμένεται να διαμορφώσει το μελλοντικό τοπίο.

Το Google έχει συνδέσει αναμφίβολα το όνομά του με την πραγματοποίηση αναζητήσεων στον Web, γεγονός που έχει προκαλέσει το ενδιαφέρον ανταγωνιστικών εταιρειών, όπως η Microsoft. Η μάχη των δύο εταιρειών δεν περιορίζεται στις αναζητήσεις στον Web, αλλά μεταφέρεται και στα επιχειρηματικά δεδομένα μετά την απόφαση του Google να επεκτείνει σταδιακά τις δραστηριότητές του και στο συγκεκριμένο τομέα. Βέβαια, εκεί πρέπει να αντιμετωπίσει τη Microsoft, η οποία διαθέτει αναμφίβολη ισχύ λόγω της κυριαρχίας της στο επιχειρηματικό λογισμικό.

Η τεχνολογία της Microsoft, Windows Live Search, αποτελεί ανεξάρτητη εφαρμογή, η οποία επιτρέπει στους χρήστες να αναζητούν πληροφορίες στα PCs, το Internet και τα επιχειρηματικά δίκτυα. Συλλέγει αποτελέσματα αναζητήσεων από τον Web χρησιμοποιώντας την υπηρεσία αναζήτησης MSN και από το PC του χρήστη με το πρόγραμμα desktop αναζητήσεων της εταιρείας και με τη νέα τεχνολογία που εξετάζει τα επιχειρηματικά δεδομένα. Ο Bill Gates θεωρεί ότι το προϊόν αναζήτησης επιχειρηματικών δεδομένων της Microsoft θα βοηθήσει τους πελάτες να καλύψουν το κενό σε αυτό που ονομάζει “το τελευταίο μίλι της παραγωγικότητας”. Δεδομένου ότι οι εργαζόμενοι υπερφορτώνονται από μεγάλους όγκους πληροφοριών, οι οποίες έρχονται από όλες τις κατευθύνσεις, αποτελεί σημαντική πρόκληση να βρουν τις πληροφορίες που χρειάζονται στην εργασία τους.

Η πρωτοβουλία της Microsoft δεν αφορά στο σύνολο του επιχειρηματικού λογισμικού. Συγκεκριμένα, για να λειτουργήσει η τεχνολογία αναζήτησης, πρέπει να διασφαλιστεί η ψηφιακή συνδεσιμότητα με τις εφαρμογές. Η Microsoft αναμένεται να παρουσιάσει ανάλογους συνδέσμους με τα προϊόντα των Siebel και SAP, ενώ σταδιακά τόσο η ίδια όσο και οι συνεργάτες της θα προχωρούν στην ανάπτυξη νέων συνδέσμων.

Από την άλλη μεριά, το Google προσπαθεί να μεταφέρει την επιτυχία του από τον Web στο επιχειρηματικό περιβάλλον. Συγκεκριμένα, η εταιρεία διαθέτει hardware συσκευές (mini και search appliances) για την πραγματοποίηση αναζητήσεων στα επιχειρηματικά δεδομένα. Ταυτόχρονα, έχει έρθει σε συμφωνία με αρκετούς μεγάλους κατασκευαστές επιχειρηματικού λογισμικού,

όπως οι Cognos, SAS και Salesforce.com, προκειμένου να καταστήσει εφικτή την πραγματοποίηση αναζητήσεων στις εφαρμογές τους.

3

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Κεφάλαιο 3 - Αρχιτεκτονική του συστήματος

Στο κεφάλαιο 2 αναλύσαμε το γενικό μοντέλο ενός συστήματος διαχείρισης εγγράφων. Στο παρόν κεφάλαιο θα παρουσιαστεί η αρχιτεκτονική του συστήματος που υλοποιήθηκε.

Σκοπός του συστήματος που υλοποιήσαμε, είναι η αποδοτική διαχείριση και οργάνωση ηλεκτρονικών εγγράφων ,που διαθέτουν κάποιοι χρήστες, από έναν κεντρικό εξυπηρετητή.

3.1 Ρόλοι του συστήματος

3.1.1 Περιγραφή

IT Project Administrator

Ο διαχειριστής έργων πληροφορικής (IT Project Administrator) συμβάλλει απαραίτητα στην διοικητική υποστήριξη για να συντονίσουν μια τεχνική ομάδα, συνήθως υπό τη διεύθυνση του διαχειριστή του έργου. Στα καθήκοντα του διαχειριστή του έργου περιλαμβάνουν όλες οι πτυχές της διευκόλυνσης του έργου: τον προγραμματισμό και το μέρος συναντήσεων, πρακτικά των συνεδριάσεων, ανάπτυξη παρουσιάσεων, καθώς και τη διοργάνωση κατάρτισης για το προσωπικό. Επιπλέον, ο διαχειριστής συμμετέχει στη διοίκηση του προϋπολογισμού, παρέχοντας ανάλυση και την πρόβλεψη των οικονομικών επιδόσεων. Διαχειριστές του έργου πληροφορικής μπορούν να συνεργαστούν στενά με άλλες εταιρικές υπηρεσίες.

IT Project Coordinator:

Συντονίζει όλα τα έργα πληροφορικής και εξασφαλίζει ότι οι πόροι της εταιρείας χρησιμοποιούνται κατάλληλα. Συντάσσει εκθέσεις για την κατάσταση του έργου, συντονίζει χρονοδιαγράμματα του έργου και εντοπίζει και επιλύει τεχνικά προβλήματα. Αναγνωρίζει και αναλύει τις απαιτήσεις των συστημάτων και ορίζει το πεδίο εφαρμογής του έργου, τις απαιτήσεις και τα παραδοτέα.

Συντονίζει τις δραστηριότητες του έργου και διασφαλίζει ότι όλες οι φάσεις του έργου είναι τεκμηριωμένα κατάλληλα.

IT Technical Manager

Ο τεχνικός διευθυντής είναι μια θέση που απαιτεί επαρκείς διευθυντικές ικανότητες, καθώς και επαρκείς τεχνικές δεξιότητες. Ένας τεχνικός διευθυντής θα πρέπει να είναι σε θέση να εκτελεί όλες τις απαραίτητες ενέργειες που σχετίζονται με τα συστήματα της εταιρείας, περιλαμβανομένων την αντιμετώπιση τυχόν τεχνικών ζητημάτων. Πρέπει να επιβλέπει την ανάπτυξη και την εφαρμογή οποιουδήποτε συστήματος σε όλα τα στάδια του έργου. Θα πρέπει να βεβαιωθεί ότι η ομάδα τους είναι πλήρως εκπαιδευμένη και ικανή για να εκτελέσουν το έργο. Ο τεχνικός διευθυντής είναι το άτομο στον οποίον απευθύνονται για τυχόν ζητήματα που προκύπτουν μέσα στην ομάδα. Είναι υπεύθυνος για τη λήψη σημαντικών αποφάσεων για τεχνικά ζητήματα που σχετίζονται με την εταιρεία.

IT Work Package Leader

Η διαχείριση του κάθε πακέτου εργασίας (Work Package) από έναν work package leader γίνεται για την εξασφάλιση της απόδοσης και της προόδου του πακέτου εργασίας σε σχέση με το συνολικό σχέδιο του έργου. Ο διαχειριστής πακέτων εργασίας θα αξιολογεί τις εκθέσεις προόδου που παρέχονται από τους συμμετέχοντες στο έργο, καθώς και για την επιτυχή ολοκλήρωση των στόχων και των καθηκόντων. Οι διαχειριστές πακέτων εργασίας διορίζονται από τους εταίρους σύμφωνα με την εμπειρία τους και τις δεξιότητες διαχείρισης που απαιτούνται σε επιμέρους πακέτα εργασίας.

Οι ευθύνες ενός διαχειριστή πακέτων εργασίας είναι τα εξής:

- παρακολούθηση της προόδου του πακέτου εργασίας κατά τη διάρκεια του έργου και τον προϋπολογισμό κονδυλίων. Πρέπει να βεβαιωθεί ότι το πακέτο εργασίας πληροί τους στόχους που προγραμματίστηκαν.
- ειδοποίηση τον συντονιστή σε περίπτωση καθυστέρησης ή παράλειψης κατά την εκτέλεση του πακέτου εργασίας.
- παράδοση εξαμηνιαίας αναφοράς προόδου στον συντονιστή (coordinator).
- προετοιμασία των προτάσεων για την επικαιροποίηση του σχεδίου εργασίας.

Task Participant

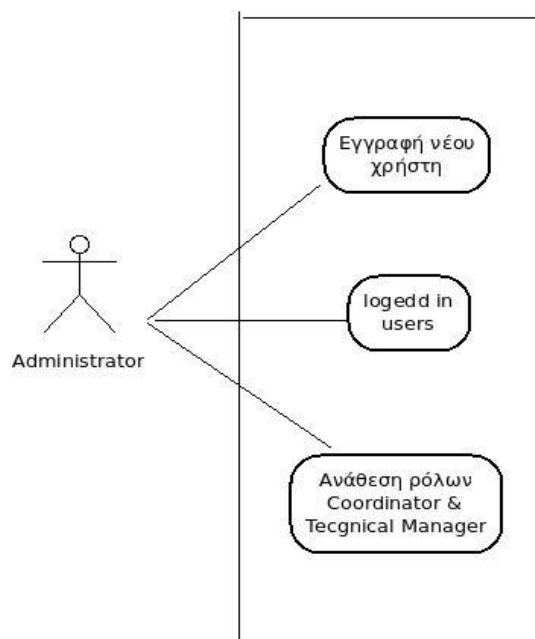
Οι συμμετέχοντες στο έργο (Task Participants), συνεργάζονται για την

επίτευξη της αποστολής του έργου. Έχουν συνήθως περιοχές ευθύνης και κάποιους συγκεκριμένους ρόλους. Σε μικρά έργα ένα άτομο μπορεί να λάβει ταυτόχρονα διάφορους ρόλους, αλλά σε μεγάλα έργα οι ρόλοι συνήθως χωρίζονται. Έτσι, μεταξύ των συμμετεχόντων του έργου, υπάρχουν ιδιαίτερες απαιτήσεις για την κατανομή των ρόλων.

3.1.2 Ρόλοι και η ανάθεση εργασίας

Παρακάτω, παρουσιάζονται οι δυνατότητες των χρηστών που είναι εγγεγραμμένοι στο σύστημα.

Οι διαχειριστές μέσω από την σελίδα που παρουσιάζει τους χρήστες που είναι εγγεγραμμένοι στο σύστημα (ή από αναζήτηση) μπορούν να αναθέσουν ρόλο σε έναν συγκεκριμένο χρήστη ή να διαγράψουν τον ρόλο αυτό αν το κατέχει ήδη. Οι ρόλοι που μπορεί να αναθέσει ο διαχειριστής σε έναν χρήστη είναι “συντονιστής έργων” και “τεχνικός διευθυντής”. Επίσης, ο διαχειριστής μπορεί να κάνει εγγραφή έναν νέο χρήστη και να του αναθέσει τον ρόλο του διαχειριστή. Να σημειωθεί εδώ ότι ο κάθε εγγεγραμμένος χρήστης έχει τουλάχιστον έναν ρόλο “user (χρήστης)”. Επίσης, εφόσον ο “user” το επιθυμεί μπορεί να διαγράψει τον εαυτό του από οτιδήποτε άλλο ρόλο (εκτός από το ρόλο “user”).



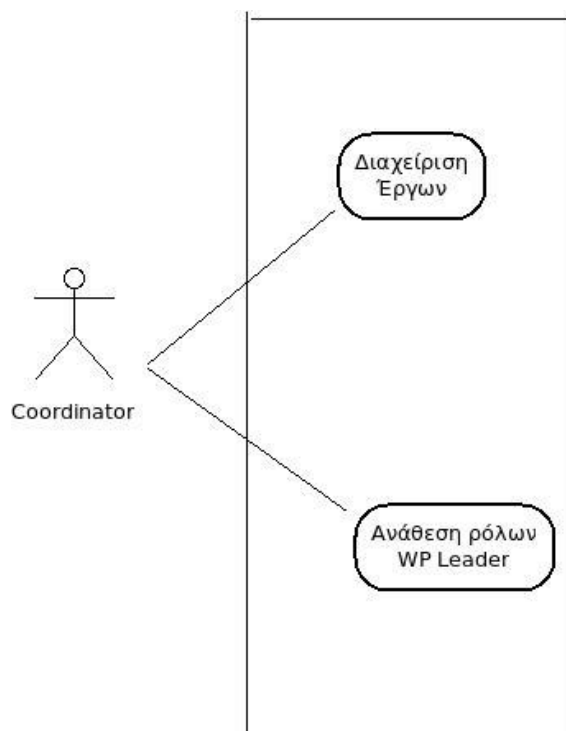
Σχήμα 3.1. Role Administrator

Ο τεχνικός διευθυντής (Technical Manager) είναι αυτός με τον οποίον επικοινωνούν οι χρήστες για διάφορα ζητήματα.



Σχήμα 3.2.Role Technical manager

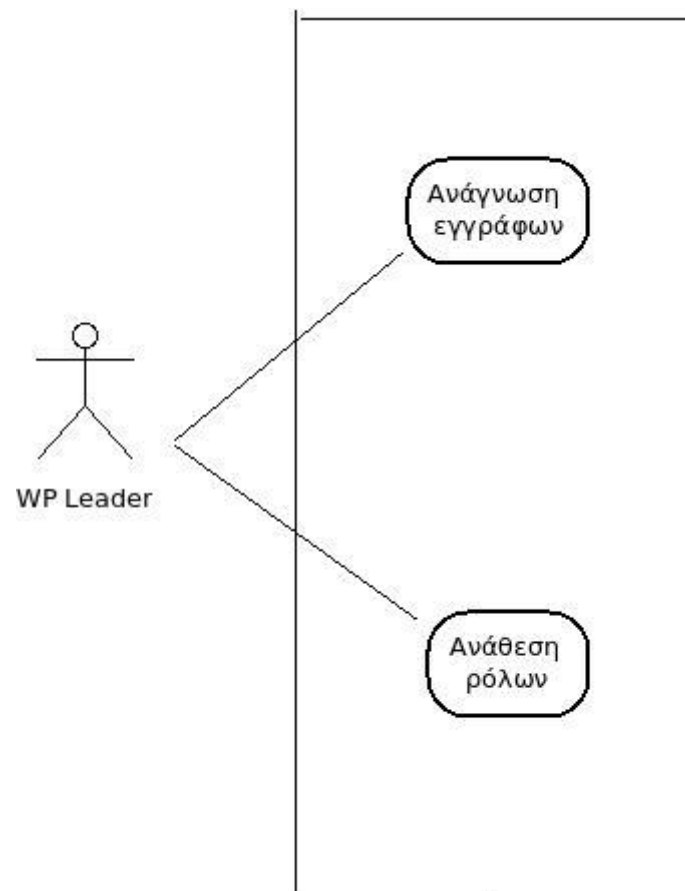
Ο συντονιστής έργων (coordinator) αναθέτει σε ένα συγκεκριμένο έργο (για την ακρίβεια σε ένα φάκελο γνωρίζοντας το id του φακέλου) έναν υπεύθυνο, των διαχειριστή πακέτων. Η ανάθεση γίνεται είτε μέσω της λίστας από “users” είτε από αναζήτηση. Επίσης, ο συντονιστής έργων έχει την δυνατότητα να αλλάξει τον υπεύθυνο (διαχειριστή πακέτων) σε ένα έργο που ήδη υλοποιείται.



Σχήμα 3.3. Role Coordinator

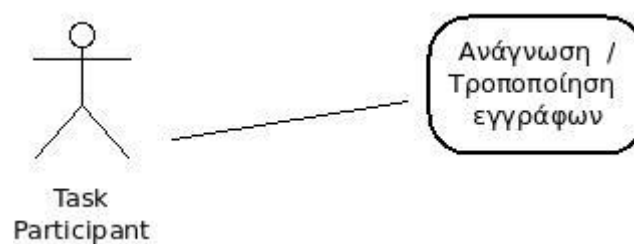
Ο διαχειριστής πακέτων (workpackage manager) αναθέτει στα έργα όπου είναι υπεύθυνος τους συμμετέχοντες σ' αυτά τους οποίους ο ρόλος στο σύστημα θα είναι “task participant” (συμμετέχοντες στο έργο), όπως επίσης, και να διαγράφει χρήστες από την συμμετοχή στο έργο. Ο διαχειριστής πακέτων επιβλέπει πως προχωράει το έργο και έχει την δυνατότητα να βλέπει τα περιεχόμενα των text αρχείων όχι όμως να τα τροποποιεί. Σε ένα έργο

υπάρχει ένας διαχειριστής πακέτων και όσους επιθυμεί συμμετέχοντες.



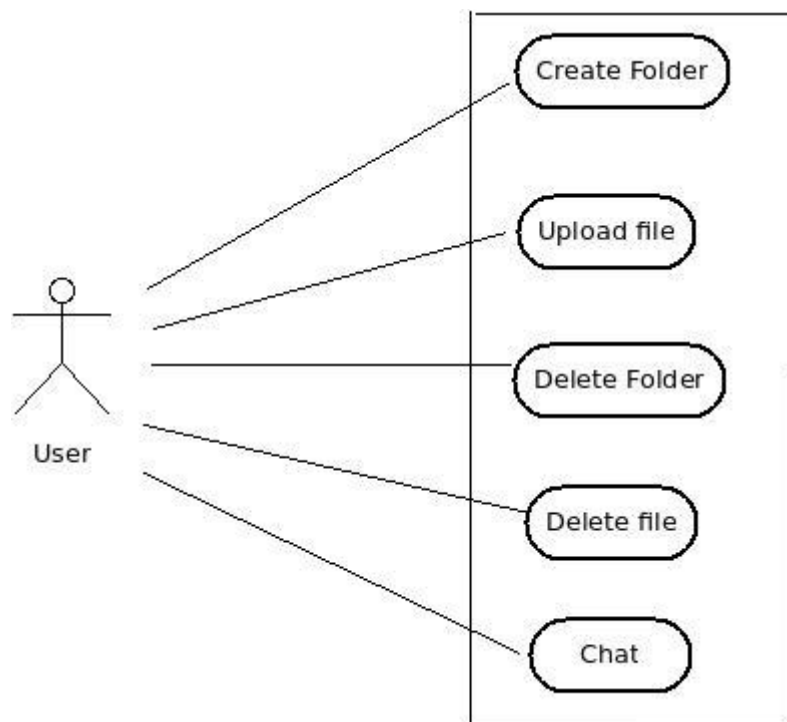
Σχήμα 3.4. Role Work package leader

Οι task participants έχουν την δυνατότητα να τροποποιούν τα text αρχεία σε φακέλους που τους δόθηκε η πρόσβαση.



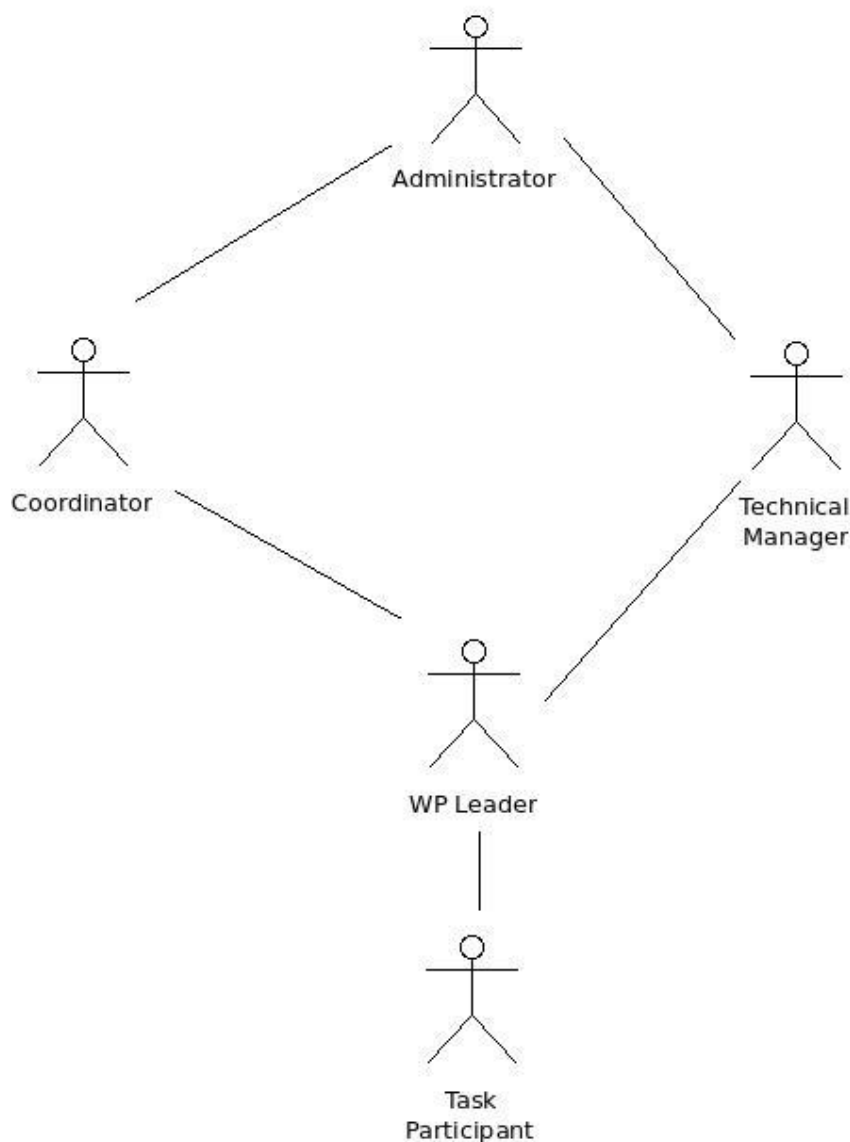
Σχήμα 3.5. Task Participant

Κανένας δεν μπορεί να διαγράψει αρχείο ή φάκελο ή να προσθέσει εκτός από το χρήστη. Επιπλέον ένα project είναι ένα συγκεκριμένο id οπότε τα id των folders βλέπουν μόνο οι owners για τους υπόλοιπους είναι N/A (not available).



Σχήμα 3.6. Role user

Παρακάτω παρουσιάζεται η ιεραρχία των ρόλων:



Σχήμα 3.7. Ιεραρχία των ρόλων

3.2 Λειτουργικότητες

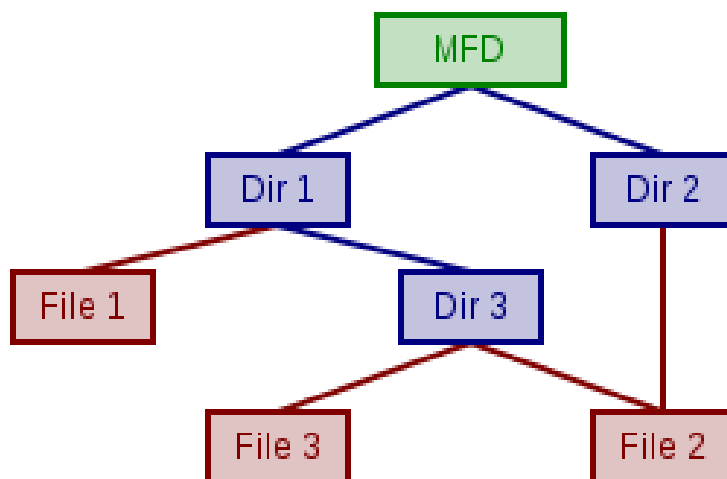
3.2.1 Διαχείριση αρχείων και φακέλων

3.2.1.1 Φάκελοι

Η δημιουργία φακέλων και εικονική αποθήκευση αρχείων σε αυτά (όπως και η διαγραφή τους) σε ένα σύστημα διαχείρισης εγγράφων κάνουν εύκολη την διαχείριση των αρχείων όπως σε έναν προσωπικό υπολογιστή. Πως, όμως, υλοποιείται ένα τέτοιο σύστημα; Με ποια κριτήρια πρέπει να είναι υλοποιημένο το κομμάτι της διαχείρισης αρχείων και φακέλων ώστε να έχει ισχυρό έλεγχο πάνω στην διαδικασία της παραγωγής, αποθήκευσης και διανομής των εγγράφων, με αποτέλεσμα την αποδοτικότερη

επαναχρησιμοποίηση της πληροφορίας;

Οι φάκελοι ή αλλιώς κατάλογοι περιέχουν πληροφορίες για άλλα αρχεία και καταλόγους. Οι πληροφορίες αυτές είναι το όνομα, μέγεθος, τοποθεσία στην οποία βρίσκεται σε κάποιο δευτερεύον αποθηκευτικό μέσο, ημερομηνία δημιουργίας, δικαιώματα προσπέλασης, ιδιοκτήτης κτλ. Ένας κατάλογος που βρίσκεται καταχωρημένος σε έναν κατάλογο ονομάζεται υποκατάλογός του. Έτσι σχηματίζεται μια ιεραρχία καταλόγων, μια δενδροειδής δομή.



Σχήμα 3.8. Diagram of a hierarchal directory tree.

Όταν δημιουργείται ένας φάκελος αυτό θα έχει ένα γνώρισμα, το id του (idFolder), το οποίο θα είναι μοναδικό, δηλαδή δεν μπορεί να υπάρχουν δύο ή παραπάνω φάκελοι με το ίδιο id, και μαζί με τα άλλα χαρακτηριστικά του (όνομα του φακέλου, ιδιοκτήτης, parentId κ.τ.λ.) αποθηκεύονται στην βάση δεδομένων. Το επόμενο πρόβλημα που παρουσιάζεται είναι σε ποια θέση στην ιεραρχία καταλόγων θα ανήκει το συγκεκριμένο φάκελο που μόλις δημιουργήθηκε. Ένα από τα γνωρίσματα που αποθηκεύονται στην βάση δεδομένων κατά την δημιουργία του φακέλου είναι το parentId. Το parentId δείχνει το id του φακέλου στον οποίο ανήκει, ή αλλιώς το parentId είναι το id του “γονιέα”. Κάθε φορά που ο χρήστης κάνει κλικ σε κάποιο φάκελο γίνεται η αναζήτηση στην βάση δεδομένων για όλους τους καταλόγους που έχουν parentId το id του φακέλου.

Λίγο πιο περίπλοκη κατάσταση επικρατεί με την διαγραφή του καταλόγου. Διαγράφοντας καταλόγους με βάση το id τους πρέπει να εξασφαλίσουμε και την διαγραφή των υποκαταλόγων και αρχείων που περιέχουν, σε διαφορετική περίπτωση ναι μεν ποτέ δεν θα έχουμε πρόσβαση σε υποκαταλόγους και τα αρχεία που περιέχει ο φάκελος που μόλις διαγράψαμε όμως αυτά μένουν στην βάση, έτσι η βάση δεδομένων γεμίζει με άχρηστα πλέον δεδομένα. Έτσι,

για την επίλυση του προβλήματος αποφασίστηκε κάθε κατάλογος να περιέχει την διαδρομή από id των καταλόγων ξεκινώντας από το αρχικό το οποίο είναι το γνώρισμα pathFolderId.

3.2.1.2 Αρχεία

Ένα αρχείο υπολογιστή μπορεί να θεωρηθεί σαν το σύγχρονο ισοδύναμο ενός εγγράφου το οποία παραδοσιακά βρίσκονταν σε αρχειοθήκες γραφείων και βιβλιοθηκών. Αυτή είναι και η προέλευση της έννοιας.

Σε κάθε αρχείο ενός υπολογιστή δίνεται ένα όνομα ώστε να ξεχωρίζει από όλα τα άλλα αρχεία.

Όταν γίνεται ανέβασμα ενός εγγράφου αυτό αποθηκεύεται στην βάση δεδομένων με την μορφή BLOB, όπως και τα χαρακτηριστικά του, όπως το id, όνομα, ιδιοκτήτης, ημερομηνία/ημερομηνία ανανέωσης κ.τ.λ. . Το id του κάθε εγγράφου είναι μοναδικό, δεν μπορούν δύο ή παραπάνω έγγραφα να έχουν το ίδιο id. Έτσι, κάθε φορά που ο χρήστης επιθυμεί την διαγραφή του εγγράφου ή κατέβασμα του εγγράφου στο τοπικό δίσκο στο προσωπικό του υπολογιστή αυτό γίνεται με βάση το id του εγγράφου.

Το ανέβασμα του αρχείου, όπως αναφέρθηκε προηγουμένως, γίνεται με Blob (binary large object, basic large object), μια συλλογή από δυαδικά δεδομένα που αποθηκεύονται ως μια ενιαία οντότητα σε ένα σύστημα διαχείρισης βάσεων δεδομένων. Τα αρχεία μπορεί να είναι ένα εγγράψιμο αρχείο, εικόνες, ήχος ή άλλα αντικείμενα πολυμέσων, ακόμη και εκτελέσιμος κώδικας.

Annotation (υποσημείωση) αρχείων (έκδοση, σχόλια, ημερομηνία αποθήκευσης, ημερομηνία τροποποίησης, ιδιοκτήτης) είναι ζήτημα πραγματικά ζωτικής σημασίας για όσους χρειάζονται να διατρέχουν πολλά αρχεία. Ο κάθε χρήστης μπορεί να αφήνει σχολείο για κάθε έγγραφο που ανεβάζει και γνωρίζει τότε έγινε αποθήκευση ή τροποποίηση του αρχείου.

3.2.1.3 Διαχείριση δικαιωμάτων αρχείων

Με την υπηρεσία Διαχείρισης δικαιωμάτων πληροφοριών δίνεται η δυνατότητα σε άτομα και διαχειριστές να καθορίσουν τα δικαιώματα σε έγγραφα. Κάποιες φορές καθιστά ανάγκη για διαχείριση κάποιου φακέλου μαζί με τα αρχεία που περιέχει από ένα δεύτερο χρήστη ή μια ομάδα χρηστών.

Παρακάτω παρουσιάζεται ο πίνακας με τα δικαιώματα πρόσβασης σε αρχεία/φακέλους των άλλων χρηστών (εφόσον το επιθυμούν οι ίδιοι χρήστες

του συστήματος), με βάση τους ρόλους και των λειτουργιών που αντιστοιχούν σε κάθε ρόλο που περιεγράφηκαν προηγουμένως:

Role	Read	Write	Download	Delete
Admin	X	X	X	X
Project coordinator	X	X	X	X
Technical manager	X	X	X	X
Work package leader	✓	X	✓	X
Task participant	✓	✓	X	✓
Owner	X	X	✓	✓

3.2.1.4 Έλεγχος έκδοσης (rollback options)

Έλεγχος έκδοσης είναι διαχείριση των αλλαγών στα έγγραφα. Οι αλλαγές, συνήθως, προσδιορίζονται με έναν αριθμό που ονομάζεται “αριθμός αναθεώρησης” ή “επίπεδο αναθεώρησης” ή απλά “αναθεώρηση”. Για παράδειγμα, ένα αρχικό αρχείο “έκδοση 1.0”, το επόμενο αρχείο με το ίδιο όνομα που γίνεται αποθήκευση ή τροποποιείται, είναι “έκδοση 2.0” κ.ο.κ. Κάθε αναθεώρηση συνδέεται με μια χρονική σφραγίδα και το άτομο που κάνει την αλλαγή.

Η ανάγκη για έναν λογικό τρόπο οργάνωσης και τον έλεγχο αναθεώρησες υπήρχε σχεδόν εφ' όλη την διάρκεια ύπαρξης των εγγράφων αλλά ο έλεγχος αναθεώρησης έγινε πολύ πιο σημαντικός και πολύπλοκος, όταν άρχισε η εποχή της πληροφορικής. Σήμερα, τα πιο ικανά (καθώς και πολύπλοκα) συστήματα ελέγχου αναθεώρησης είναι εκείνα που χρησιμοποιούνται στην ανάπτυξη λογισμικού, όπου μια ομάδα χρηστών μπορούν να αλλάξουν τα ίδια αρχεία.

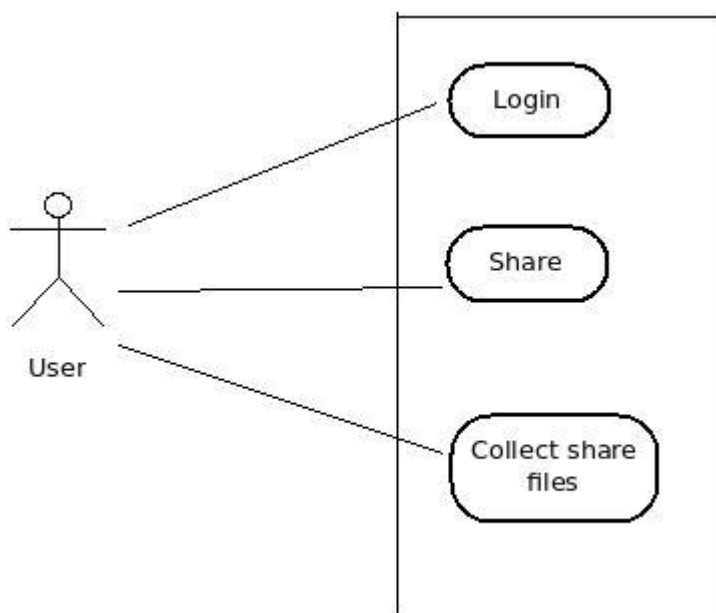
3.2.2 Διαμοιρασμός Αρχείων

3.2.2.1 Εισαγωγή

Η κοινή χρήση αρχείων είναι η πρακτική της παροχής πρόσβασης σε ψηφιακά μέσα, όπως τα προγράμματα υπολογιστών, πολυμέσων (ήχου, εικόνων και βίντεο), εγγράφων ή ηλεκτρονικών βιβλίων. Ανταλλαγή αρχείων μπορεί να επιτευχθεί με διάφορους τρόπους. Κοινές μέθοδοι αποθήκευσης, μεταφοράς και διασποράς χρησιμοποιούν αναιρούμενα μέσα, κεντρικούς εξυπηρετητές σε δίκτυα υπολογιστών, World Wide Web-based υπερσύνδεσης και τη χρήση των κατανεμημένων peer-to-peer δικτύων. Οι χρήστες μπορούν να χρησιμοποιήσουν το λογισμικό που συνδέεται σε ένα δίκτυο peer-to-peer για την αναζήτηση για τα κοινόχρηστα αρχεία στους υπολογιστές των άλλων χρηστών που είναι συνδεδεμένοι στο δίκτυο. Τα αρχεία που ενδιαφέρουν μπορεί στη συνέχεια να τα κατεβάζονται απευθείας από άλλους χρήστες στο δίκτυο. Συνήθως, τα μεγάλα αρχεία κατανέμονται σε μικρότερα κομμάτια, τα οποία μπορούν να ληφθούν από πολλούς χρήστες (ομότιμους/peers) και στη συνέχεια να συγκεντρωθούν από το downloader.

3.2.2.2 Διαμοιρασμός αρχείων για το Σύστημα Διαχείρισης Εγγράφων που υλοποιήθηκε

Ο κάθε χρήστης όταν κάνει ανέβασμα ενός εγγράφου έχει την επιλογή να είναι το έγγραφο “public for all” και αυτό αποθηκεύεται στον φάκελο Public. Ο φάκελος Public περιέχει όλα τα αρχεία που είναι για κοινή χρήση με τους άλλους χρήστες. Κάθε αρχείο μέσα στο Public έχει μια επιλογή “go to URL” με την οποία αποκτά μια διεύθυνση URL από την οποία οι χρήστες ελεύθερα κατεβάζουν το συγκεκριμένο αρχείο.



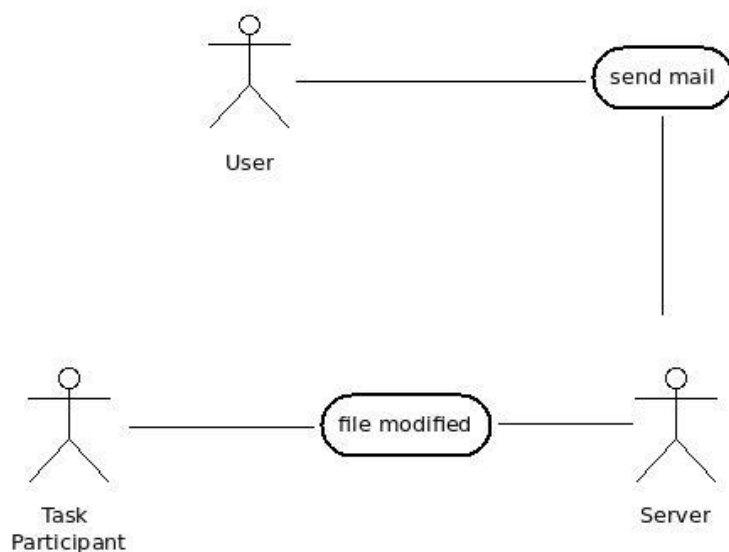
Έγινε σκόπιμα οι χρήστες να είναι εγγεγραμμένοι στο σύστημα ώστε να μπορούν να κατεβάζουν τα αρχεία προς κοινή χρήση.

3.2.3 Επιλογές εγγραφής για αποστολή ειδοποιήσεων

Ένα σύγχρονο σύστημα κοινοποίησης είναι ένας συνδυασμός λογισμικού και υλικού που παρέχει ένα μέσο για την παράδοση ενός μηνύματος σε ένα σύνολο παραληπτών. Για παράδειγμα, ένα σύστημα κοινοποίησης μπορεί να στείλει ένα e-mail όταν ένα αντικείμενο έχει αλλάξει στη Wikipedia. Η πολυπλοκότητα του συστήματος κοινοποίησης είναι συχνά εξαρτάται από το είδος των μηνυμάτων που αποστέλλονται. Προηγμένα συστήματα κοινοποίησης υποστηρίζουν τουλάχιστον ένα και μερικές φορές όλα από τα ακόλουθα μέσα επικοινωνίας:

- Text messaging (SMS)
- E-mail (POP, IMAP, SMTP)
- Desktop alert (dialog, balloon, modal window, toast)
- Pager (SNPP)
- Instant messaging (IRC, ICQ, AIM, Yahoo! Messenger, MSN, XMPP, iMessage)
- RSS (RSS reader, digital signage)
- Web page (Javascript, XML)
- Fax

Στο σύστημά μας ο χρήστης ειδοποιείται μέσω e-mail για κάθε αλλαγή/τροποποίηση αρχείου όταν αυτό γίνεται από άλλους χρήστες που έχουν τα δικαιώματα πρόσβασης στους συγκεκριμένους φακέλους και αρχεία του χρήστη. Κάθε τέτοιο μήνυμα περιλαμβάνει το όνομα του αρχείου που άλλαξε/τροποποιήθηκε και την διαδρομή του.

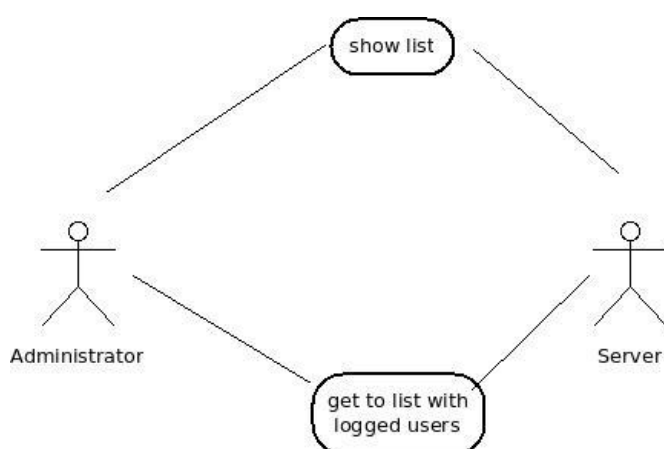


Σχήμα 3.10. Notifications

3.2.4 Πρόσβαση χρηστών.

Μια σύνδεση αναφέρεται στις πιστοποιήσεις που απαιτούνται για να αποκτήσουν οι χρήστες πρόσβαση σε ένα σύστημα διαδικτυακών υπηρεσιών όπου η πρόσβαση είναι περιορισμένοι. Πρόσβαση σε ένα σύστημα υπολογιστή ελέγχεται με τον εντοπισμό και την αναγνώριση του χρήστη μέσα από τις πιστοποιήσεις που παρουσιάζονται από το χρήστη.

Σύνδεση, συνήθως, χρησιμοποιείτε για την πρόσβαση σε μία σελίδα όπου οι υπόλοιποι χρήστες του διαδικτύου δεν μπορούν να δουν. Μόλις συνδεθεί ο χρήστης η σύνδεση αυτή μπορεί να χρησιμοποιηθεί για να παρακολουθούνται οι ενέργειές του όσο είναι συνδεδεμένος.



Σχήμα 3.11. Logged in χρήστες

3.2.5 Chat

Online chat μπορεί να αναφέρεται σε οποιοδήποτε είδος επικοινωνίας μέσω του Διαδικτύου που προσφέρει σε πραγματικό χρόνο μετάδοση των μηνυμάτων κειμένου από τον αποστολέα στο δέκτη. Συνήθως, τα chat μηνύματα είναι μικρά ώστε να επιτρέψουν σε άλλους παράγοντες να ανταποκριθούν γρήγορα. Με αυτό τον τρόπο δημιουργείται ένα αίσθημα παρόμοιο με μια προφορική συζήτηση, το οποίο διακρίνει την συνομιλία από τις υπόλοιπες ηλεκτρονικές φόρμες επικοινωνίας, όπως ηλεκτρονικό ταχυδρομείο, συζήτηση στο forum κ.τ.λ. Online chat μπορεί να απευθύνεται σε δύο ή μια ομάδα ατόμων και τα μηνύματα μπορεί να διακρίνονται σε κείμενο, φωνή, video κλήση.

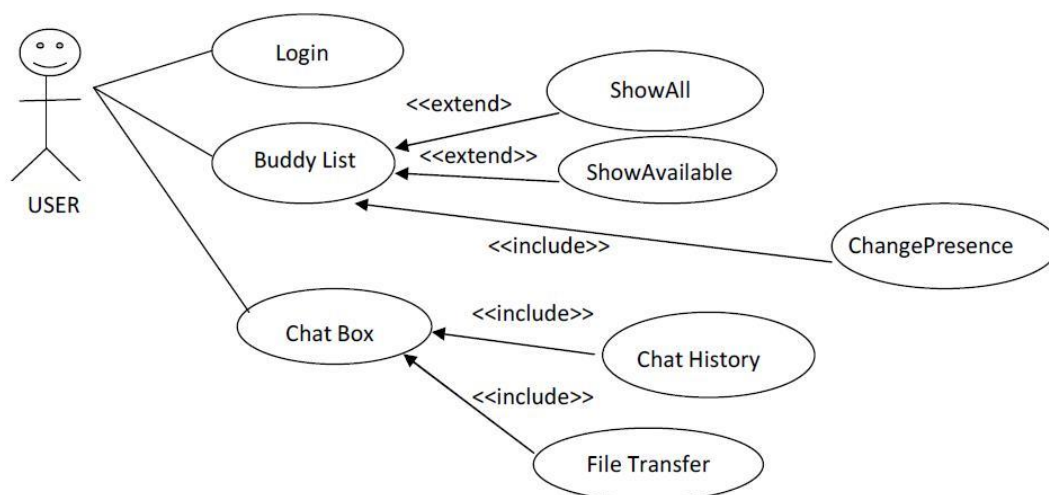
Το πρώτο online σύστημα συνομιλίας ονομαζόταν Talkomatic, που δημιουργήθηκε από Doug Brown και ο David R. Woolley το 1973 στο Πανεπιστήμιο του Ιλινόις. Προσέφερε πολλά κανάλια, καθένα από τα οποία θα μπορούσε να φιλοξενήσει μέχρι και πέντε άτομα, με μηνύματα που εμφανίζονται στις οθόνες όλων των χρηστών χαρακτήρα προς χαρακτήρα,

όπως είχαν πληκτρολογήσει. Talkomatic ήταν πολύ δημοφιλής μεταξύ των χρηστών μέχρι τα μέσα της δεκαετίας του 1980. Το 2014 Brown και Woolley κυκλοφόρησε μια web-based έκδοση του Talkomatic.

Κατά παράδοση, τα chatrooms ήταν δημοφιλή στους νέους ως τρόπος δημιουργίας νέων φίλων, "χώρος" όπου μοιράζονται με άλλους τα χόμπι και τα ενδιαφέροντά τους και συζητούσαν για ζητήματα που τους ήταν προσφιλή. Τα τελευταία χρόνια, εντούτοις, οι ιστότοποι κοινωνικής δικτύωσης τα΄χουν αντικαταστήσει ως κύρια πλατφόρμα για τη συνομιλία των εφήβων στο διαδίκτυο.

Τα άμεσα μηνύματα παραμένουν δημοφιλή καθώς επιτρέπουν στα παιδιά και τους εφήβους να συνομιλούν σε πραγματικό χρόνο με φίλους και με την οικογένειά τους, από τον υπολογιστή τους και άλλες συσκευές με δυνατότητα διαδικτύου, όπως κινητά τηλέφωνα για παράδειγμα.

Αναγνωρίζοντας ότι οι νέοι, στα chat rooms, θα μπορούσαν να είναι εν γένει ευάλωτοι, πολλοί από τους κύριους παρόχους υπηρεσιών διαδικτύου και κινητής τηλεφωνίας επιτρέπουν στους κάτω των 18 ετών την πρόσβαση μόνο σε ελεγχόμενα chatrooms. Αυτά έχουν κανονισμούς που βοηθούν στην προστασία των νεότερων χρηστών, και παρακολουθούνται διαρκώς από επαγγελματίες συντονιστές που έχουν το νου τους μήπως μια συμπεριφορά είναι ενοχλητική και ανάρμοστη.

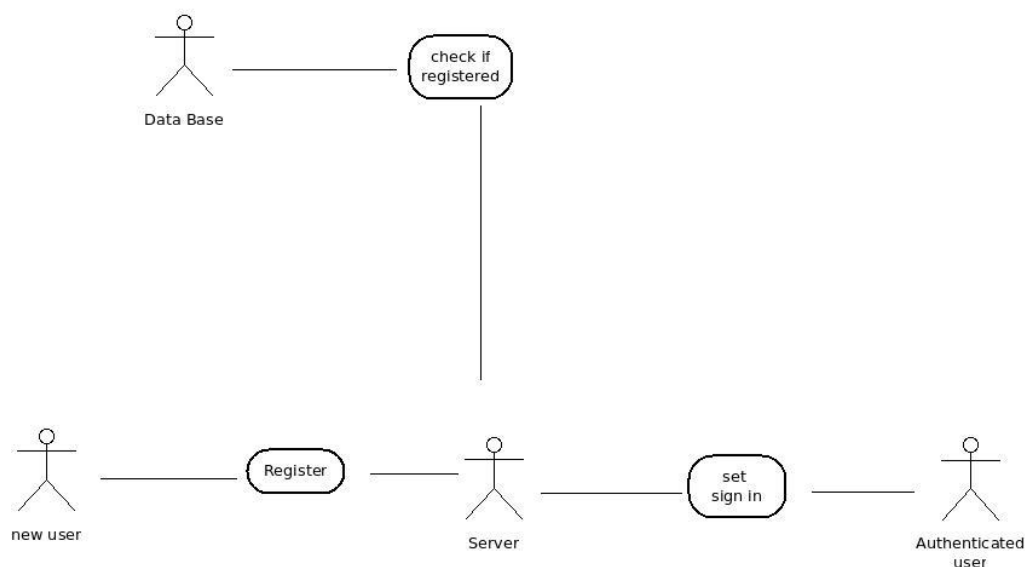


Σχήμα 3.12 Chat

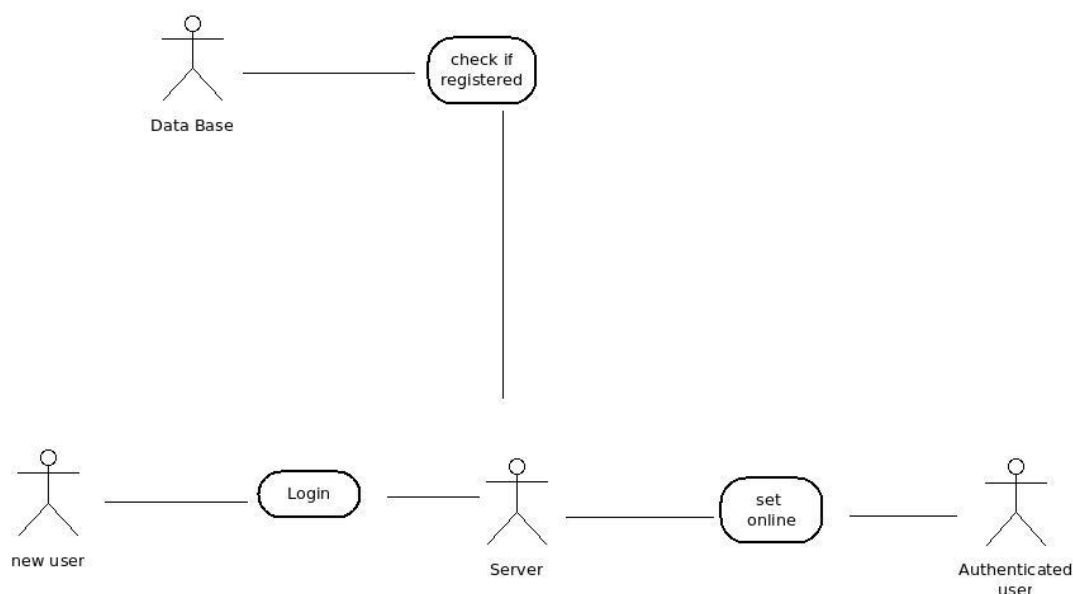
3.2.6 Ανάλυση των Απαιτήσεων Ασφάλειας

3.2.6.1 Authentication-Πιστοποίηση

Με την έννοια “Πιστοποίηση” αναφερόμαστε στην διαδικασία με την οποία καθορίζεται ποιος έχει την πρόσβαση στο LAN. Μέθοδοι έγκρισης χρησιμοποιούν μια βάση δεδομένων που αποτελείται από το όνομα χρήστη και τον κωδικό πρόσβασης. Έτσι εάν το όνομα του χρήστη και ο κωδικός ταιριάζουν με αυτά που είναι αποθηκευμένα στην βάση δεδομένων τότε ο χρήστης θα έχει την πρόσβαση στο δίκτυο, σε διαφορετική περίπτωση η πρόσβαση αποτυγχάνει.



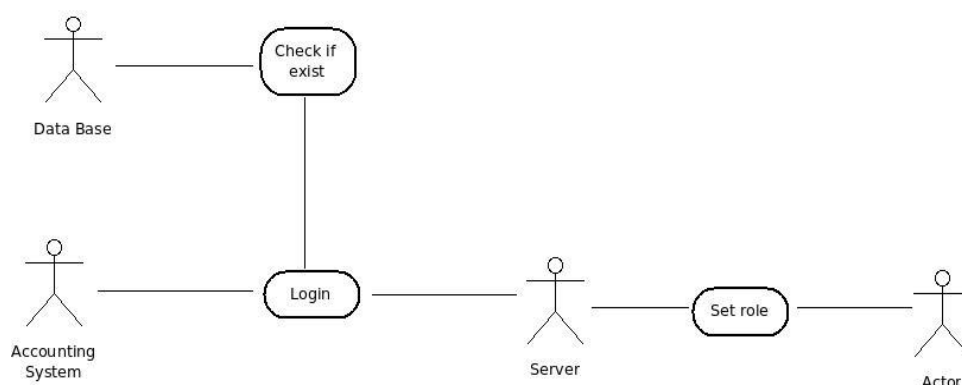
Σχήμα 3.13. Σενάριο πρώτης εγγραφής



Σχήμα 3.14. Σενάριο σύνδεσης υφιστάμενου χρήστη

3.2.6.2 Authorization-Έγκριση

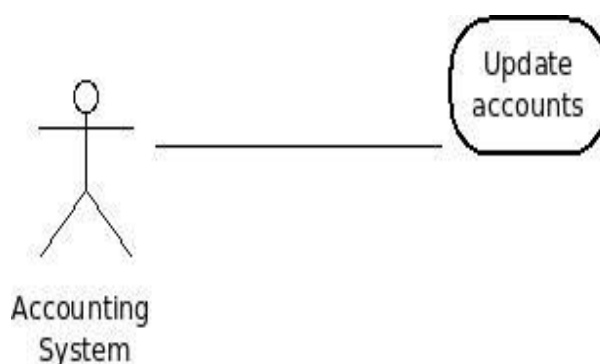
Η Έγκριση είναι η ικανότητα περιορισμού υπηρεσιών του δικτύου με βάση της δυναμικής λίστας πρόσβασης που μερικές φορές ονομάζεται και ως “ρόλος του χρήστη”. Ο κάθε χρήστης έχει δικό του ρόλο. Μετά την πρόσβαση στο σύστημα καθορίζονται ποια είδη δραστηριοτήτων, υπηρεσιών επιτρέπονται στον χρήστη με βάση τον ρόλο που κατέχει.



Σχήμα 3.15 Σενάριο έγκρισης υφιστάμενου χρήστη

3.2.6.3 Accounting- Παρακολούθηση

Με την έννοια “Παρακολούθηση” αναφερόμαστε στην καταγραφή στατιστικών του χρήστη όπως η διάρκεια παραμονής στο σύστημα ή ποσότητα δεδομένων που έχει λάβει/στείλει ο χρήστης κατά την διάρκεια μιας σύνδεσης.



Σχήμα 3.16. Σενάριο παρακολούθησης του χρήστη

3.3 Συμπεράσματα

Από την ανάλυση του συστήματος διαχείρισης εγγράφων προκύπτουν τα ακόλουθα συμπεράσματα:

Ένα σύστημα διαχείρισης εγγράφων αποτελεί βασική επένδυση για κάθε οργάνωση που επιθυμεί να αυξήσει την παραγωγικότητά με αποτέλεσμα την

αποδοτικότερη επαναχρησιμοποίηση της πληροφορίας. Αποτελούν απλά τεχνολογικά εργαλεία, που καθιστούν την διαχείριση των εγγράφων ευκολότερη. Οι χρήστες στέλνουν τα ηλεκτρονικά έγγραφα που θέλουν στην κεντρική ηλεκτρονική αποθήκη και μετά έχουν την δυνατότητα να τα διαχειρίζουν.

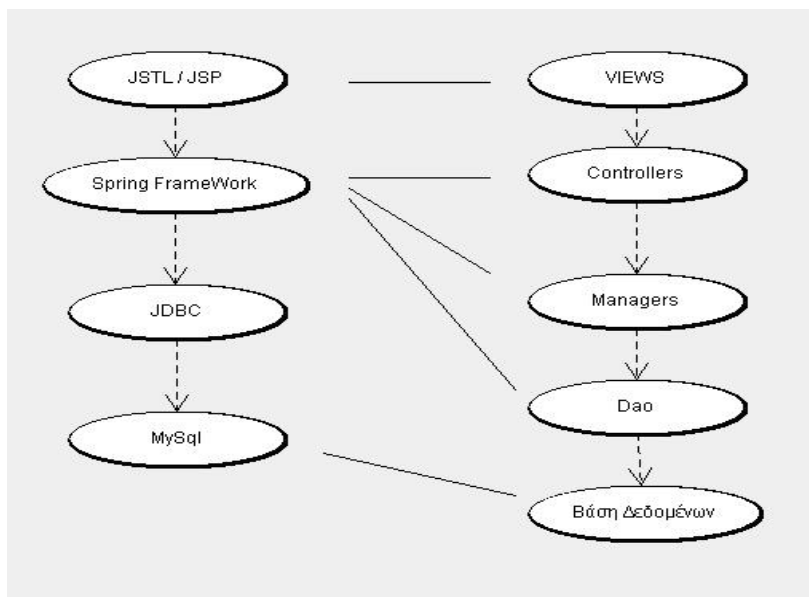
Έλεγχος ταυτότητας, εξουσιοδότησης και παροχή δικαιωμάτων πρόσβασης σε συγκεκριμένες ομάδες δίνει την δυνατότητα σε συνεργασία ανάμεσα σε διαφορετικές εταιρείες. Τα συστήματα διαχείρισης εγγράφων μπορούν να συντελέσουν στην γενική μείωση του κόστους λειτουργίας της επιχείρησης, όπως μειωμένο κόστος αποθήκευσης εγγράφων, αυξημένη παραγωγικότητα, λιγότερα λάθη ανθρώπινου παράγοντα, μικρότερη ανάγκη για επιπλέον προσωπικό. Βελτιωμένη ποιότητα λόγω ταχύτερης εξυπηρέτησης, αξιόπιστη προσφορά υπηρεσιών, προχωρημένες δυνατότητες αναζήτησης εγγράφων. Παρουσιάζουν μεγαλύτερη ασφάλεια έναντι ηλεκτρονικών αρχείων σε εναλλακτικές τοποθεσίες. Αποδίδονται ρόλοι στους υπαλλήλους και με βάση αυτούς γίνεται έλεγχος σε ποια έγγραφα έχει πρόσβαση ο κάθε υπάλληλος και για ποιο λόγο.

4

Τεχνολογίες

Κεφάλαιο 4 - Τεχνολογίες

Στο κεφάλαιο αυτό περιγράφονται οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση των επιμέρους οντοτήτων της αρχιτεκτονικής της εφαρμογής. Όπως φαίνεται και στην παρακάτω εικόνα η τεχνολογία των JavaServer Pages Standard Tag Library (Jstl) χρησιμοποιήθηκε για την υλοποίηση των αρχείων view. Ενώ το Spring Framework χρησιμοποιήθηκε για την υλοποίηση των controllers και των επιπέδων service και repository. Η τεχνολογία που επιτρέπει την επικοινωνία ανάμεσα στις μεθόδους που περιέχει το επίπεδο του repository με την βάση δεδομένων είναι η Java Database Connectivity (JDBC). Η βάση δεδομένων δημιουργήθηκε με MySql.



4.1 Δυνατότητες *Spring Framework*

Το Spring Framework είναι περιβάλλον εργασίας ανοιχτού κώδικα (open source) βασισμένο στον κώδικα που δημοσιεύεται στο «Expert One-on-One J2EE Design and Development» από τον Rob Johnson και σκοπό έχει την διευκόλυνση της ανάπτυξης εφαρμογών Java. Το Spring είναι εμβατικό και έχει διαιρεθεί σε ανεξάρτητα πακέτα, τα οποία μπορούν όμως να λειτουργήσουν ανεξάρτητα με άλλα frameworks προσφέροντας την λειτουργικότητα που υπόσχονται με ελάχιστες απαιτούμενες ρυθμίσεις. Μειώνει την προσπάθεια για την ανάπτυξη μίας εφαρμογής και το κόστος ενώ προσφέρει διευκολύνσεις στη βελτίωση των test δοκιμών και ποιότητας. Παρέχει μηχανισμούς που επιτρέπουν στα beans να υποστηρίζονται από δυναμικές γλώσσες προγραμματισμού.



Οι βασικότερες υπηρεσίες που προσφέρει είναι:

- Inversion of Control container (IoC)
- Aspect-oriented Programming (AOP)
- Data Access
- Transaction Management
- Model-and-View Controller
- Remote Access Framework
- Batch Processing
- Authentication και Authorization
- Remote Management
- Messaging
- Testing

Οι πιο σημαντικές λειτουργίες του Spring Framework είναι η Inversion-of-Control (IoC) και η AOP (aspect oriented programming). Στην πρώτη παρέχεται ένα σύνολο από τεχνικές στις οποίες η ροή ελέγχου αντιστρέφεται σε σχέση με τον τρόπο που γινόταν μέχρι τώρα. Το framework καλεί πλέον τις

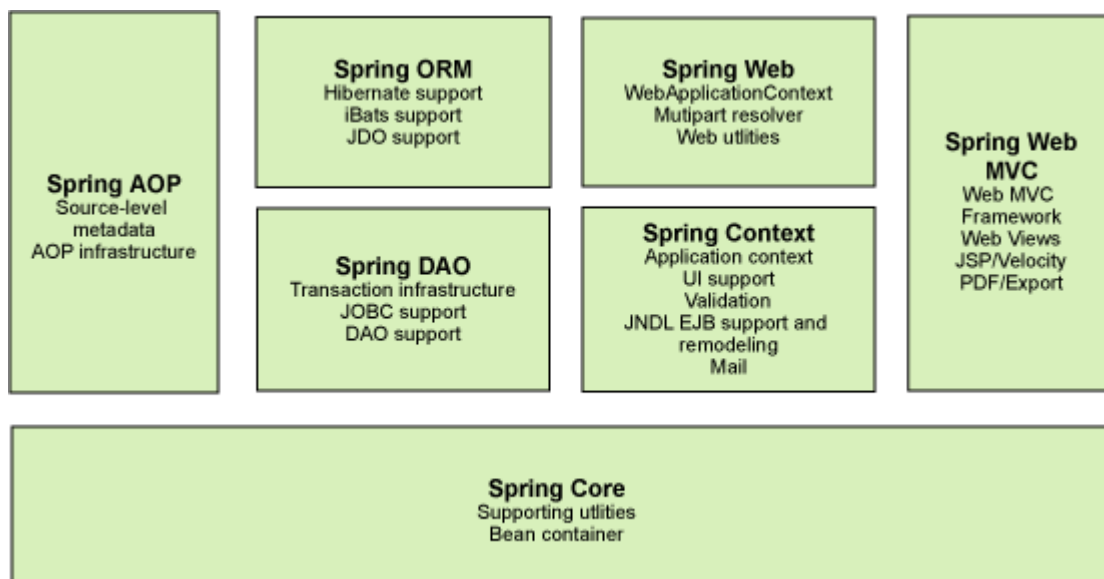
διαδικασίες/μεθόδους που έχουν δημιουργηθεί από τον προγραμματιστή και όχι ο προγραμματιστής τις διαδικασίες/μεθόδους του framework. Η δεύτερη προσφέρει μία ευέλικτη λύση στην επίληψη των cross-cutting concerns, όπως η transaction management.

Το Spring Framework είναι ιδανικό όταν η εφαρμογή μας περιλαμβάνει αντικείμενα των οποίων το implementation αλλάζει συχνά. Τα αντικείμενα αυτά ορίζονται σε ένα αρχείο απομονώνοντας το κώδικα από τις αλλαγές αυτές στην εφαρμογή. Ακόμη, όταν η εφαρμογή μας εφοδιάζει την αρχικοποίηση τιμών μέσω constructors.

Μερικά από τα εργαλεία για το Spring είναι τα SpringSourceTool Suite, SpringSource Application Platform, MyEclipse Spring Tools και πολλά άλλα χρήσιμα εργαλεία που ικανοποιούν τις περισσότερες πλατφόρμες και γλώσσες προγραμματισμού.

4.2 Αρχιτεκτονική του Spring Framework

Η αρχιτεκτονική του Spring Framework όπως φαίνεται και στην εικόνα 3.1 αποτελείται από επτά καλά καθορισμένες ενότητες. Το Spring είναι ένας container που περιέχει και διαχειρίζεται τον κύκλο ζωής και τη διαμόρφωση των αντικειμένων της εφαρμογής. Μπορείτε να διαμορφώσετε τον τρόπο με τον οποίο τα beans σας πρέπει να δημιουργηθούν και πώς να συσχετίζονται μεταξύ τους. Όπως διακρίνεται και στην εικόνα όλες οι ενότητες έχουν τοποθετηθεί πάνω από τον πυρήνα. Ο πυρήνας προσδιορίζει πως έχουν δημιουργηθεί, διαμορφωθεί και διαχειριστεί τα beans.



Οι επτά ενότητες είναι:

-**Core** , που είναι και η πιο σημαντική ενότητα περιλαμβάνει την IoC και Dependency injection. Το βασικότερο “συστατικό” είναι ο BeanFactory και επιτρέπει την διαμόρφωση ενός xml αρχείου το οποίο καθορίζει ποια beans δημιουργούνται και ποια είναι τα χαρακτηριστικά τους. Το πλεονέκτημα του BeanFactory είναι ότι διαχωρίζει το configuration της εφαρμογής και τις προδιαγραφές σχετικές με τις εξαρτήσεις από τον κώδικα της εφαρμογής. Έτσι ο προγραμματιστής μπορεί να ορίσει πως θα διαμορφώνονται τα αντικείμενα και οι μεταξύ τους σχέσεις.

-**Application context**, είναι ένα configuration αρχείο το οποίο παρέχει πληροφορίες στο Spring Framework σχετικά με το περιεχόμενο της εφαρμογής. Το αρχείο αυτό “κληρονομεί” τα χαρακτηριστικά που ορίζονται στην περιγραφή των beans. Περιλαμβάνει υπηρεσίες όπως e-mail, JNDI access, EJB integration, πρόσβαση από απόσταση και σχεδιασμό. Επίσης υποστηρίζει μηνύματα, events για τον κύκλο ζωής της εφαρμογής και επικύρωση.

-**AOP** (Aspect-oriented programming) παρέχει την λειτουργικότητα της aspect-oriented programming και επιτρέπει την επικοινωνία μεταξύ του Spring και άλλων AOP πλαισίων. Αναφέρεται στην δυνατότητα που προσφέρει στον προγραμματιστή να προσθέτει λειτουργίες στη μέθοδο μίας κλάσης χωρίς παρ' όλα αυτά να επηρεαστεί ο πηγαίος κώδικας της κλάσης. Επίσης, επιτρέπει στους προγραμματιστές να διαμορφώσουν τις cross-cutting ανησυχίες.

-**DAO**, το επίπεδο που διαχειρίζεται την πρόσβαση στην βάση δεδομένων αλλά και τον χειρισμό exceptions μεταξύ των βάσεων αποτρέποντας έτσι τον προγραμματιστή να γράφει επιπλέον κώδικα.

-**ORM**, το επίπεδο που συνδέει το Spring με άλλα πλαίσια (π.χ ORM,Hibernate,SQL Maps)

-**Web**, στηρίζει την δημιουργία περιεχομένου στην εφαρμογή που είναι κατάλληλο για web-based εφαρμογές. Διευκολύνει την περάτωση πολλών αιτήσεων και την συσχέτιση παραμέτρων στις αιτήσεις με αντικείμενα.

MVC framework, το Spring εφοδιάζει με ένα δικό του Model/View/Controller (MVC) για να κατασκευάζει web εφαρμογές. Είναι ευέλικτο ως προς την διαμόρφωση του και προσαρμόζεται σε πολλές τεχνολογίες όπως JSP, Velocity, Tiles, iText, and POI.

4.3 Spring MVC

Το Spring Framework περιλαμβάνει και το δικό του πλήρως ανεπτυγμένο πλαίσιο Model/View/Controller (MVC) για την υλοποίηση web εφαρμογών . Αν και όπως αναφέρθηκε παραπάνω το Spring μπορεί να λειτουργεί το ίδιο αποδοτικά με διαφορετικές τεχνολογίες όπως Struts, το προβάδισμα που δίνει το Spring MVC είναι η δυνατότητα του να εκμεταλλεύεται πλήρως το Inversion of Control του container προσφέροντας έτσι μία φανερή διαχώριση της λογικής των Controller και τα αντικείμενα του μοντέλου. Οι τρεις βασικές

οντότητες που το χαρακτηρίζουν είναι ο Controller, το Model και View . Ο Controller χειρίζεται την navigation λογική και αλληλεπιδρά με την υπηρεσία που χειρίζεται την business λογική. Το Model αποτελεί τον συνδετικό κρίκο ανάμεσα στον Controller και το View. Τέλος, το View προσφέρει την ανταπόκριση στην αίτηση του χρήστη. Τα δεδομένα που είναι απαραίτητα για να πλαισιώσουν το οπτικό αποτέλεσμα στον χρήστη δρομολογούνται από το View αλλά παράγονται από τον Controller.

Η βάση του Spring MVC είναι ο DispatcherServlet, ένα αρχείο το οποίο αναθέτει τις εισερχόμενες αιτήσεις στους controllers που είναι υπεύθυνοι για την επεξεργασία τους. Ο DispatcherServlet συμβουλευέται τους handler-mappings του στο ApplicationContext. Το ApplicationContext είναι ένα αρχείο που περιέχει «ζευγάρια» διευθύνσεων και controllers.

Τα ζευγάρια αυτά ανάλογα με τις απαιτήσεις των αιτήσεων περιγράφονται στο ApplicationContext από τους ViewResolvers οι οποίοι μπορεί να είναι παραπάνω του ενός. Οι ViewResolvers αντιστοιχούν (έτσι όπως έχουν διαμορφωθεί) τα ονόματα των σελίδων που απαιτούν οι αιτήσεις στις πραγματικές τους πηγές και δεν είναι τίποτα άλλο από αρχεία JSP που περιέχουν πληροφορίες σχετικά με τα δεδομένα όπως ονόματα μεταβλητών στα οποία πρέπει να αποδώσει ο controller τιμές. Οι πιο σημαντικοί ViewResolvers είναι οι UriBasedViewResolver, ResourceBundleViewResolver και InternalResourceViewResolver οι οποίοι μαζί με τους υπόλοιπους περιγράφονται στο έκτο κεφάλαιο. Οι Controllers είναι υπεύθυνοι για την παραγωγή των δεδομένων που χρειάζονται για την αίτηση.

Μερικοί από τους controllers είναι οι AbstractController, MultiActionController και UriFilenameViewController. Αυτοί μαζί με τους υπόλοιπους controllers περιγράφονται στο έκτο κεφάλαιο. Ο εξ ορισμού handler του Spring MVC είναι ένας απλός controller με μία μόνο μέθοδο, την ModelAndView handleRequest(request, response). Το Spring MVC προσφέρει μία ποικιλία από controllers που ικανοποιούν όλες τις απαιτήσεις της εφαρμογής. Ένα από

τα βασικά πλεονεκτήματα του MVC είναι η συμβατότητα του με πολλές διαφορετικές εχνολογίες ανάμεσα τους οι JSTL, Velocity, FreeMarker, JasperReports, PDF, Excel. Άλλο ένα πλεονέκτημα είναι ότι το MVC εκμεταλλεύεται όλες τις υπηρεσίες του Spring ανάμεσα τους το IN messaging και το validation. Ακόμη, ο business κώδικας μπορεί να επαναχρησιμοποιηθεί απαλείφοντας έτσι την πιθανότητα για περιττή επανάληψη. Απλά χρησιμοποιούμε τα ήδη υπάρχοντα αντικείμενα ως αντικείμενα που περιέχονται στην εντολή. Σε διαφορετική περίπτωση απαιτείται η αναπαράσταση τους προκειμένου να επεκταθεί μια συγκεκριμένη βασική κλάση.

Η υπηρεσία validation που προσφέρει το Spring MVC είναι ένα ακόμη πλεονέκτημα σε σύγκριση με τα υπόλοιπα MVC πλαίσια. Η δυνατότητα του validation να ελέγχει και να αποτρέπει τον χρήστη να δίνει λάθος δεδομένα στη φόρμα επιτρέπει στον προγραμματιστή να μην αναγκάζεται να ορίζει τον τύπο των πεδίων σε μία φόρμα να είναι string και στη συνέχεια να τα μετατρέπει στα αντικείμενα που αποτελούν το μοντέλο. Αντιθέτως, ορίζει εξ ορισμού τα πεδία στις φόρμες στα αντικείμενα του μοντέλου έχοντας σαν αποτέλεσμα την μείωση περιττού κώδικα.

Τέλος, το Spring MVC περιλαμβάνει και την δική του βιβλιοθήκη με εντολές για σύνδεση μεταβλητών με δεδομένα και θέματα. Και εδώ, όπως σχεδόν με όλα τα εργαλεία που προσφέρει το Spring Framework, είναι συμβατό με άλλες τεχνολογίες στη περίπτωση που ο προγραμματιστής δεν θέλει να εκμεταλλευτεί όλα όσα προτείνει το Spring.

4.3.1 Ο «κύκλος ζωής» μίας αίτησης στο Spring MVC

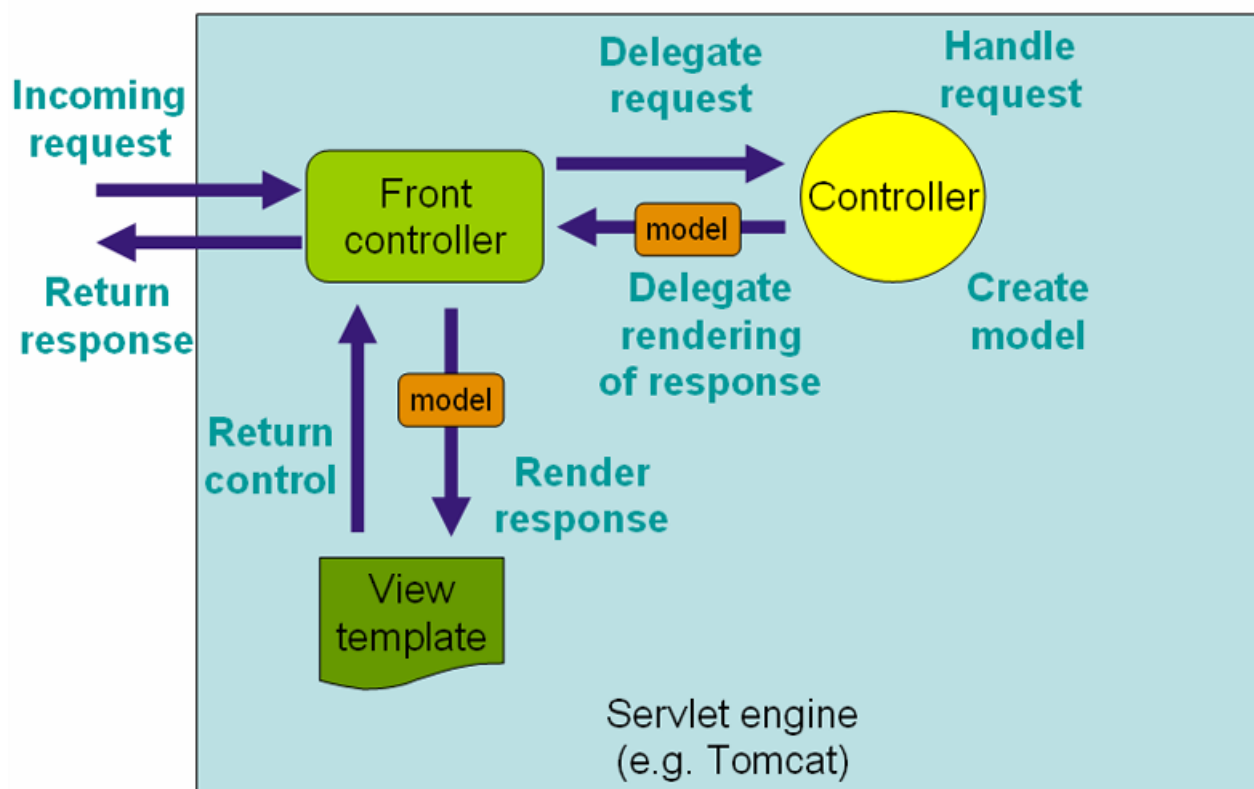
1. Με κάθε καινούργια αίτηση ο browser ζητά μία URL. Η αίτηση αυτή μπορεί να περιλαμβάνει και μεταβλητές.
2. Το DispatcherServlet εξετάζει αρχικά την αίτηση προκειμένου να γίνει η δρομολόγηση προς τον σωστό controller.

3. Συμβουλευτείται τους handler-mappings του που έχουν οριστεί και επιλέγει τον σωστό controller και αναθέτει σε αυτόν να χειριστεί την αίτηση.

4. Ο controller εφαρμόζει την λογική που απαιτείται για να επεξεργαστεί την αίτηση κάτι που έχει σαν επακόλουθο την δημιουργία δεδομένων και πληροφοριών. Τα δεδομένα αυτά αντιστοιχούν σε ονόματα-μεταβλητές το σύνολο των οποίων σχηματίζει ένα ModelAndView αντικείμενο το οποίο μαζί με την αίτηση επιστρέφουν στο DispatcherServlet.

5. Το DispatcherServlet στη συνέχεια συνδέει το όνομα της επιθυμητής url με το ModelAndView αντικείμενο .

6. Τέλος, παραδίδει το μοντέλο και την αίτηση στο view implementation το οποίο με την σειρά του στέλνει το τελικό αποτέλεσμα και απάντηση στην αίτηση του χρήστη στον browser.



4.4 Πλεονεκτήματα του Spring Framework

Το Spring Framework γίνεται όλο και πιο δημοφιλές χάρη στα πλεονεκτήματα που παρέχει σε σχέση με τα υπόλοιπα frameworks. Όπως έχει αναφέρει και ο Rick Hightower , CTO της ARCMIND,INC στο “An Introduction to Spring: For Technical Managers, Developers and Architects” : “Ένα Spring πρότυπο μειώνει τις δυνατότητες των ατελειών που διαρρέουν τους πόρους, και μειώνει τις γραμμές κώδικα. Το Spring παίρνει τον κοινό, επιρρεπή σε λάθη κώδικα και τοποθετεί τη λογική για τη διαχείριση των πόρων και τη δυσνόητη εξαίρεση που χειρίζονται έτσι ώστε να μπορεί να επαναχρησιμοποιηθεί, με το λιγότερο περιττό κώδικα. Το Spring είναι πιο παραγωγικό από την παραδοσιακές J2EE υλοποίηση και επιτρέπει στους προγραμματιστές να είναι περισσότερο παραγωγικοί με πιο ελαφριά πλαίσια που μπορούν να τρέχουν εσωτερικά και εξωτερικά από έναν J2EE container ” .

Μία από τις βασικές λειτουργίες που παρέχει το Spring Framework η AOP συντελεί στο να γίνει το OOP (object-oriented programming) περισσότερο επεκτάσιμο. Ενώ χρησιμοποιεί την λειτουργία dependency injection έχοντας σαν αποτέλεσμα την απλοποίηση στην υλοποίηση και την αύξηση του testable κώδικα. Επίσης προσφέρει μία ξεκάθαρη διαχώριση ανάμεσα στους controllers, τα JavaBeans μοντέλα και το τελικό εμφανισιακό αποτέλεσμα στον χρήστη . Επιπλέον, οι controllers διαμορφώνονται μέσω της λειτουργίας IoC κάτι που επιτρέπει στον προγραμματιστή να τους εξετάσει πιο εύκολα, και να τους ενσωματώσει με άλλα αντικείμενα.

Άλλο ένα βασικό πλεονέκτημα του Spring είναι ότι απλοποιεί την ανάπτυξη ενός project παρέχοντας του εργαλεία και αφηρημένες λειτουργίες που μπορούν να χρησιμοποιηθούν από μία πλειάδα διαφορετικών ιδεολογιών projects. Τα αφηρημένα αυτά επίπεδα που προσφέρει πάνω σε τεχνολογίες (πχ. JDBC και Hibernate) διευκολύνουν την χρήση και οργάνωση της διαχείρισης του configuration. Ακόμη η μη καθοδήγηση του το καθιστά ιδανικό καθώς οποιαδήποτε "σχέση" ή συνάφεια μπορεί να προσαρμοστεί και

παραμετροποιηθεί στις ανάγκες του project. Κατά συνέπεια ο προγραμματιστής μπορεί να επιλέξει ποια εργαλεία ή λειτουργίες από το Spring επιλύουν τα προβλήματα του καλύτερα σε σχέση με τις λύσεις που προτείνουν οι άλλες τεχνολογίες.

Ένα πιο πρακτικό πλεονέκτημα έτσι όπως το τονίζει ο Rob Johnson είναι η προσφορά από το Spring ενός πιο ελαφρύ JavaBean container που απαλείφει την ανάγκη να γραφτεί ο επαναλαμβανόμενος κώδικας «σωληνώσεων» όπως τα lookups. Έτσι αποτρέπεται και η μέχρι τώρα χρήση των EJBs για την επίτευξη λειτουργιών. Επίσης παρέχει ένα framework που επιτρέπει στις εξαρτήσεις μεταξύ των beans να επιλύονται αυτόματα στην διαδικασία αρχικοποίησης των αντικειμένων.

Επιτρέπει σε cross-cutting concerns όπως η transaction management να επιλύονται στα beans ως «πλευρές» παρά να προκαλούν ανησυχίες για το επιχειρησιακό αντικείμενο. Η ευθύνη για την αναζήτηση και δημιουργία εξαρτημένων αντικειμένων δίνεται στο configuration. Ακόμη υποστηρίζει και «ενθαρρύνει» την δημιουργία interface. Τέλος, επιτρέπει και «ενθαρρύνει» την λειτουργία testable code αποτρέποντας έτσι τον προγραμματιστή από λάθη και exceptions.

Σαν framework είναι τόσο ελαφρύ ώστε όλο το πακέτο να περιορίζεται σε ένα μόνο αρχείο (JAR) το οποίο μόλις που ξεπερνά το 1 MB.

4.5 JavaServer Pages Standard Tag Library (Jstl)

Η JavaServer Pages Standard Tag Library (Jstl) είναι ένα από τα ουσιαστικά συστατικά της Java EE Web application τεχνολογία που προσφέρει στον προγραμματιστή μια πλειάδα εντολών που σκοπεύουν στην εμφώλευση λειτουργιών χωρίς την τοποθέτηση κώδικα Java . Επιτρέπει στον προγραμματιστή να χρησιμοποιεί ετικέτες (tags) αντί για scriptlet κώδικα για την δυναμική δημιουργία του περιεχομένου μίας σελίδας. Περιλαμβάνει τέσσερις συνήθεις tag βιβλιοθήκες με στόχο την υποστήριξη απλών και δομικών στόχων. Η πρώτη βιβλιοθήκη με όνομα Core Tag Library περιέχει

ετικέτες που αναλαμβάνουν να εκτελέσουν λειτουργίες όπως επαναλήψεις, εκχωρήσεις τιμών σε μεταβλητές, εφαρμογή υπό όρους περιεχομένου και βασικές εντολές εισόδου και εξόδου. Ακόμη αλληλεπιδρά και με διευθύνσεις URL όπως την εισαγωγή περιεχομένου από διαφορετική url ή αρχείου, την κωδικοποίηση μίας διεύθυνσης εμφωλεύοντας παραμέτρους και την προώθηση σε άλλη διεύθυνση.

Η δεύτερη βιβλιοθήκη με όνομα Formatting/Internationalization Tag Library περιέχει ετικέτες που διεθνοποιούν τα περιεχόμενα μίας αίτησης. Ανάλογα με την χώρα από την οποία προέρχεται ο χρήστης και την γλώσσα αυτής ή τη διαφορά ώρας που έχει διαμορφώνονται και τα περιεχόμενα της σελίδας. Οι περισσότερες ετικέτες επιτρέπουν την διαμόρφωση της εμφάνισης των δεδομένων όπως για παράδειγμα τον τρόπο με τον οποίο θα εμφανίζεται η ημερομηνία ή ο χρόνος ή οι αριθμοί. Επιτρέπει ακόμη και την αντιστοίχιση λέξεων ή κειμένου με μία κωδική ονομασία και σε συνδυασμό με ετικέτες της βιβλιοθήκης core ο προγραμματιστής εμφανίζει λέξεις ή κείμενα με την χρήση μίας λέξης.

Ακόμη η Τρίτη βιβλιοθήκη με όνομα Database Tag Library περιέχει ετικέτες που σχετίζονται με εντολές με SQL και επιτρέπουν την άμεση ανάκτηση και επεξεργασία δεδομένων που βρίσκονται αποθηκευμένα σε μία βάση δεδομένων.

Τέλος, η τέταρτη βιβλιοθήκη με όνομα XML Tag Library περιέχει ετικέτες για το την πρόσβαση σε στοιχεία ενός XML αρχείου ακόμη και την επέμβαση στο περιεχόμενο ενός εγγράφου XML. Επιπλέον ετικέτες επιτρέπουν την μετατροπή ενός αρχείου XML σε XSL. Δεδομένου ότι η γλώσσα XML χρησιμοποιείται κατά κόρον στην web εφαρμογές η βιβλιοθήκη αυτή θεωρείται πολύ χρήσιμη.

4.6 *Hibernate ORM*

Οι περισσότερες εφαρμογές πρέπει να χρησιμοποιήσουν κάποιου είδους

δεδομένα. Παρά το γεγονός ότι το πρόβλημα της διατήρησής τους δεν αποτελεί κάποιο καινούριο ή ασυνήθιστο θέμα για τις εφαρμογές που είναι γραμμένες σε Java ο προγραμματιστής δεν έχει την δυνατότητα απλά να αποφασίσει ποιο δρόμο να ακολουθήσει επιλέγοντας από μια γκάμα παρόμοιων και ευρέως χρησιμοποιούμενων λύσεων.

Για πολλά χρόνια η διατήρηση δεδομένων (persistence) υπήρξε θέμα συζήτησης στην κοινότητα των προγραμματιστών της Java. Πολλοί προγραμματιστές δε συμφωνούν ούτε καν με το θέμα του προβλήματος. Είναι η διατήρηση δεδομένων ένα πρόβλημα που έχει ήδη λυθεί από τη σχεσιακή τεχνολογία (relational technology) και επεκτάσεις (extensions) αυτής, όπως οι αποθηκευμένες διαδικασίες (stored procedures), ή αποτελεί ένα πιο σημαντικό πρόβλημα το οποίο χρειάζεται διαχείριση μέσω ειδικών μοντέλων όπως τα Enterprise Java Beans (EJB). Πρέπει να χρησιμοποιείται SQL κώδικας ακόμη και για τις πιο κλασικές ενέργειες όπως create, read, update, delete ή πρέπει αυτές να γίνονται αυτόματα. Πώς επιτυγχάνεται μεταφερσιμότητα, όταν κάθε σύστημα διαχείρισης βάσεων δεδομένων χρησιμοποιεί τη δική του διάλεκτο. Πρέπει η SQL να εγκαταλειφθεί τελείως και να υιοθετηθεί ένα διαφορετικό σύστημα διαχείρισης βάσεων δεδομένων. Η συζήτηση συνεχίζεται. Παρό όλα αυτά, για την επίλυση του προβλήματος έχει αναπτυχθεί μία προγραμματιστική τεχνική γνωστή ως 'Αντιστοίχιση μεταξύ Δεδομένου - Σχέσης' — object/relational mapping (ORM) — η οποία έχει αποκτήσει ευρεία αποδοχή.

Διατήρηση δεδομένων στην Java

Για την διατήρηση των δεδομένων (persistence) στην Java, στα πλαίσια των Enterprise Java Beans 3, έχει οριστεί μία τεχνολογία που ονομάζεται Java Persistence API. Η τεχνολογία αυτή ορίζει ένα περιβάλλον εργασίας (framework) για την αντιστοίχιση οντοκεντρικών μοντέλων (object - oriented models) σε σχήματα παραδοσιακών σχεσιακών βάσεων δεδομένων. Κεντρικός στόχος για την ανάπτυξή της υπήρξε ο ορισμός ενός ενιαίου τρόπου για την διατήρηση δεδομένων σε όλες τις Java εφαρμογές.

Η Hibernate αποτελεί μία βιβλιοθήκη ελεύθερου λογισμικού για τη γλώσσα

προγραμματισμού Java, η οποία υλοποιεί την διεπαφή που ορίζεται στα πλαίσια του Java Persistence API. Είναι μία ολοκληρωμένη λύση στο πρόβλημα της διαχείρισης δεδομένων εκτελώντας όλες τις ενέργειες μεταξύ των εφαρμογών και των σχεσιακών βάσεων δεδομένων απαλλάσσοντας τον προγραμματιστή από επιπρόσθετο κόπο και τον κίνδυνο αστοχιών στην αποτύπωση σχέσεων μεταξύ κλάσεων και σχήματος βάσης.

Ο κεντρικός στόχος αυτής είναι η δημιουργία μίας διεπαφής (interface) μεταξύ των διαδεδομένων σχεσιακών βάσεων δεδομένων και του αντικειμενοστραφούς προγραμματισμού. Με άλλα λόγια, επιτρέπει τη χρήση μίας σχεσιακής βάσης δεδομένων ως αντικειμενοστραφή. Για την επίτευξη αυτού, δημιουργούνται αντιστοιχίες μεταξύ των εννοιών του αντικειμενοστραφούς προγραμματισμού, όπως οι συσχετίσεις, η κληρονομικότητα και ο πολυμορφισμός (τα οποία δεν υπάρχουν σε μία σχεσιακή βάση δεδομένων), και των πινάκων και σχέσεων μεταξύ αυτών μίας σχεσιακής βάσης. Με αυτόν τον τρόπο ο προγραμματιστής βλέπει τελικά μία αντικειμενοστραφή βάση δεδομένων, παρ' όλο που στην ουσία χρησιμοποιεί μία σχεσιακή. Έτσι ο προγραμματιστής χρησιμοποιεί τα αντικείμενα της συγκεκριμένης εφαρμογής, τα τροποποιεί σχετικά με τη λογική της εφαρμογής που αναπτύσσει και τα αποθηκεύει (τροποποιεί, διαγράφει και αναζητά) στη βάση ως αντικείμενα. Σκέπτεται, δηλαδή, με αντικειμενοστραφείς έννοιες και όχι με βάση το σχήμα της σχεσιακής βάσης δεδομένων. Έτσι η Hibernate, γνωρίζοντας την αντιστοιχία μεταξύ βάσης και λογικής της εφαρμογής, αναλαμβάνει να κατασκευάσει τον κατάλληλο κώδικα τον οποίο στέλνει τελικά στη βάση δεδομένων. Έπειτα τα αποτελέσματα που επιστρέφονται από τη βάση στην Hibernate δίνονται στην εφαρμογή ως αντικείμενα. Πρόκειται, δηλαδή, ένα ενδιάμεσο επίπεδο μεταξύ εφαρμογής και βάσης δεδομένων.

Πλεονεκτήματα χρήσης της Hibernate

Υπάρχουν πάρα πολλοί λόγοι για τους οποίους κάποιος θα επέλεγε να χρησιμοποιήσει τη Hibernate για την ανάπτυξη μίας εφαρμογής που χρειάζεται κάποιου είδους πρόσβαση σε βάση δεδομένων. Οι πιο σημαντικοί από αυτούς συνοψίζονται παρακάτω :

-Η Hibernate ενδείκνυται για ανάπτυξη πολύπλοκων εφαρμογών. Παρέχει μια σειρά από εργαλεία που επιτρέπουν την εύκολη αποτύπωση αντικειμενοστραφών εννοιών σε σχήματα βάσης ασχέτως του είδους της βάσης που χρησιμοποιείται. Επιτρέπει τη χρήση κληρονομικότητας, πολυμορφισμού και σύνθεσης (composition) στις αντιστοιχιζόμενες κλάσεις και παρέχει μια πολύ ισχυρή γλώσσα ερωτημάτων, την HQL, η οποία επιτρέπει τη χρήση όλων των παραπάνω.

-Μπορεί να βοηθήσει να βελτιωθεί η απόδοση της εφαρμογής. Αυτή η βελτίωση πολύ δύσκολα θα επιτυγχανόταν δημιουργώντας απ' ευθείας κώδικα που να εκτελεί τις επιθυμητές λειτουργίες (hand-coding). Η Hibernate δημιουργεί πολύ αποδοτικά ερωτήματα, πράγμα που διασφαλίζει την απόδοση σε πολλές περιπτώσεις. Εκτός αυτού, όμως, υποστηρίζει μία πολύ έξυπνη και αποδοτική πολιτική πρώτου και δεύτερου επιπέδου caching. Με αυτόν τον τρόπο επιτυγχάνεται μεγάλη δυνατότητα κλιμάκωσης. Επίσης δίνει τη δυνατότητα στον προγραμματιστή να επιλέξει το επίπεδο caching που επιθυμεί, όντας πολύ έξυπνη στην πολιτική των write-backs. Επίσης, μπορεί να συνδυαστεί πολύ καλά με κάποια από τα σημαντικότερα λογισμικά caching τόσο ελεύθερου λογισμικού όσο και εμπορικά πακέτα.

-Επιτρέπει σε μεγάλο βαθμό τη μεταφερισιμότητα των εφαρμογών μεταξύ σχεσιακών βάσεων δεδομένων. Η μόνη διαφοροποίηση στις περισσότερες των περιπτώσεων είναι η αλλαγή μίας μόνο παραμέτρου — της διαλέκτου επικοινωνίας με την βάση. Επιπλέον, οι κλάσεις που αντιστοιχίζονται είναι απλές κλάσεις Java (Plain Old Java Objects — POJOs). Έτσι δε χρειάζεται να είναι απόγονοι μιας πολύπλοκης υποχρεωτικής δομής, με αποτέλεσμα να αυξάνεται ακόμη περισσότερο η μεταφερισιμότητα.

-Παρά όλο που χρειάζεται κάποιο χρόνος για την εκμάθησή της, η Hibernate τελικά αυξάνει σε μεγάλο βαθμό την παραγωγικότητα. Ο προγραμματιστής ενδιαφέρεται μόνο για τα αντικείμενα. Αυτά αποθηκεύονται στη βάση και ανακτώνται από αυτήν με ελάχιστο κόπο.

4.7 JavaScript

Η JavaScript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

Μοντέλο εκτέλεσης

Η αρχική έκδοση της Javascript βασίστηκε στη σύνταξη στη γλώσσα προγραμματισμού C, αν και έχει εξελιχθεί, ενσωματώνοντας πια χαρακτηριστικά από νεότερες γλώσσες.

Αρχικά χρησιμοποιήθηκε για προγραμματισμό από την πλευρά του πελάτη (client), που ήταν ο φυλλομετρητής (browser) του χρήστη, και χαρακτηρίστηκε σαν client-side γλώσσα προγραμματισμού. Αυτό σημαίνει ότι η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού).

Παρά την ευρεία χρήση της Javascript για συγγραφή προγραμμάτων σε περιβάλλον φυλλομετρητή, από την αρχή χρησιμοποιήθηκε και για τη συγγραφή κώδικα από την πλευρά του διακομιστή, από την ίδια τη Netscape στο προϊόν LiveWire, με μικρή επιτυχία. Η χρήση της Javascript στο διακομιστή εμφανίζεται πάλι σήμερα, με τη διάδοση του Node.js, ενός μοντέλου προγραμματισμού βασισμένο στα γεγονότα (events).

4.8 Java

Η ραγδαία εξάπλωση του Internet και του World-Wide Web δημιούργησαν την ανάγκη νέων τρόπων ανάπτυξης και διανομής του λογισμικού. Οι απαιτήσεις

αυτές οδήγησαν στην δημιουργία της γλώσσας προγραμματισμού Java, από την εταιρία Sun microsystems TM.

Η Java σχεδιάστηκε με σκοπό την ανάπτυξη εφαρμογών που θα τρέχουν σε ετερογενή δικτυακά περιβάλλοντα. Η Java έχει τα ακόλουθα χαρακτηριστικά :

- Αντικειμενοστραφής (ομοιότητες εντολών με τη C++).
- Δημιουργία ανεξάρτητων εφαρμογών και applets (applet = προγράμματα που περιλαμβάνονται σε HTML σελίδες και εκτελούνται από τον Web Browser).
- Είναι Interpreted γλώσσα. Αυτό σημαίνει ότι ο java compiler δεν παράγει εκτελέσιμο κώδικα αλλά μια μορφή ψευδοκώδικα (bytecode) το οποίο από μόνο του δεν τρέχει σε καμία μηχανή. Προκειμένου λοιπόν να εκτελεστεί απαιτείται η χρήση ενός interpreter (=διερμηνέα) για να μετατρέψει το bytecode σε πραγματικό εκτελέσιμο κώδικα. Αυτό το χαρακτηριστικό δίνει τη δυνατότητα στα java bytecodes να μπορούν να τρέξουν σε οποιοδήποτε μηχανήμα, κάτω από οποιοδήποτε λειτουργικό, αρκεί να έχει εγκατασταθεί ένας java interpreter. Επίσης ένα άλλο χαρακτηριστικό του java bytecode είναι το μικρό του μέγεθος, (μόλις λίγα Kilobytes). Αυτό το κάνει ιδανικό για μετάδοση μέσω του δικτύου.
- Κατανεμημένη (distributed). Δηλαδή ένα πρόγραμμα σε Java είναι δυνατό να το φέρουμε από το δίκτυο και να το τρέξουμε. Επίσης είναι δυνατό διαφορετικά κομμάτια του προγράμματος να έρθουν από διαφορετικά site.
- Ασφαλής (secure). Στο δίκτυο όμως ελλοχεύουν πολλοί κίνδυνοι για τον χρήστη -παραλήπτη μιας δικτυακής εφαρμογής, γι' αυτό η Java έχει σχεδιαστεί έτσι ώστε να ελαχιστοποιείται η πιθανότητα προσβολής του συστήματος του χρήστη από κάποιο applet γραμμένο για τέτοιο σκοπό.
- Είναι multithreaded. Η Java υποστηρίζει εγγενώς την χρήση πολλών threads. Προκειμένου να το επιτύχει αυτό σε συστήματα με έναν επεξεργαστή, το Java runtime system (interpreter) υλοποιεί ένα δικό χρονοδρομολογητή (scheduler), ενώ σε συστήματα που υποστηρίζουν πολυεπεξεργασία η δημιουργία των threads ανατίθεται στο λειτουργικό σύστημα. Φυσικά όλα αυτά είναι αόρατα τόσο στον προγραμματιστή όσο και στον χρήστη.

- Υποστηρίζει multimedia εφαρμογές. Με αυτό εννοούμε ότι η Java παρέχει ευκολίες στη δημιουργία multimedia εφαρμογών. Αυτό επιτυγχάνεται τόσο με την ευελιξία της σαν γλώσσα όσο και με τις πλούσιες και συνεχώς εμπλουτιζόμενες βιβλιοθήκες της.

4.9 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι (αγγλ. My). Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιοκτητών συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.

Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το Flickr, το YouTube, η Wikipedia, το Google, το Facebook και το Twitter.

4.10 Λοιπές τεχνολογίες

Eclipse

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το εργαλείο Eclipse Platform και συγκεκριμένα η έκδοση 4.3.2 (Eclipse Kepler) . Περισσότερα μπορείτε να μάθετε στον ιστότοπο <http://www.eclipse.org/platform>

Apache Tomcat

Ο Apache Tomcat είναι εφαρμογή ανοιχτού κώδικα και εφαρμόζει τις

τεχνολογίες JavaServlet και JavaServer Pages και προσφέρει έναν HTTP web server περιβάλλον για οποιαδήποτε εφαρμογή γραμμένη σε java να «τρέξει». Η εφαρμογή «έτρεχε» πάνω στο server Apache Tomcat Server v7

5

Βιβλιογραφία

Κεφάλαιο 5 - Βιβλιογραφία

Βιβλιογραφία:

1. Kenneth C.Laudon, Jane P.Laudon.'Management Information Systems 7 th edition'
2. Turban,McLean,Wetherbe. 'Information Technology:Transforming business into digital
3. Robert V. Williams. 'History of Information Science and Technology.'
4. C. Bourne&T. B. Hahn. 'A History of Information Serviceseconomy 3 rd edition'
5. Tim Jennings.' Defining The Document And Content Management
6. Ecosystem'
7. Σπύρος Μουζακίτης "Σχεδιασμός και Υλοποίηση Συστήματος Διαχείρισης Εγγράφων"
8. Παπαγεωργίου Παρασκευή «Ανάπτυξη διαδικτυακής εφαρμογής κοινωνικής δικτύωσης για ψυχαγωγικά γεγονότα»

Σύνδεσμοι:

1. Spring Documentation (<http://www.springsource.org/documentation>)
2. <http://old.ceid.upatras.gr/courses/katanemhmena/ds2/index.php/2009-10.09-17-08-27-11/61-hibernate>
3. http://www.ebooks4greeks.gr/downloads/Pliroforiki/Glosses.program./Java__Downloaded_from_eBooks4Greeks.gr.pdf
4. <http://webapptester.com/mvc-framework-first-impression/>
5. <http://monster.salary.com/salarywizard/IT-Project-Coordinator-Job-Description.aspx>
6. <http://el.wikipedia.org/wiki/MySQL>
7. http://en.wikipedia.org/wiki/Spring_Framework
8. [http://en.wikipedia.org/wiki/Hibernate_\(Java\)](http://en.wikipedia.org/wiki/Hibernate_(Java))