



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΔΗΜΙΟΥΡΓΙΑ – ΔΙΟΡΘΩΣΗ ΚΑΙ ΠΑΡΑΓΩΓΗ
ΕΚΤΥΠΩΣΙΜΟΥ ΟΔΗΓΟΥ ΣΠΟΥΔΩΝ ΜΕ ΣΥΝΔΥΑΣΜΟ
ΣΤΑΤΙΚΗΣ ΚΑΙ ΔΥΝΑΜΙΚΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕΣΩ
ΔΙΑΔΙΚΤΥΟΥ**

Διπλωματική Εργασία
ΤΟΥ
Ιωάννη Η. Κωνσταντίνου

Επιβλέπων : Νικόλαος Μήτρου
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2004



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΔΗΜΙΟΥΡΓΙΑ – ΔΙΟΡΘΩΣΗ ΚΑΙ ΠΑΡΑΓΩΓΗ
ΕΚΤΥΠΩΣΙΜΟΥ ΟΔΗΓΟΥ ΣΠΟΥΔΩΝ ΜΕ ΣΥΝΔΥΑΣΜΟ
ΣΤΑΤΙΚΗΣ ΚΑΙ ΔΥΝΑΜΙΚΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕΣΩ
ΔΙΑΔΙΚΤΥΟΥ**

Διπλωματική Εργασία
ΤΟΥ
Ιωάννη Η. Κωνσταντίνου

Επιβλέπων : Νικόλαος Μήτρου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 27η Οκτωβρίου 2004

.....
Μήτρου Νικόλαος
Καθηγητής Ε.Μ.Π.

.....
Θεολόγου Μιχαήλ
Καθηγητής Ε.Μ.Π.

.....
Συκάς Ευστάθιος
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2004

.....

ΙΩΑΝΝΗΣ Η. ΚΩΝΣΤΑΝΤΙΝΟΥ

© 2004 – All rights reserved

Περίληψη

Το αντικείμενο της παρούσης διπλωματικής εργασίας είναι η δημιουργία μιας δικτυακής εφαρμογής για την επεξεργασία, μορφοποίηση και αποθήκευση σε εκτυπώσιμη μορφή (PDF) του Οδηγού Σπουδών της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ με χρήση τεχνολογιών XML-XSL. Η εφαρμογή είναι εγκαταστημένη στον δικτυακό τόπο της σχολής και παρέχει την δυνατότητα διαχείρισης και ανανέωσης τόσο των δυναμικών στοιχείων του οδηγού που υπάρχουν στην βάση δεδομένων της σχολής, όσο και των στατικών δεδομένων που υπάρχουν σε αρχεία κειμένου. Για την υλοποίηση της εφαρμογής έγινε χρήση πολλών τεχνολογιών διαδικτύου, οι οποίες και αναλύονται στη συνέχεια.

Abstract

The scope of this diploma thesis is the creation of an internet web application to manage, format and save in a printable format (PDF) the Students Guide of the School of Electrical and Computer Engineering using XML – XSL technologies. The application is installed in the school's web site and it can be used to easily manage and update both the dynamic data which is extracted from the school' s database and the static data which is contained in text files. For the creation of the application a variety of internet technologies are used, which are going to be presented.

Πίνακας Περιεχομένων

Περίληψη.....	5
Abstract.....	6
Πίνακας Περιεχομένων.....	7
Συνομογραφίες.....	9
1. Περιγραφή Διπλωματικής Εργασίας.....	10
2. Αρχιτεκτονική και λειτουργική περιγραφή της εφαρμογής.....	12
2.1 Γενική Περιγραφή προβλήματος.....	12
2.2 Λύση προβλήματος.....	15
2.3 Διαχείριση Δυναμικού Περιεχομένου.....	18
2.4 Διαχείριση Στατικού Περιεχομένου.....	20
2.5 Σύνθεση Στατικού - Δυναμικού.....	23
2.6 Περιγραφή διαθέσιμων επιλογών εφαρμογής σε UML.....	25
3. Τεχνολογίες υλοποίησης.....	27
3.1 Γλώσσα Προγραμματισμού Java.....	28
3.2 Java Servlets.....	29
3.3 Adobe PDF.....	30
3.4 CVS (Concurrent Version System).....	32
3.5 Struts.....	33
3.6 XML-XSL.....	34
4. Υλοποίηση - Τεχνική Περιγραφή.....	35
4.1 Περιγραφή χρησιμοποιούμενων βιβλιοθηκών.....	35
4.2 Περιγραφή “Διαχείρισης Δυναμικού Περιεχομένου”.....	36
4.3 Περιγραφή “Διαχείρισης Δυναμικού Περιεχομένου που δεν περιέχεται στην βάση”.....	37
4.4 Περιγραφή “Σύνθεσης Τελικού Οδηγού Σπουδών”.....	37
4.5 Περιγραφή “Διαχείρισης Στατικού Περιεχομένου”.....	39
5. Οδηγός χρήσης της εφαρμογής.....	41
5.1 Κυρίως Μενού.....	42
5.2 Διαχείριση Δυναμικού Περιεχομένου.....	43
5.2.1 Ενημέρωση δεδομένων κεφαλαίων 5,7,8,9,10.....	43
5.2.2 Ενημέρωση δεδομένων που δεν περιέχονται στην βάση δεδομένων.....	44
5.3 Διαχείριση Στατικού Περιεχομένου.....	46
5.3.1 Προσθήκη.....	47
5.3.2 Αντικατάσταση.....	48
5.3.3 Επεξεργασία.....	48
5.3.4 Μετονομασία.....	49
5.3.5 Διαγραφή.....	50
5.3.6 Επιστροφή.....	51
5.3.7 Βοήθεια.....	51
5.4 Σύνθεση Τελικού Οδηγού Σπουδών.....	52
5.4.1 Δημιουργία Οδηγού.....	53
5.4.2 Επεξεργασία Οδηγού.....	54
5.4.3 Διαγραφή Οδηγού.....	55
5.4.4 Προσθήκη νέου Κεφαλαίου.....	55
5.4.5 Διόρθωση Κεφαλαίου.....	57
5.4.6 Διαγραφή Κεφαλαίου.....	58
5.4.7 Δημιουργία οδηγού σε PDF.....	59
5.4.8 Επιστροφή.....	60
5.4.9 Βοήθεια.....	60
5.5 Προβολή διαθέσιμων Οδηγών Σπουδών.....	61
5.5.1 Προβολή.....	61

5.5.2	Μετονομασία.....	62
5.5.3	Διαγραφή.....	63
5.5.4	Επιστροφή.....	63
5.5.5	Βοήθεια.....	63
6.	Σενάρια πρακτικών εφαρμογών.....	64
6.1	Ενημέρωση οδηγού για αλλαγές σε στατικό περιεχόμενο.....	64
6.2	Ενημέρωση οδηγού για αλλαγές σε δυναμικό περιεχόμενο.....	69
6.3	Ενημέρωση οδηγού σε δυναμικό περιεχόμενο που δεν περιέχεται στην βάση δεδομένων.....	71
6.4	Προσθήκη νέου κεφαλαίου στον οδηγό.....	73
7.	Συμπεράσματα.....	78
	ΠΑΡΑΡΤΗΜΑ - ΚΩΔΙΚΑΣ.....	81
Π.1	gr.ntua.ece.util.NavigateXmlUtils.java.....	81
Π.2	gr.ntua.ece.util.XmlUtils.java.....	86
Π.3	gr.ntua.ece.util.fileUtils.java.....	95
Π.4	addNewChapter.jsp.....	97
Π.5	bottomFrame.jsp.....	99
Π.6	chapterAdded.jsp.....	102
Π.7	chapterEdited.jsp.....	103
Π.8	deleteChapter.jsp.....	104
Π.9	deletecourse.jsp.....	105
Π.10	deleteFile.jsp.....	106
Π.11	deleteFileOdhgos.jsp.....	106
Π.12	deleteNodetree.jsp.....	107
Π.13	deletesecretary.jsp.....	108
Π.14	docList.jsp.....	109
Π.15	editChapter.jsp.....	110
Π.16	editNonDbData.jsp.....	114
Π.17	errorhandler.jsp.....	122
Π.18	extractDbData.jsp.....	122
Π.19	extractPdf.jsp.....	124
Π.20	extractPdfAction.jsp.....	129
Π.21	loadNodetree.jsp.....	131
Π.22	mainMenu.jsp.....	132
Π.23	newFile.jsp.....	133
Π.24	newFileAction.jsp.....	134
Π.25	newNodeTree.jsp.....	135
Π.26	pdfDocMenu.jsp.....	136
Π.27	pdfList.jsp.....	138
Π.28	renameFile.jsp.....	139
Π.29	renameFileOdhgos.jsp.....	141
Π.30	replaceFile.jsp.....	142
Π.31	replaceFileAction.jsp.....	143
Π.32	showGuidesBottom.jsp.....	145
Π.33	showGuidesMain.jsp.....	146
Π.34	tree.jsp.....	147

Συντομογραφίες

XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language
PDF	Portable Document Format
JSP	Java Server Pages
HTML	Hyper Text Markup Language
API	Application Protocol Interface
FOP	Formatting Objects Processor
CVS	Concurrent Version System

1. Περιγραφή Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία αποτελεί ένα εργαλείο για τη διαχείριση του οδηγού σπουδών της σχολής. Με τη χρήση του εν λόγω εργαλείου μπορεί ο κάθε υπεύθυνος να δημιουργήσει τον οδηγό σπουδών της σχολής σε εκτυπώσιμη μορφή από την αρχή ή απλά να τον ενημερώσει. Στη βάση δεδομένων της σχολής είναι αποθηκευμένες πληροφορίες που αφορούν το διδακτικό προσωπικό της σχολής, τα διδασκόμενα μαθήματα, τα εργαστήρια, κ.ά. Από εκεί αντλούνται όλες οι πληροφορίες που ενημερώνουν δυναμικά τα αντίστοιχα κεφάλαια του οδηγού σπουδών.

Η ιδέα έχει ως εξής : Η πληροφορία που χρειάζεται για την σύνθεση του οδηγού σπουδών είναι 2 ειδών : στατική και δυναμική. Η δυναμική προέρχεται από την βάση δεδομένων της σχολής, και συλλέγεται αυτόματα από την εφαρμογή με την επιλογή «Διαχείριση δυναμικού Περιεχομένου». Με την πληροφορία αυτή δημιουργούνται τα κεφάλαια 5,7,8,9,10 του οδηγού της σχολής (κατά σειρά Προσωπικό Σχολής, Πρόγραμμα Κορμού, Πρόγραμμα Ρωών, Περιγραφή Μαθημάτων Κορμού και Περιγραφή Μαθημάτων Ρωών) σε μορφή PDF.

Τα στατικά δεδομένα υπάρχουν σε έναν κατάλογο στον Server σε μορφή .doc και .pdf, και είναι προσβάσιμα στον χρήστη μέσω της επιλογής “Διαχείριση Στατικού Περιεχομένου” στο αρχικό μενού της εφαρμογής. Για επεξεργαστεί κανείς αυτά τα δεδομένα πρέπει να τα “κατεβάσει” τοπικά στον υπολογιστή του σε μορφή .doc, να κάνει τις αλλαγές που αυτός θέλει, να ξαναδημιουργήσει τα αντίστοιχα .pdf αρχεία με την εφαρμογή Adobe Acrobat (όχι τον Reader) και να ανεβάσει και τα 2 αλλαγμένα αρχεία (τόσο το .doc όσο και το .pdf) πάλι στον server χρησιμοποιώντας την εφαρμογή αυτή. Κατ’αυτόν τον τρόπο η αρχική πληροφορία των αρχείων .doc θα υπάρχει σε μόνο μία θέση στον server, και δεν θα υπάρχει πρόβλημα με ταυτόχρονες εκδόσεις.

Εφόσον ο χρήστης δημιουργήσει τα απαραίτητα κεφάλαια σε μορφή .pdf (τόσο τα δυναμικά όσο και τα στατικά) θα πρέπει αυτά να τα συνενώσει και να δημιουργήσει τον τελικό οδηγό σπουδών. Αυτό γίνεται από την επιλογή “Σύνθεση Τελικού Οδηγού Σπουδών” στο Αρχικό μενού της εφαρμογής. Σε γενικές γραμμές, σε αυτό το μέρος συνενώνονται τα επιμέρους κεφάλαια του οδηγού που έχουν δημιουργηθεί σε .pdf και δημιουργείται ο τελικός οδηγός Σπουδών. Παρέχονται βέβαια πολλές δυνατότητες στον χρήστη, όπως η αυτόματη δημιουργία του πίνακα

των περιεχομένων του οδηγού, προεπισκόπηση σε pdf του τελικού οδηγού κατά το στάδιο της δημιουργίας του και άλλα που θα αναλυθούν σε παρακάτω ενότητα.

Η εφαρμογή δίνει την δυνατότητα διαχείρισης διαφορετικών εκδόσεων των οδηγών σπουδών (πχ ανά ακαδημαϊκό έτος), μετονομασία αυτών, διαγραφή και προεπισκόπηση αυτών, με την επιλογή “Προβολή διαθέσιμων οδηγών Σπουδών” στο αρχικό μενού της εφαρμογής.

Η εργασία αυτή είναι συνέχεια και επέκταση της διπλωματικής εργασίας του Βλοτομά Νεόφυτου με θέμα Επεξεργασία και Μορφοποίηση σε εκτυπώσιμη μορφή πληροφορίας από βάση δεδομένων με χρήση τεχνολογίας XML-XSL. Η εν λόγω διπλωματική είχε δημιουργηθεί με το περιβάλλον Java Swing, και χρειαζόταν την γλώσσα προγραμματισμού Java 1.3.1 σε κάθε υπολογιστή που γινόταν εγκατάσταση.

2. Αρχιτεκτονική και λειτουργική περιγραφή της εφαρμογής.

Στο κεφάλαιο που ακολουθεί θα γίνει μια αναλυτική περιγραφή της αρχιτεκτονικής που χρησιμοποιήθηκε για την υλοποίηση του συγκεκριμένου θέματος. Θα περιγράψουμε ποιες είναι οι απαιτήσεις της εφαρμογής σε γενικό επίπεδο, ποια προβλήματα προκύπτουν από τις απαιτήσεις αυτές, και πως αποφασίστηκε να αντιμετωπιστούν τα προβλήματα αυτά. Κατόπιν, θα γίνει μια περιγραφή των διαθέσιμων επιλογών του χρήστη, και για αυτό το σκοπό παραθέτουμε και ένα διάγραμμα UML.

2.1 Γενική Περιγραφή προβλήματος

Το πρόβλημα που καλούμαστε να αντιμετωπίσουμε με την συγκεκριμένη εφαρμογή είναι η διαχείριση του οδηγού σπουδών της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ με τρόπο εύκολο και κατανοητό από έναν απλό χρήστη, αποφεύγοντας παρεμβάσεις του που απαιτούν ιδιαίτερες γνώσεις υπολογιστών.

Ο οδηγός σπουδών είναι ένα κείμενο που είναι ανάγκη αλλάζει αρκετά συχνά, καθώς περιέχει μεγάλο όγκο συχνά μεταβαλλόμενης πληροφορίας. Ο οδηγός περιγράφει την δομή της σχολής, τους καθηγητές που την απαρτίζουν, τα μαθήματα που γίνονται με μια σύντομη περιγραφή για το καθένα, πράγματα που βρίσκονται σε συνεχή εξέλιξη για την βελτίωση του παρεχόμενου επιπέδου σπουδών.

Ένας ακόμα λόγος για τον οποίο πρέπει να ανανεώνεται συχνά ο οδηγός σπουδών, είναι πως ο οδηγός είναι δημοσιευμένος στην ηλεκτρονική διεύθυνση της Σχολής, την οποία αρκετός κόσμος την επισκέπτεται καθημερινά. Ο ανανεωμένος οδηγός σπουδών μπορεί να δώσει up to date πληροφορίες στους χρήστες αυτούς, είτε αυτοί είναι άτομα που θέλουν να φοιτήσουν στην συγκεκριμένη σχολή και ψάχνουν πληροφορίες για αυτή, είτε αυτοί είναι φοιτητές σε αυτή και ψάχνουν συγκεκριμένες πληροφορίες για μαθήματα που ενδιαφέρονται να παρακολουθήσουν.

Το πρόβλημα όμως που προκύπτει είναι πως ο οδηγός είναι σε μορφή pdf. Το πρότυπο pdf είναι ένα πρότυπο κλειστό προς επεξεργασία, έτσι ο οδηγός σε μορφή pdf δεν μπορεί να υποστεί καμία επεξεργασία στην μορφή αυτή.

Επομένως, για να ανανεώσει κανείς τον οδηγό, θα πρέπει να ανανεώσει τα δεδομένα από τα οποία δημιουργήθηκε αυτός, και κατόπιν να τον επανασυνθέσει από την αρχή.

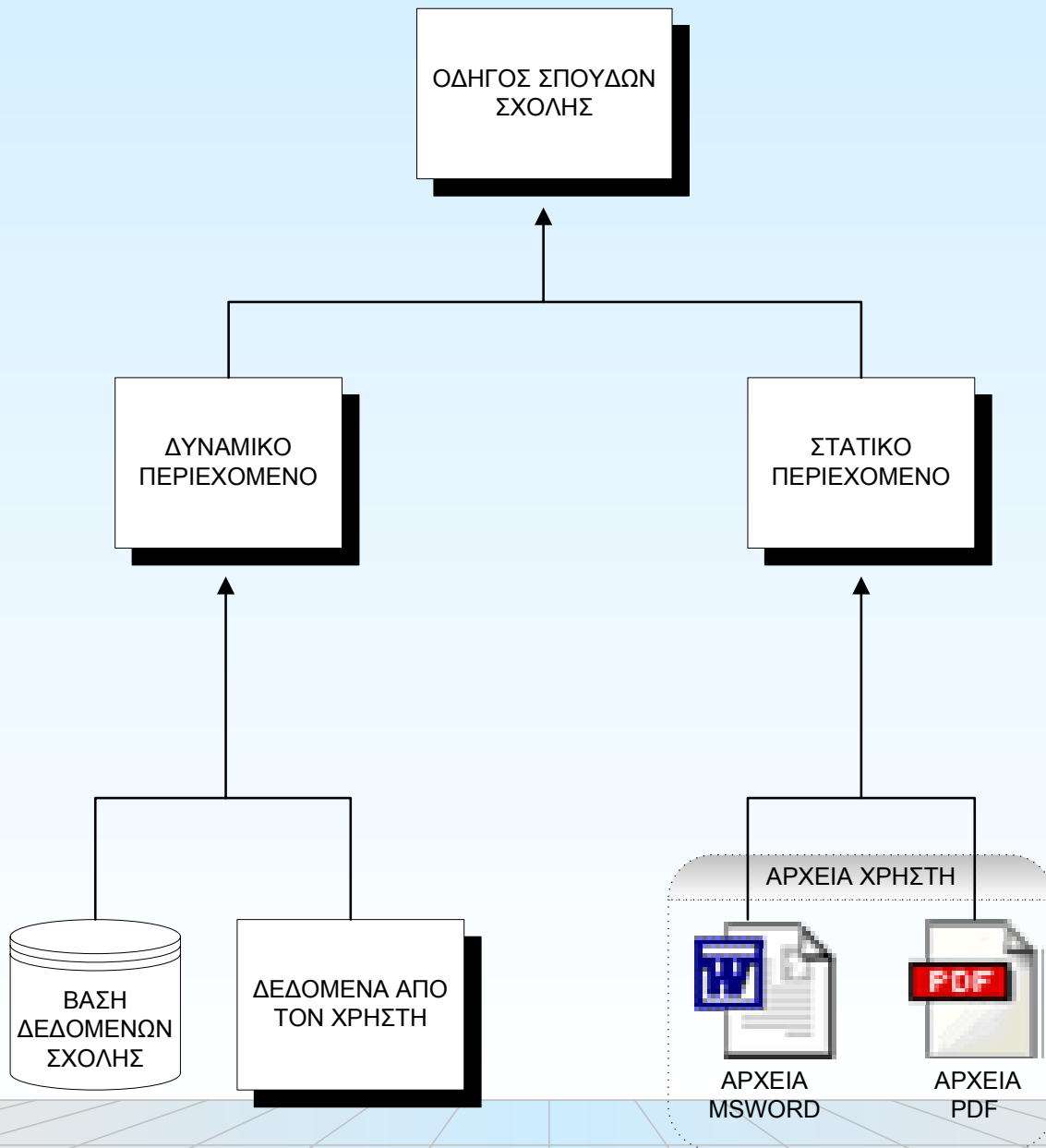
Για τις ανάγκες της εφαρμογής, θεωρούμε ότι ο οδηγός αποτελείται από δυναμικό και στατικό περιεχόμενο.

Με τον όρο δυναμικό περιεχόμενο εννοούμε τα στοιχεία που υπάρχουν αποθηκευμένα στην βάση δεδομένων της σχολής (υπάρχον προσωπικό – διδασκόμενα μαθήματα – περιγραφή αυτών, κα) καθώς και άλλα στοιχεία που δεν υπάρχουν σε αυτό το μέρος, και πρέπει να γίνει ιδιαίτερη αντιμετώπιση για αυτά από την εφαρμογή. Το δυναμικό περιεχόμενο αποτελεί κάποια συγκεκριμένα κεφάλαια του οδηγού σπουδών.

Με τον όρο στατικό περιεχόμενο εννοούμε κάποια κείμενα του οδηγού σπουδών που δεν είναι αποθηκευμένα με τον προηγούμενο τρόπο, απλά υπάρχουν μόνο σε ηλεκτρονική μορφή αρχείων κειμένου MS Word. Τέτοια κείμενα (όπως ο πρόλογος του οδηγού) μέχρι τώρα για να αλλάξουν έπρεπε να υπάρχουν τοπικά στον υπολογιστή του χρήστη που θέλει να κάνει την αλλαγή, και εάν αυτό δεν ήταν δυνατό, μερικές φορές χρειαζόταν ο χρήστης να τα ξαναγράψει από την αρχή.

Παρακάτω παραθέτουμε ένα διάγραμμα που περιγράφει σε γενικές γραμμές το συγκεκριμένο πρόβλημα.

ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ



2.2 Λύση προβλήματος

Μελετώντας την συγκεκριμένη μορφή του προβλήματος καθώς και την ανομοιογένεια της πληροφορίας από την οποία παράγεται ο οδηγός σπουδών, καλούμαστε να πάρουμε αποφάσεις για την αρχιτεκτονική του συστήματος, έτσι ώστε να προβλέπει εύκολη διαχείριση τέτοιου είδους πληροφορίας.

Για την αντιμετώπιση αυτού του προβλήματος, καλό θα ήταν να το χωρίσουμε σε μικρότερα υποπροβλήματα, που όλα μαζί απαρτίζουν το αρχικό.

Κατ' αυτόν τον τρόπο, και λαμβάνοντας υπ' όψη την διαφοροποίηση της μορφής του περιεχομένου, μπορούμε να χωρίσουμε το πρόβλημα της ανάπτυξης σε 3 μικρότερα υποπροβλήματα :

- Διαχείριση δυναμικού περιεχομένου
- Διαχείριση στατικού περιεχομένου
- Σύνθεση Στατικού – δυναμικού περιεχομένου

Με τον διαχωρισμό αυτό μπορούμε να αντιμετωπίσουμε την διαφορετικότητα της πληροφορίας, χωρίζοντας την εφαρμογή σε 3 μέρη, σύμφωνα με τα διαφορετικά προβλήματα που καλούμαστε να αντιμετωπίσουμε.

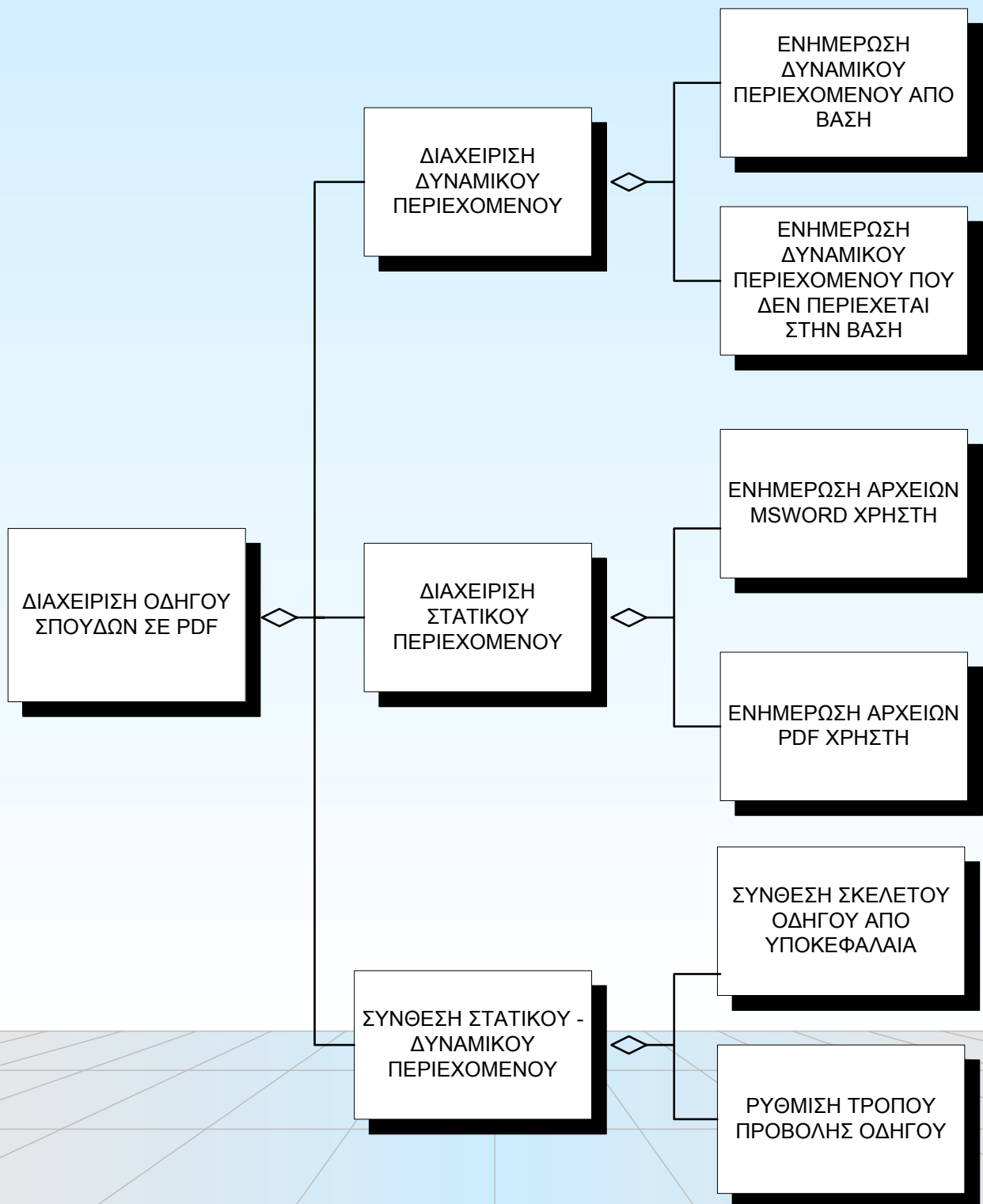
Έτσι, σε γενικές γραμμές, στο μέρος της διαχείρισης δυναμικής πληροφορίας ο χρήστης μπορεί να ανανεώσει τα κεφάλαια του οδηγού που προκύπτουν από δυναμική πληροφορία (με τον τρόπο που αυτή ορίστηκε στην προηγούμενη ενότητα).

Στο μέρος της διαχείρισης στατικού περιεχομένου ο χρήστης έχει την δυνατότητα να διαχειριστεί τα αρχεία κειμένου MS Word από τα οποία είναι κατασκευασμένο το στατικό μέρος του οδηγού σπουδών, και να κατασκευάσει τα αντίστοιχα pdf αρχεία.

Στο τρίτο μέρος, “Σύνθεση στατικού - δυναμικού περιεχομένου” ο χρήστης αφού τελειώσει με την ενημέρωση του στατικού – δυναμικού μέρους το οδηγού, μπορεί να τα ενώσει έτσι ώστε να δημιουργήσει τον τελικό οδηγό σπουδών με την ανανεωμένη πληροφορία με τρόπο εύκολο. Εδώ ο χρήστης μπορεί να δημιουργήσει έναν “σκελετό” για τον οδηγό, περιγράφοντας στον σκελετό αυτό από ποια υποκεφάλαια θα αποτελείται ο οδηγός και τι είδους κεφάλαια είναι αυτά. Με αυτόν τον τρόπο, κάθε φορά που ο χρήστης θα θέλει να αλλάζει μια μικρή λεπτομέρεια σε ένα κεφάλαιο, δεν θα χρειάζεται να αλλάζει όλο τον σκελετό του εγγράφου, απλά να τον ενημερώνει για τις αλλαγές.

Παρακάτω παραθέτουμε ένα διάγραμμα που περιγράφει από ποια υποπροβλήματα αποτελείται το βασικό πρόβλημα, και πως αυτά συνδέονται μεταξύ τους.

ΛΥΣΗ ΠΡΑΚΤΙΚΟΥ ΠΡΟΒΛΗΜΑΤΟΣ



2.3 Διαχείριση Δυναμικού Περιεχομένου

Σε αυτή την ενότητα θα περιγράψουμε αναλυτικότερα την αρχιτεκτονική που ακολουθήσαμε για την διαχείριση Δυναμικού Περιεχομένου, όπως αυτό ορίστηκε σε προηγούμενη υποενότητα.

Η εσωτερική δομή ενός αρχείου pdf είναι παρόμοια με αυτή ενός xml. Είναι και οι 2 “tag-based”, αποτελούνται δηλαδή από ετικέτες της μορφής <tag> «κειμενο» </tag>, οι οποίες προσδίδουν μια ιδιότητα στο κείμενο που εσωκλείουν. Επίσης, έχουν δημιουργηθεί αρκετές βιβλιοθήκες τόσο για την επεξεργασία αρχείων xml, όσο και για τον μετασχηματισμό τους σε άλλης μορφής αρχεία, στην προκειμένη περίπτωση δηλαδή αρχεία pdf.

Έτσι, αποφασίστηκε η χρήση αρχείων xml σαν ενδιάμεσο μέσο αποθήκευσης της πληροφορίας, όπως αυτή εξάγεται από την βάση δεδομένων της σχολής και από τον χρήστη με κάποιες φόρμες που κατασκευάστηκαν. Τα αρχεία αυτά βρίσκονται στον server της σχολής, και είναι μη ορατά στον τελικό χρήστη. Τα xml αρχεία περιέχουν μόνο δεδομένα, δεν υπάρχει δηλαδή πουθενά μέσα στο αρχείο πληροφορία για τον τρόπο εμφάνισης του αρχείου αυτού.

Ο τρόπος εμφάνισης των xml αρχείων βρίσκεται στα αρχεία xsl, που και αυτά είναι αποθηκευμένα στον server. Στα αρχεία αυτά είναι αποθηκευμένοι σε γενικές γραμμές κανόνες με τους οποίους επιλέγονται συγκεκριμένοι κόμβοι του xml αρχείου, και τρόποι μορφοποίησης που εφαρμόζονται πάνω στους κόμβους αυτούς. Κάθε xml αρχείο που υπάρχει στον server έχει και το αντίστοιχο αρχείο xsl που είναι υπεύθυνο για τον τρόπο εμφάνισής του. Όταν αλλάζουν τα δεδομένα, μόνο τα αρχεία xml αλλάζουν τα xsl παραμένουν αναλλοίωτα.

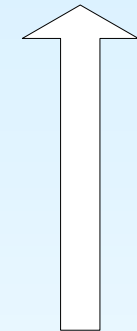
Κατ’ αυτόν τον τρόπο, έχουμε διαχωρίσει την αυτό καθ’ αυτό πληροφορία από τον τρόπο εμφάνισής της.

Χρησιμοποιώντας τώρα τους μετασχηματισμούς xsl σε συνδυασμό με τους μετασχηματισμούς for παίρνουμε το τελικό αποτέλεσμα : το αρχείο pdf. Οι μετασχηματισμοί for γίνονται από μία βιβλιοθήκη που περιέχει τον απαραίτητο κώδικα για την εγγραφή σε αρχεία pdf των αποτελεσμάτων των αρχείων xsl επάνω στα xml. (Γράφονται οι απαραίτητοι headers και άλλα στοιχεία από τα οποία αποτελείται ένα αρχείο pdf). Η βιβλιοθήκη for έχει υλοποιηθεί σε αρκετές γλώσσες προγραμματισμού. Παρακάτω, παραθέτουμε σε διάγραμμα την αρχιτεκτονική που επιλέξαμε για την αντιμετώπιση του προβλήματος της διαχείρισης του δυναμικού περιεχομένου του οδηγού σπουδών.

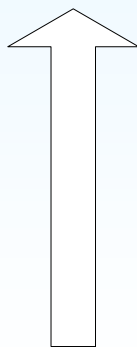
ΔΙΑΧΕΙΡΙΣΗ ΔΥΝΑΜΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ



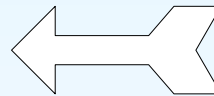
ΔΥΝΑΜΙΚΟ
ΑΡΧΕΙΟ PDF



Αρχεία
XML



XSL
TRANSFORMATIONS



FOP
TRANSFORMATIONS

ΔΕΔΟΜΕΝΑ ΑΠΟ
ΤΟΝ ΧΡΗΣΤΗ

ΒΑΣΗ
ΔΕΔΟΜΕΝΩΝ
ΣΧΟΛΗΣ

2.4 Διαχείριση Στατικού Περιεχομένου

Σε αυτό το μέρος περιγράφεται αναλυτικά η αρχιτεκτονική του συστήματος που θα χρησιμοποιηθεί για την διαχείριση του στατικού περιεχομένου του οδηγού σπουδών.

Όπως προαναφέρθηκε, το στατικό περιεχόμενο αποτελείται από αρχεία κειμένου doc (MS Word). Η επεξεργασία των αρχείων αυτών από τον χρήστη απαιτεί την ύπαρξη της εφαρμογής Microsoft Word στον υπολογιστή του. Η εφαρμογή αυτή εκτελείται μόνο σε περιβάλλον Microsoft Windows. Το γεγονός όμως ότι ο server της σχολής χρησιμοποιεί Linux λειτουργικό σύστημα, καθιστά την αδύνατη την εγκατάσταση της εφαρμογής, και κατά συνέπεια την επεξεργασία των doc αρχείων από τον server.

Έτσι, αποφασίστηκε τα αρχεία doc να υπάρχουν στον server, αλλά για να γίνει επεξεργασία σε αυτά να χρειάζεται ο χρήστης να τα κατεβάζει (download) τοπικά στον υπολογιστή του.

Μόλις ο χρήστης κάνει τις απαραίτητες αλλαγές στο αρχείο που έχει “κατεβάσει”, πρέπει να το αντικαταστήσει στον server με το καινούριο, έτσι ώστε εάν κάποιος άλλος θέλει να κάνει και αυτός αλλαγές να έχει πρόσβαση στο καινούριο, και όχι σε μια παλιά έκδοση του.

Βέβαια, από την στιγμή που ο server δεν έχει την δυνατότητα να επεξεργαστεί τα αρχεία doc, στην ουσία δεν του είναι χρήσιμα. Ο λόγος που υπάρχουν εκεί είναι για να είναι εύκολα προσβάσιμα στους συγγραφείς του οδηγού. Τα αρχεία που ο server χρειάζεται για την δημιουργία του τελικού οδηγού είναι τα pdf που εξάγονται από τα doc.

Την λειτουργία αυτή κάνει το πρόγραμμα “Acrobat Distiller” της εταιρίας Adobe Acrobat. Το πρόγραμμα αυτό κατ’ αυτόν τον τρόπο πρέπει να είναι εγκατεστημένο στον υπολογιστή του χρήστη, και μόλις τελειώνει τις αλλαγές στα doc, να τα μετατρέπει σε pdf με αυτό, και να ανεβάζει και τα 2 αλλαγμένα αρχεία στον server (τόσο το doc όσο και το pdf).

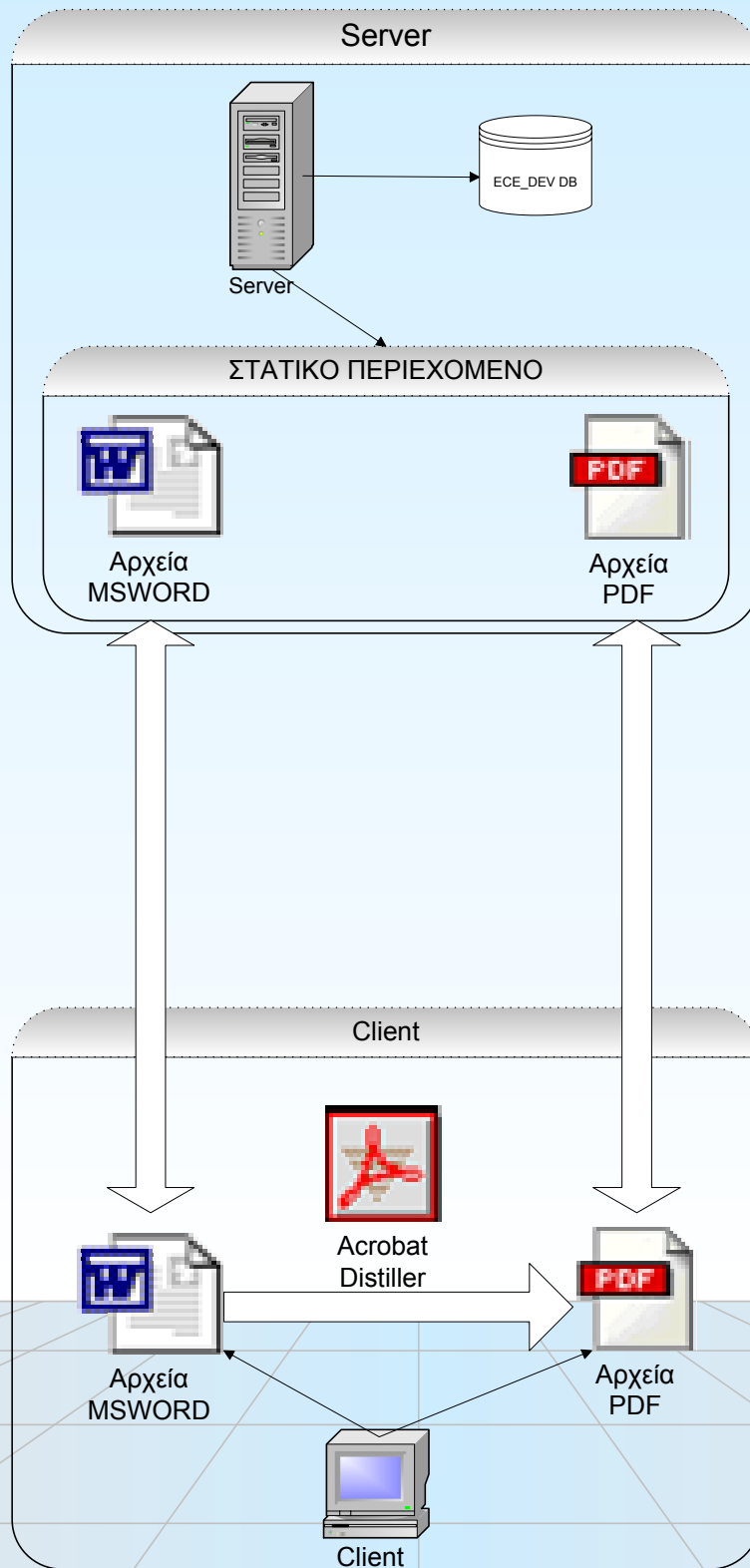
Ο λόγος που επιλέχθηκε η μετατροπή από doc σε pdf να γίνεται στον υπολογιστή του client και όχι του server είναι οι περιορισμοί που μας θέτει το λειτουργικό σύστημα Linux, καθώς για να γίνει αυτή η μετατροπή απαιτούνται τόσο το πρόγραμμα Acrobat Distiller όσο και το πρόγραμμα MS Word.

Ο Acrobat Distiller υπάρχει για :Linux, αλλά ο Ms Word όχι.

Σε αυτό το σημείο θα πρέπει να τονίζεται στον χρήστη κατά την ενημέρωση του στατικού περιεχομένου να ενημερώνει τόσο τα doc όσο και τα pdf αρχεία, και να φροντίζει τα pdf να συμβαδίζουν με τα αντίστοιχα doc από τα οποία δημιουργήθηκαν, γιατί αλλιώς δεν θα λειτουργεί σωστά το σύστημα που από μόνο του δεν έχει την δυνατότητα να κάνει αυτόν τον έλεγχο.

Παρακάτω παραθέτουμε ένα σχήμα που περιγράφει παραστατικά την αρχιτεκτονική που αναλύθηκε σε αυτό το κεφάλαιο.

ΔΙΑΧΕΙΡΙΣΗ ΣΤΑΤΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ



2.5 Σύνθεση Στατικού - Δυναμικού

Σε αυτό το μέρος περιγράφουμε την αρχιτεκτονική που ακολουθήθηκε για την σχεδίαση του μέρους Σύνθεσης Στατικού – Δυναμικού Περιεχομένου, από όπου θα δημιουργεί ο χρήστης τον τελικό οδηγό σπουδών.

Σε αυτό το μέρος, έπρεπε να χρησιμοποιηθεί μια ενδιάμεση αποθηκευτική δομή που θα περιέχει πληροφορίες για τον σκελετό του προς δημιουργία οδηγού σπουδών.

Σε αυτή την δομή πρέπει να περιέχονται πληροφορίες για τον τρόπο με τον οποίο θα ενωθούν τα διάφορα κομμάτια pdf του οδηγού που δημιουργήθηκαν είτε από στατικό είτε από δυναμικό περιεχόμενο, έτσι ώστε ο server να μπορεί να δημιουργήσει τον τελικό οδηγό βασισμένος στην πληροφορία της δομής αυτής και σε αυτή του περιεχομένου.

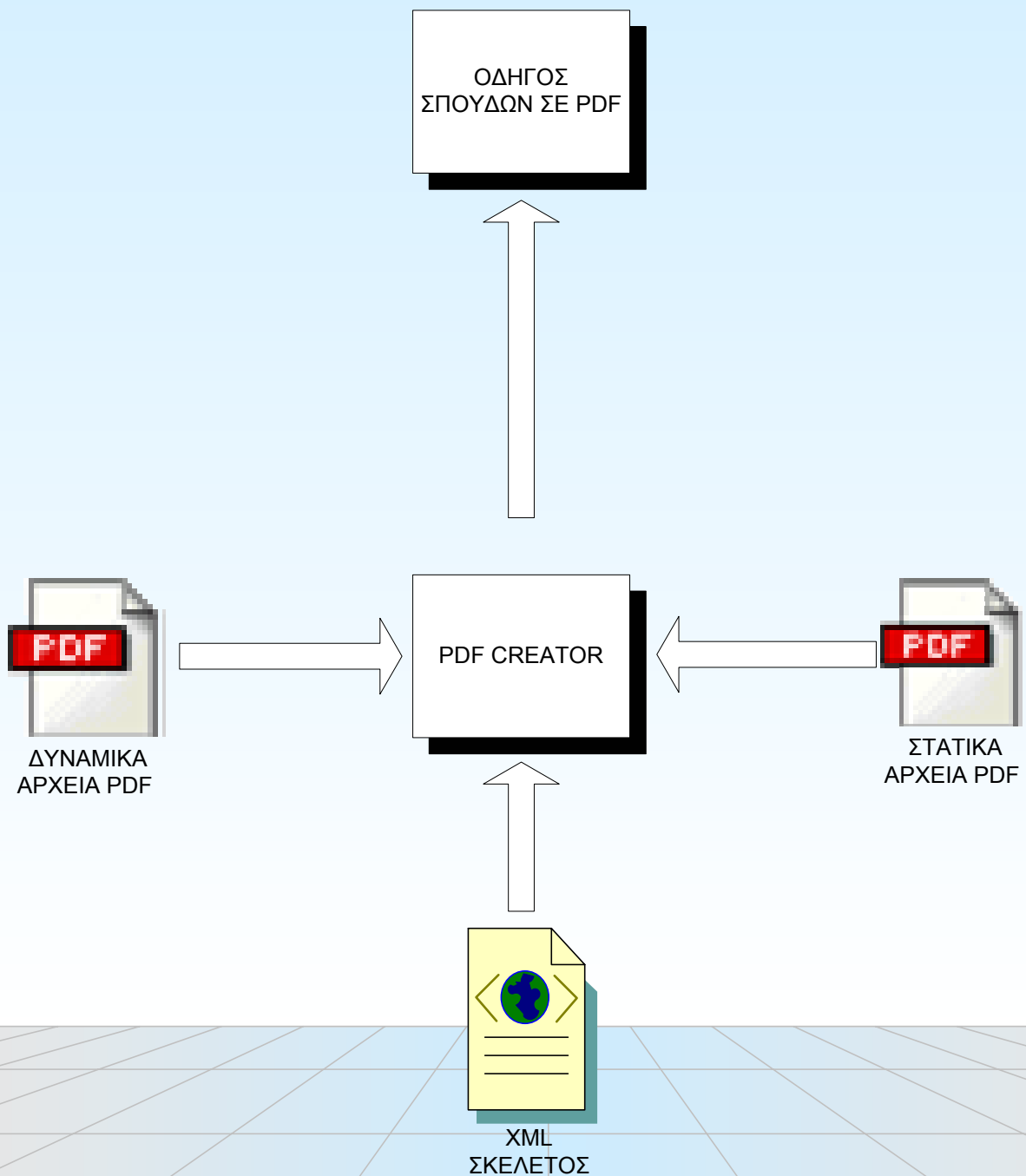
Για την δομή αυτή επιλέχθηκε η xml, για αρκετούς λόγους. Λαμβάνοντας υπ'όψη ότι ο οδηγός θα αποτελείται από κεφάλαια, τα οποία μπορεί να περιέχουν υποκεφάλαια, που με την σειρά τους άλλα υποκεφάλαια, συμπεραίνουμε ότι πρέπει να έχουμε δενδρική δομή για να περιγράψουμε τις σχέσεις πατέρα – παιδί. Κατάλληλη λοιπόν δομή για αυτό είναι η xml, όπου μέσα σε μια ετικέτα μπορούμε να συμπεριλάβουμε και άλλες ετικέτες, υποδηλώνοντας την σχέση πατέρα – παιδί.

Μόλις ο χρήστης τελειώσει με την σχεδίαση της δομής του οδηγού, πρέπει να ρυθμίσει κάποιες επιλογές για τον τρόπο εμφάνισης του αρχείου pdf του οδηγού.

Μόλις τελειώσει αυτή η εισαγωγή στοιχείων από τον χρήστη, μία κλάση (έστω PDFCreator) θα πάρει αυτά τα στοιχεία, και με τις οδηγίες που υπάρχουν στην XML δομή του οδηγού θα δημιουργήσει το τελικό αρχείο στον server.

Παρακάτω παραθέτουμε ένα σχήμα που περιγράφει συνοπτικά αυτά που αναλύσαμε στο παρόν κεφάλαιο.

ΣΥΝΘΕΣΗ ΣΤΑΤΙΚΟΥ - ΔΥΝΑΜΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ



2.6 Περιγραφή διαθέσιμων επιλογών εφαρμογής σε UML

Σε αυτήν την ενότητα περιγράφουμε πιο συγκεκριμένα τις διαθέσιμες επιλογές που έχει ο χρήστης από την εφαρμογή για να δημιουργήσει τον οδηγό σπουδών.

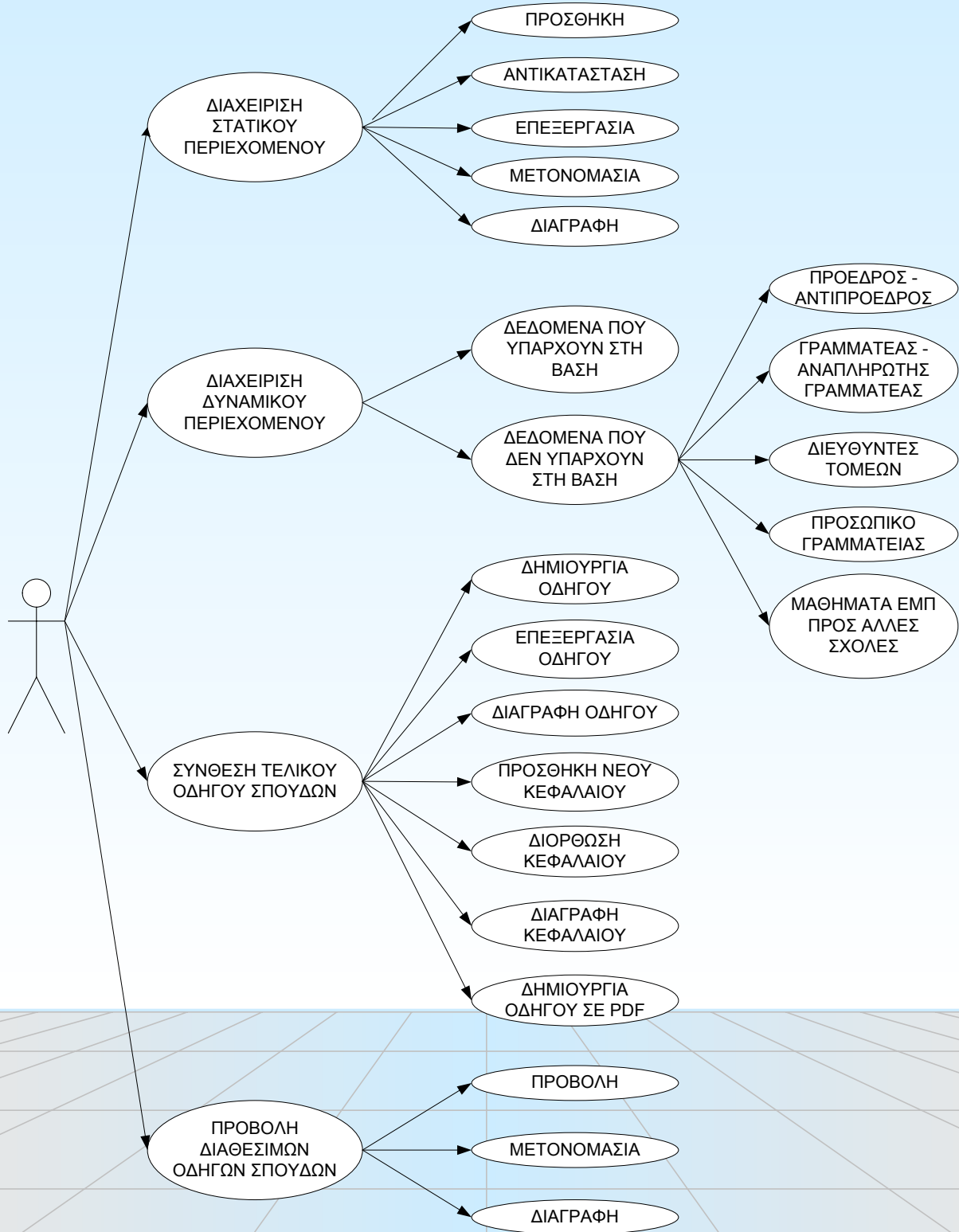
Γενικότερα, ο χρήστης έχει πρόσβαση αρχικά σε ένα μενού τεσσάρων επιλογών, σύμφωνα με τον διαχωρισμό του γενικότερου προβλήματος σε υποπροβλήματα που έγινε σε προηγούμενη ενότητα.

Σε γενικές γραμμές, ο χρήστης μπορεί σε κάθε ενότητα να κάνει τις βασικές “ενέργειες” (actions) διαχείρισης περιεχομένου (content management) στην αντίστοιχη οντότητα που διαχειρίζεται στην ενότητα αυτή.

Μπορεί κατ’ αυτόν τον τρόπο να κάνει “Δημιουργία”, “Μετονομασία”, “Διαγραφή” και “Επεξεργασία” των βασικών διαχειριζόμενων οντοτήτων, δηλαδή των Στατικών Αρχείων, των Διαθέσιμων Οδηγών, των δεδομένων που δεν υπάρχουν στην βάση δεδομένων και του σκελετού του τελικού οδηγού σπουδών.

Παρακάτω παραθέτουμε ένα UML διάγραμμα με τις διαθέσιμες επιλογές που περιγράψαμε στην ενότητα αυτή.

ΕΠΙΛΟΓΕΣ ΕΦΑΡΜΟΓΗΣ



3. Τεχνολογίες υλοποίησης

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν οι εξής τεχνολογίες:

- Γλώσσα Προγραμματισμού Java
- Java Servlets
- Adobe PDF
- Σύστημα Ελέγχου ταυτόχρονων εκδόσεων WinCVS Version 1.2 (<http://www.cvshome.org/>)
- Struts Version 1.1 (<http://jakarta.apache.org/struts>)
- XML-XSL

3.1 Γλώσσα Προγραμματισμού Java

Στην ανάπτυξη του κώδικα της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java λόγω των πλεονεκτημάτων από τα οποία χαρακτηρίζεται και τα οποία περιγράφονται στη συνέχεια. Η Java είναι μια σχετικά καινούργια γλώσσα προγραμματισμού υψηλού επιπέδου που έχει δύο κύρια χαρακτηριστικά: είναι **μεταφέρσιμη** (portable) και **αντικειμενοστρεφής** (object-oriented).

Η μεταφερσιμότητα είναι ένας όρος που χρησιμοποιείται για να χαρακτηρίσει γλώσσες προγραμματισμού που έχουν τη δυνατότητα να μεταδίδονται μέσα από ετερογενή δίκτυα και να εκτελούνται αυτόματα μόλις φτάνουν στον προορισμό τους. Τα πιο γνωστά παραδείγματα τέτοιων γλωσσών αποτελούν η Java και η Postscript. Τα πλεονεκτήματα που προσφέρει η χρήση τέτοιων γλωσσών είναι:

1. Η βελτίωση της απόδοσης των εφαρμογών αφού η επεξεργασία γίνεται τοπικά και εξαλείφει την ανάγκη επικοινωνίας με απομακρυσμένους κόμβους, κάτι που θα οδηγούσε σε χαμηλές ταχύτητες σε δίκτυα με μεγάλες καθυστερήσεις
2. Η απλοποίηση και αποκέντρωση των διαδικασιών παρακολούθησης και συντήρησης εφαρμογών στο δίκτυο μιας και ο κάθε κόμβος που αποφασίζει να χρησιμοποιήσει μια εφαρμογή είναι υπεύθυνος και για τη σωστή εγκατάστασή του στο τοπικό σύστημα.
3. Ο περιορισμός των απαιτήσεων σε αποθήκευση που έχουν προγράμματα τα οποία πρέπει να αποθηκευθούν τοπικά πριν εκτελεστούν. Στην περίπτωση των μεταφέρσιμων γλωσσών η μεταφορά του εκτελέσιμου κώδικα γίνεται μια μόνο φορά κατά τη χρησιμοποίησή του και δεν απαιτείται η δυνατότητα τοπικής αποθήκευσης του.

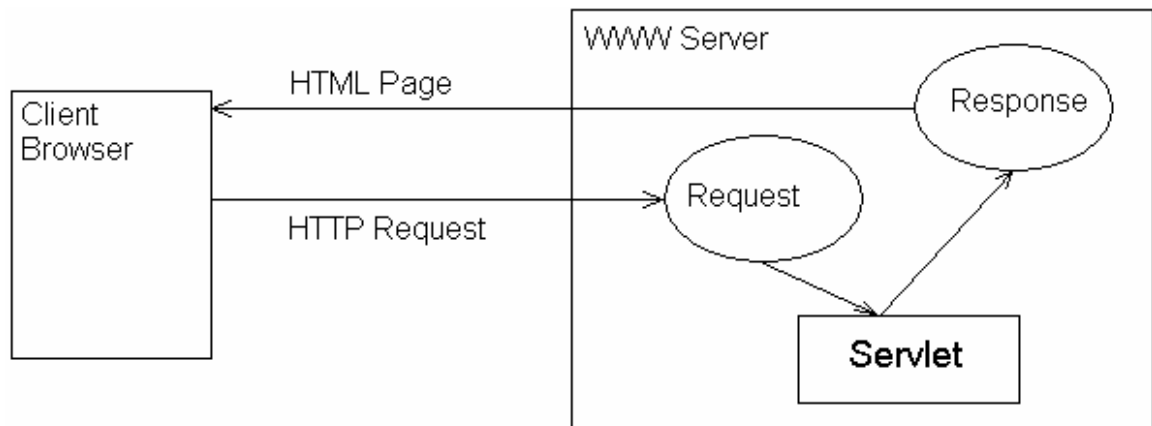
Η αντικειμενοστρέφεια, εξάλλου, είναι ένας όρος που χρησιμοποιείται για να περιγράψει τη χρήση αντικειμένων ως κύρια δομή δεδομένων στην οποία στηρίζεται η ανάπτυξη του προγράμματος. Τα αντικείμενα αυτά αποτελούνται τόσο από δεδομένα όσο και από μεθόδους που υλοποιούν βασικές λειτουργίες πάνω στα δεδομένα αυτά. Επίσης τα αντικείμενα αυτά μπορούν να συνδυάζονται σε ιεραρχικά δέντρα που αποτυπώνουν σχέσεις εγκλεισμού (υποσύνολο/υπερσύνολο) μεταξύ τους. Τα αντικείμενα μπορούν να επικοινωνούν μεταξύ τους με ανταλλαγή

μηνυμάτων (messages). Το κύριο πλεονέκτημα των αντικειμενοστρεφών γλωσσών προγραμματισμού είναι ότι προσφέρουν μια μέθοδο μοντελοποίησης των δεδομένων και των διαδικασιών που χρησιμοποιούνται για την επίλυση ενός προβλήματος που προσομοιάζει τον ανθρώπινο τρόπο οργάνωσης της πληροφορίας.

Για την υποστήριξη ανάπτυξης εφαρμογών η γλώσσα παρέχει αρκετές πλούσιες βιβλιοθήκες γνωστές ως Java APIs (Application Programming Interfaces). Οι βιβλιοθήκες περιλαμβάνουν μεταξύ των άλλων μεθόδους για την υποστήριξη δομών δεδομένων παρόμοιων με αυτές που χρησιμοποιούνται από γλώσσες γενικού σκοπού, για την δημιουργία applets, για την υλοποίηση πρωτοκόλλων ασφαλείας (Java Security), για την προσπέλαση σχεσιακών βάσεων δεδομένων (Java Database Connectivity, JDBC), για τη δικτυακή επικοινωνία με άλλους κόμβους στο δίκτυο και τέλος για την ενσωμάτωση αυτόνομων προγραμματιστικών εφαρμογών που μπορούν να συνδυαστούν με οπτικό τρόπο για την παραγωγή πιο σύνθετων εφαρμογών (JavaBeans).

3.2 Java Servlets

Τα Java servlets είναι στην ουσία μέθοδοι που υποστηρίζουν την ανάπτυξη δικτυακών εφαρμογών. Ένα servlet 'τρέχει' σε έναν web server και αλληλεπιδρά με το servlet engine που είναι εγκατεστημένο στον client μέσω των αντικειμένων request και response. Αναλυτικότερα, ο ρόλος του servlet είναι να εξετάζει τις αιτήσεις (requests) του κάθε χρήστη (client) και να στέλνει την ανάλογη απάντηση (response), δηλαδή την κατάλληλη σελίδα στον browser του χρήστη. Η λειτουργία αυτή αναπαρίσταται στο παρακάτω σχήμα



Εικόνα 3.2Α. Java Servlets

Τα servlets δημιουργούνται και καταστρέφονται κατ' απαίτηση του χρήστη. Αυτό μεταφράζεται στην ελάχιστη χρησιμοποίηση μνήμης στον www server στον οποίο είναι εγκατεστημένα.

Επόμενο βήμα της τεχνολογίας ήταν η ανάπτυξη των Java Server Pages(JSP). Οι σελίδες JSP βασίζονται στη φιλοσοφία των servlets και πρόκειται για λειτουργικά ισοδύναμα με τα servlets. Παρέχουν όμως επιπρόσθετα πλεονεκτήματα όπως:

- Γράφονται ευκολότερα καθώς διευκολύνεται η χρήση του απαιτούμενου API
- Παρέχεται η δυνατότητα Server Side Scripting, τμήματα κώδικα που εκτελούνται στον server και όχι στον client
- Ενσωματώνουν τη λειτουργικότητα στη διεπαφή για το χρήστη

Οι σελίδες JSP αντικατέστησαν τα applets τα οποία ήταν μέχρι τότε διαδεδομένα σε εφαρμογές δικτύου. Τα applets είναι προγράμματα γραμμένα σε java με περιορισμένες δυνατότητες, μεταφράζονται και εκτελούνται μέσα στον browser του χρήστη όπως ακριβώς και τα servlets. Η βασική τους διαφορά έγκειται στο ότι τα servlets προσφέρουν επιπλέον τα παρακάτω:

- Ασφάλεια (τόσο από την άποψη των Servers, όσο και από την άποψη των applets)
- Προστασία από πιθανό decompilation
- Ευκολία ανάπτυξης (ιδιαίτερα για απλές εφαρμογές)

3.3 Adobe PDF

Το PDF είναι ουσιαστικά ένα πρότυπο δημιουργημένο από την Adobe που αποσκοπεί στην αξιόπιστη και ασφαλή μεταφορά κειμένου σε ψηφιακή μορφή. Έχει τύχει ευρείας αποδοχής και αναγνώρισης, κάτι που το μαρτυρά η δεκαετής παρουσία του. Το κύριο πλεονέκτημα του PDF είναι ότι διατηρεί τις γραμματοσειρές, τα γραφικά, τις εικόνες και τη σελιδοποίηση οποιουδήποτε κειμένου ανεξάρτητα από την πλατφόρμα στην οποία αυτό δημιουργήθηκε. Τα αρχεία PDF είναι αποθηκευμένα ώστε να καταλαμβάνουν τον ελάχιστο χώρο ώστε να διευκολύνεται και η γρήγορη μεταφορά τους μέσω δικτύου. Επίσης, ένα αρχείο PDF μπορεί να το διαβάσει, να το μοιραστεί και να το τυπώσει οποιοσδήποτε διαθέτει το Adobe Reader το οποίο διατίθεται δωρεάν. Έως σήμερα πάνω από 500 εκατομμύρια αντίγραφα του Adobe Reader έχουν διανεμηθεί από το επίσημο site της Adobe.

Επιχειρήσεις, οργανισμοί και κυβερνήσεις έχουν υιοθετήσει το πρότυπο PDF για τη διευκόλυνση της διαχείρισης των κειμένων τους, για την αύξηση της παραγωγικότητας και τη χαμηλότερη κατανάλωση γραφικής ύλης.

Προβλήματα

- Οι αποδέκτες δεν έχουν τη δυνατότητα πρόσβασης στα αρχεία γιατί δεν έχουν το κατάλληλο λογισμικό
- Η αναζήτηση σε αρχεία που συνυπάρχουν σε ηλεκτρονική και έντυπη μορφή είναι χρονοβόρα.
- Τα αρχεία δεν απεικονίζονται πάντα σωστά σε φορητές συσκευές

Λύσεις από το PDF

- Το μόνο που χρειάζεται για την ανάγνωση PDF αρχείων είναι το PDF Reader της Adobe που διατίθεται δωρεάν
- Η αναζήτηση σε αρχεία PDF είναι εύκολη και γρήγορη μέσα από τον PDF Reader
- Το πρότυπο PDF υποστηρίζεται και από τις πλατφόρμες που υποστηρίζονται από φορητές συσκευές όπως το Palm OS, το Symbian αλλά και από τα Pocket PC.

Αν εξετάσουμε το πρότυπο PDF ως προς τον τρόπο με τον οποίο αποθηκεύεται, παρατηρούμε ότι η δομή του είναι παρόμοια με αυτή των XML. Ένα αρχείο PDF χρησιμοποιεί αντικείμενα για να περιγράψει το περιεχόμενό του τα οποία αποθηκεύονται σε ιεραρχική δομή. Διευκολύνεται κατ' επέκταση και ενθαρρύνεται η

χρησιμοποίηση XML αρχείων για την περιγραφή δεδομένων ενός PDF αρχείου, το οποίο μπορεί να έχει στατικό και δυναμικό περιεχόμενο.

3.4 CVS (Concurrent Version System)

Το CVS είναι ένα σύστημα ελέγχου ταυτόχρονων εκδόσεων αντικειμένου που επεξεργάζεται από ομάδα προγραμματιστών. Χρησιμεύει για το συντονισμό είτε μεμονωμένων προγραμματιστών είτε μεγάλων, διανεμημένων ομάδων. Μέσα από το CVS ο κάθε προγραμματιστής μπορεί να έχει πρόσβαση στον πιο πρόσφατο κώδικα μέσα από το internet, να τον επεξεργαστεί και να τον ανανεώσει. Το CVS αναλαμβάνει την αποφυγή συγκρούσεων μεταξύ των προγραμματιστών. Για τη λειτουργία ανάπτυξης κώδικα μέσα από το CVS προϋποτίθεται η ύπαρξη υπεύθυνου ή ομάδας υπευθύνων για το συντονισμό των προγραμματιστών.

Όπως γίνεται αντιληπτό, το CVS είναι πολύτιμο σε μια ομάδα προγραμματιστών καθώς προσφέρει λειτουργικότητα που δίνει απαντήσεις σε ερωτήματα όπως πότε έγιναν αλλαγές στον κώδικα, από ποιον και γιατί. Ταυτόχρονα, εξασφαλίζει ότι δε θα χαθεί ο κώδικας από οποιοδήποτε αστάθμητο παράγοντα καθώς αυτός διατηρείται σε έναν κεντρικό υπολογιστή και όχι τοπικά σε κάθε developer. Διατηρείται έλεγχος εκδόσεων του τελικού κώδικα, κάτι που επιτρέπει στην ομάδα να προχωρήσει στην επόμενη έκδοση αφού διαπιστωθεί η σταθερότητά της.

Κάθε χρήστης του CVS που αναπτύσσει κώδικα έχει τη δυνατότητα να 'δανειστεί' ένα αρχείο, να κάνει δηλαδή checkout το αρχείο, να το τροποποιήσει και στη συνέχεια να το αποθηκεύσει πάλι στον κεντρικό υπολογιστή, να κάνει δηλαδή checkin πάλι το αρχείο. Το CVS φροντίζει ώστε σε κάθε στιγμή να μη δίνεται η δυνατότητα μόνο σε ένα άτομο να κάνει checkout ένα αρχείο αλλά σε πολλά άτομα. Όταν οι χρήστες κάνουν checkin το αρχείο, οι πιο πρόσφατες αλλαγές δε θα επικαλύψουν τις παλιότερες. Αντίθετα, θα διατηρηθεί η ιστορία των αλλαγών.

3.5 Struts

Το Struts πρόκειται για ένα πλαίσιο εργασίας που διαχωρίζει το 'μοντέλο' από την 'όψη' μέσω ενός 'ελεγκτή' σε μια εφαρμογή διαδικτύου όπως αναλύεται στη συνέχεια. Συγκεκριμένα, διαχωρίζει τη λογική που διέπει την πλοήγηση στο δικτυακό τόπο από την παρουσίαση στην οθόνη του χρήστη των κατάλληλων πληροφοριών.

Όταν επινοήθηκαν τα java servlets σε συνδυασμό με τις σελίδες jsp παρείχαν ένα ισχυρό και ολοκληρωμένο εργαλείο για τη δημιουργία εφαρμογών διαδικτύου. Οι jsp σελίδες που παρουσιάζονται στο χρήστη έχουν τη δυνατότητα επικοινωνίας με τα αντίστοιχα servlets και σε συνεργασία να διαμορφώνουν το τελικό αποτέλεσμα. Καμία όμως λύση δεν παρέχεται σε προβλήματα όπως: έλεγχος ροής μεταξύ των σελίδων, ομοιομορφία των σελίδων, κεντρικός έλεγχος των πληροφοριών που απεικονίζονται κ.ο.κ.

Το Struts framework έρχεται να δώσει λύση σ' αυτά τα προβλήματα εισάγοντας διαχωρισμό μεταξύ των δομικών στοιχείων του site. Η λογική, η οποία καθορίζει τη συμπεριφορά του site, αποτελείται από αντικείμενα κλάσης Action τα οποία μετατρέπουν τα αιτήματα HTTP σε ενέργειες εφαρμογής. Αναλυτικότερα, με την αίτηση του χρήστη για μια σελίδα ενεργοποιείται μια Action η οποία χειρίζεται το αίτημα και επιδρά ανάλογα σε ιδιότητες της εφαρμογής όπως εγγραφές στη βάση δεδομένων ή εγγραφές στη session, ακόμη και σε ιδιότητες των javabeans της εφαρμογής. Η Action είναι στη συνέχεια υπεύθυνη για τη μεταβίβαση του ελέγχου σε άλλη σελίδα, στην κατάλληλη δηλαδή 'όψη'. Το σύνολο των ενεργειών ελέγχου αλλά και προώθησης του ελέγχου αποτελεί την επιχειρησιακή λογική της εφαρμογής, το 'μοντέλο'. Οι jsp σελίδες που καθορίζονται από το μοντέλο αποτελούν την 'όψη' της εφαρμογής. Το Struts είναι ο ελεγκτής που αποσαφηνίζει το πλαίσιο συνεργασίας των σελίδων και της αλληλεπίδρασής τους με το χρήστη αλλά και μεταξύ τους.

Γίνεται αντιληπτό ότι με τη χρήση του Struts μπορεί η φάση ανάπτυξης της εφαρμογής να διαχωριστεί μεταξύ μοντέλου και όψης. Πρακτικά, αυτό μεταφράζεται ότι μπορεί ο υπεύθυνος για τη γραφική αναπαράσταση της σελίδας να εργαστεί ανεξάρτητα από τον υπεύθυνο για τη λογική της σελίδας. Μπορεί για παράδειγμα ένας γραφίστας να σχεδιάσει μια σελίδα και να προσθέσει ένα καλάθι αγορών χωρίς να χρειάζεται να γνωρίζει τη λειτουργία του καλαθιού ή, αντίστοιχα, να σχεδιάσει μια φόρμα χωρίς να ανησυχεί για τους ελέγχους που χρειάζεται να πραγματοποιηθούν. Εκεί βρίσκεται και το πλεονέκτημα του Struts έναντι των απλών jsp, ότι η ανάπτυξη του site γίνεται χωρίς να εργάζεται ταυτόχρονα πολύς κόσμος πάνω στο ίδιο αρχείο.

3.6 XML-XSL

Η XML (Extended Mark-up Language) είναι μια δηλωτική γλώσσα που ήρθε μετά την ASN1. Κύρια διαφορά είναι ότι στην XML χρησιμοποιούνται αποκλειστικά χαρακτήρες (character based, έναντι της ASN1 που είναι bit coded). Επιπλέον, η δομή της είναι πιο απλή και πιο εύκαμπτη. Η XML χρησιμοποιείται για την αποθήκευση δομημένης πληροφορίας. Η δομή της πληροφορίας προσδιορίζεται με τη βοήθεια των tags (ετικετών) μέσα στα οποία περικλείονται συναφή τμήματα πληροφορίας.

Η αναπαράσταση της πληροφορίας σε 'ανθρώπινη' μορφή επιτυγχάνεται με τη χρήση της XSL (eXtensible Structure Language). Η XSL αναλαμβάνει να μετασχηματίσει το κείμενο του XML ανάλογα με τις απαιτήσεις ενός άλλου αποδέκτη. Πρόκειται για κείμενο γραμμένο όπως αυτό του XML πλην όμως ότι περιέχει οδηγίες για την εμφάνιση του XML. Η εμφάνιση ή αλλιώς η έξοδος του XSL μπορεί να είναι πάλι XML ή HTML ή οτιδήποτε.

Διαφανής προς το χρήστη είναι η χρησιμοποίηση ενός XML Parser που είναι υπεύθυνος για την ανάγνωση του αρχείου και την αποθήκευσή του σε μια Java κλάση τύπου Document. Προαιρετική είναι η χρήση ενός DTD αρχείου που παρέχει τους περιορισμούς στην ελευθερία του XML και εγγυάται ότι το αρχείο θα υπακούει σε κάποιες συμβάσεις όσον αφορά την ονοματοδοσία των ετικετών και των ιδιοτήτων τους αλλά και τη δομή του αρχείου. Στην περίπτωση αυτή, χρησιμοποιείται ένας Validator που εγγυάται ότι το XML περιέχει τα κατάλληλα δεδομένα για να μορφοποιηθεί με XSL. Στη συνέχεια, ένας XSLT (XSL Transformation) Processor, αναλαμβάνει την τελική μορφοποίηση του XML ώστε να παραχθεί το τελικό αποτέλεσμα.

Για τη διαχείριση των XML αρχείων παρέχονται από τη Java οι κλάσεις javax.xml.* στις οποίες συγκαταλέγονται οι XML Parsers SAX (Simple Api for XML) και DOM (Document Object Model) μαζί με πλήθος βοηθητικών κλάσεων για την παραγωγή επιθυμητού αποτελέσματος με χρήση XML και XSL. Στην εν λόγω διπλωματική εργασία, ωστόσο, έγινε χρήση του api dom4j το οποίο θεωρήθηκε πιο πλήρες όσον αφορά τις παρεχόμενες διευκολύνσεις.

Από αυτή την πληροφορία και με χρήση συναρτήσεων FOP μετασχηματίζουμε το αρχείο σε μορφή PDF. Συγκεκριμένα, οι συναρτήσεις FOP χρησιμοποιούνται για την παροχή οδηγιών για τη δημιουργία του τελικού αρχείου pdf.

4. Υλοποίηση - Τεχνική Περιγραφή

Σε αυτό το κεφάλαιο θα περιγράψουμε αναλυτικά τον τρόπο με τον οποίο δημιουργήθηκε η εφαρμογή. Πρόκειται για μία διαδικτυακή εφαρμογή που στηρίζεται στην τεχνολογία των JSPs (Java Server Pages) κυρίως, και στο Struts Framework.

Επίσης, χρησιμοποιήθηκε αρκετά η γλώσσα Javascript. Η γλώσσα αυτή χρησιμοποιείται για validation των φορμών, έτσι ώστε ο χρήστης να είναι αναγκασμένος να εισάγει σωστά στοιχεία. Βέβαια, θα μπορούσε να γίνει server side validation, αλλά κάτι τέτοιο θα καθυστερούσε αρκετά την εφαρμογή, και δεν χρειάζεται καθώς τα δεδομένα δεν είναι τόσο σημαντικά (το κακό με την javascript είναι πως κάποιος “έξυπνος” χρήστης μπορεί να παρακάμψει τον έλεγχο, και να στείλει λάθος δεδομένα).

4.1 Περιγραφή χρησιμοποιούμενων βιβλιοθηκών

Τα XML αρχεία που δημιουργήθηκαν στην εφαρμογή έγιναν με την χρήση της βιβλιοθήκης dom4j, που περιέχει τις απαραίτητες συναρτήσεις για την διαχείριση των αρχείων XML. Επιπλέον, αρκετά χρήσιμη στην δημιουργία της εφαρμογής φάνηκε και η γλώσσα ερωτημάτων για XML, η Xpath (περισσότερες πληροφορίες www.w3c.org). Η xpath είναι μια γλώσσα με την οποία μπορεί κανείς να επιλέξει κάποιους συγκεκριμένους κόμβους σε ένα xml αρχείο, ανάλογα με τον τρόπο που θα διατυπωθεί το ερώτημα.

Η διαχείριση των pdf αρχείων στην εφαρμογή έγινε με 2 βιβλιοθήκες : την βιβλιοθήκη iText και την βιβλιοθήκη for. Η iText χρησιμοποιήθηκε για την συνένωση των διαφορετικών pdf για την δημιουργία του τελικού οδηγού σπουδών (στο “Σύνθεση Τελικού Οδηγού σπουδών”), και η βιβλιοθήκη for χρησιμοποιείται για την δημιουργία των pdf από δυναμικά δεδομένα που υπάρχουν στην βάση. Με την βιβλιοθήκη αυτή, τα δεδομένα από την βάση μεταφέρονται σε αρχεία xml, και με την χρήση αρχείων xsl παράγονται τα απαραίτητα pdf αρχεία.

Μια άλλη βιβλιοθήκη που χρησιμοποιήθηκε είναι η oreily, και αυτή χρησιμοποιήθηκε για το upload των αρχείων στον server.

Μαζί με τις προαναφερθείσες βιβλιοθήκες, δημιουργήθηκαν και κάποιες άλλες, οι οποίες χρησιμοποιήθηκαν από την εφαρμογή :

- **XmlUtils.java**

Η βιβλιοθήκη αυτή περιέχει τις απαραίτητες συναρτήσεις για συγχρονισμό των αρχείων xml με τα δεδομένα της βάσης, συναρτήσεις για την δημιουργία νέου κεφαλαίου και την διαγραφή ήδη υπάρχοντος που χρησιμοποιείται στην Σύνθεση του τελικού οδηγού Σπουδών, και συναρτήσεις για την εγγραφή αρχείων XML στον δίσκο του server.

- **fileUtils.java**

Στην βιβλιοθήκη αυτή περιέχονται οι απαραίτητες συναρτήσεις για την διαχείριση αρχείων στον server (δημιουργία κενών, διαγραφή ήδη υπαρχόντων, μετονομασία και προβολή περιεχομένων ενός καταλόγου).

- **NavigateXmlUtils.java**

Εδώ έχουν δημιουργηθεί οι απαραίτητες συναρτήσεις για την δημιουργία του δέντρου στην “Σύνθεση τελικού οδηγού Σπουδών”. Περιέχονται συναρτήσεις που επιστρέφουν το βάθος ενός κόμβου σε xml, δοσμένης της XPath expression του, συναρτήσεις που επιστρέφουν την λίστα των κόμβων του xml σε σωστή σειρά για να τυπωθούν, και άλλες συναρτήσεις που επιστρέφουν τον κώδικα HTML που χρειάζεται για να απεικονιστούν σωστά οι κόμβοι αυτοί (ανάλογα εάν ο κόμβος έχει και άλλα παιδιά ή είναι μόνο “φύλλο”)

4.2 Περιγραφή “Διαχείρισης Δυναμικού Περιεχομένου”

Αυτό το μέρος αποτελείται από μία jsp φόρμα που έχει δημιουργηθεί με την τεχνολογία Struts, την extractDbData.jsp. Μόλις αυτή η φόρμα γίνει submit, εκτελείται κώδικας που υπάρχει στην extractDbDataAction.java και ανάλογα με τις επιλογές εκτελούνται συναρτήσεις της βιβλιοθήκης XmlUtils.java που ενημερώνουν τα απαραίτητα κεφάλαια.

Οι συναρτήσεις αυτές ανοίγουν σύνδεση με την βάση δεδομένων της σχολής, κάνουν τα απαραίτητα SQL ερωτήματα εκεί, και αποθηκεύουν τα αποτελέσματα σε μία δομή XML στην μνήμη (ένα DOM Document). Κατόπιν η δομή αυτή εγγράφεται σε ένα XML αρχείο στον server στο path pdfManager/data της εφαρμογής.

Τα xml αρχεία είναι τα chap5.xml, chap7-9.xml, chap8-10.xml ανάλογα με τα κεφάλαια του οδηγού. Όλα τα απαραίτητα δυναμικά δεδομένα στα xml αυτά είναι αποθηκευμένα μέσα στον κόμβο database/database_data. Τα κεφάλαια 7-9 και 8-10 είναι αποθηκευμένα μαζί, λόγω της ομοιότητας της πληροφορίας που περιέχουν.

Τέλος, με την χρήση της συνάρτησης PdfCreator και των αντίστοιχων αρχείων xsl, το xml μετασχηματίζεται σε pdf και αποθηκεύεται και αυτό στον server.

4.3. Περιγραφή “Διαχείρισης Δυναμικού Περιεχομένου που δεν περιέχεται στην βάση”

Όπως έχει προαναφερθεί σε προηγούμενη παράγραφο, για την δημιουργία των δυναμικών κεφαλαίων του οδηγού σπουδών είναι απαραίτητη επιπλέον πληροφορία, που δεν περιέχεται στην βάση δεδομένων. Κατόπιν τούτου, δημιουργήσαμε την σελίδα διαχείρισης αυτού του περιεχομένου, την editNonDbData.jsp. Η σελίδα αυτή συλλέγει τα δεδομένα και τα αποθηκεύει στα xml που έχουν δημιουργηθεί από την extractDbData.jsp σε συγκεκριμένα μέρη.

Συγκεκριμένα, αποθηκεύονται στους κόμβους database/addinfo των αρχείων chap8-10.xml και chap5.xml.

Η σελίδα αυτή με το κουμπί Submit κάνει post τα δεδομένα στον εαυτό της ,κατόπιν τα δεδομένα συλλέγονται, τα xml που περιέχουν την προς αντικατάσταση πληροφορία περνούν στην μνήμη σε μορφή DOM Document, και με XPATH επιλέγονται τα δεδομένα αυτά και αντικαθίστανται. Τέλος τα DOM Documents εγγράφονται στον υπολογιστή με την συνάρτηση writeXml της βιβλιοθήκης XmlUtils.

4.4 Περιγραφή “Σύνθεσης Τελικού Οδηγού Σπουδών”

Αυτό το μέρος αποτελείται από ένα HTML frameset, το guideEditor.jsp που έχει 3 σελίδες :

Την tree.jsp που είναι η αριστερή σελίδα και είναι υπεύθυνη για την προβολή σε δέντρο των περιεχομένων ενός συγκεκριμένου οδηγού σπουδών.

Την mainFrame.jsp στην οποία παρουσιάζεται προεπισκόπηση σε pdf του επιλεγμένου κεφαλαίου του χρήστη.

Την bottomFrame.jsp που περιέχει το μενού των επιλογών του χρήστη, και μία φόρμα HTML με κάποια “κρυφά” πεδία που περιέχουν απαραίτητη πληροφορία για τις επιλογές του χρήστη (πχ ποιον οδηγό επεξεργάζεται εκείνη την στιγμή ο χρήστης, ποιο κεφάλαιο έχει επιλέξει εκείνη την στιγμή, κα)

Η tree.jsp δείχνει τα περιεχόμενα ενός xml αρχείου που έχει την μορφή του παρακάτω αρχείου odhgos.xml. Συντάσσεται tree.jsp?fname="όνομα αρχείου".

```
<?xml version="1.0" encoding="ISO-8859-7"?>
```

```

<root>
  <chapter type="file" source="chap4.pdf">
    <chaptertitle>kef5</chaptertitle>
    <chapter type="file" source="chap4.pdf">
      <chaptertitle>kef6</chaptertitle>
      <chapter type="file" source="chap1.pdf">
        <chaptertitle>kef9</chaptertitle>
        <chapter type="bookmark" page="4" father="chap1.pdf">
          <chaptertitle>kef9sel4</chaptertitle>
        </chapter>
        <chapter type="bookmark" page="7" father="chap1.pdf">
          <chaptertitle>kef9sel7</chaptertitle>
        </chapter>
        <chapter type="bookmark" page="12" father="chap1.pdf">
          <chaptertitle>kef23</chaptertitle>
        </chapter>
      </chapter>
    <chapter type="file" source="chap4.pdf">
      <chaptertitle>kef67</chaptertitle>
      <chapter type="file" source="chap1.pdf">
        <chaptertitle>aaaa</chaptertitle>
      </chapter>
    </chapter>
  </chapter>
</chapter>
<chapter type="file" source="chap2.pdf">

<chaptertitle>&#954;&#949;&#966;&#940;&#955;&#945;&#953;&#959;9</chap
tertitle>
  </chapter>
</root>

```

Αρχείο odhgos.xml

Το αρχείο πρέπει να περιέχει τα δεδομένα του μέσα στο κόμβο root. Κάθε κόμβος που έχει το όνομα chapter είναι κεφάλαιο ή υποκεφάλαιο. Ο τίτλος του κάθε κεφαλαίου περιέχεται στο κόμβο chaptertitle σε text μορφή και είναι άμεσο παιδί του αντίστοιχου chapter.

Ο τύπος του κεφαλαίου, εάν είναι δηλαδή αρχείο pdf ή σελιδοδείκτης, προσδιορίζεται από τη ιδιότητα type του κόμβου chapter. Εάν το κεφάλαιο είναι τύπου file, μία άλλη ιδιότητα με το όνομα source προσδιορίζει το όνομα του αρχείου

pdf. Εάν είναι bookmark, μία ιδιότητα με το όνομα page προσδιορίζει την σελίδα του σελιδοδείκτη, και μια άλλη με το όνομα father το pdf αρχείο στο οποίο υπάρχει αυτός ο σελιδοδείκτης.

Για την εμφάνιση τώρα αυτού του αρχείου με τον δυναμικό τρόπο χρησιμοποιήθηκε κώδικας javascript που βρίσκεται στην βιβλιοθήκη foldernav.js. Για να λειτουργήσει αυτός ο κώδικας, το παραπάνω xml έπρεπε να μετασχηματιστεί σε κώδικα HTML. Η tree.jsp δημιουργεί δυναμικά τον κώδικα αυτό χρησιμοποιώντας συναρτήσεις της βιβλιοθήκης NavigateXmlUtils.java.

Για την σωστή απεικόνιση του αρχείου xml, απαραίτητη ήταν μια λίστα με τους κόμβους με όνομα chapter του αρχείου xml, αλλά με τέτοια σειρά έτσι ώστε κάθε πατέρας να έχει από κάτω τα παιδιά του. Η λίστα αυτή παρέχεται από την συνάρτηση getXpathList της βιβλιοθήκης NavigateXmlUtils. Πιο συγκεκριμένα, σε αυτή την συνάρτηση γίνεται ένα ερώτημα xpath στο αρχείο xml, με την εντολή List elems = doc.selectNodes("//chapter"). Το xpath "//chapter" επιστρέφει μια λίστα με όλους τους κόμβους που έχουν όνομα chapter στο συγκεκριμένο xml. Το πρόβλημα εδώ ήταν ότι οι κόμβοι στην λίστα ήταν ταξινομημένοι ανάλογα με το βάθος τους (ήταν δηλαδή όλοι οι κόμβοι επιπέδου 1, μετά αυτοί επιπέδου 2, κλπ) κάτι που δεν το θέλαμε. Λύση σε αυτό το πρόβλημα δόθηκε με την εντολή Collections.sort(xpathList), η οποία τοποθετεί τα στοιχεία της λίστας κατά φθίνουσα αλφαβητική σειρά, κάτι που είχε σαν αποτέλεσμα την σωστή τοποθέτηση των κόμβων αυτών.

Επίσης, όπως μπορεί κανείς να παρατηρήσει στο αρχείο xml δεν υπάρχουν ελληνικοί χαρακτήρες, και στην θέση τους είναι χαρακτήρες της μορφής κ Αυτό συνέβη καθώς υπήρχε πρόβλημα με την εγγραφή ελληνικών χαρακτήρων σε αρχείο στον server, καθώς χρησιμοποιούσε το λειτουργικό σύστημα Linux. Λύση σε αυτό δόθηκε με την ρύθμιση του εγγραφέα xml να μην γράφει χαρακτήρες στο αρχείο xml η δεκαδική τιμή των οποίων είναι μεγαλύτερη του 255 (οι ελληνικοί είναι πάνω από αυτόν τον αριθμό). Αντ' αυτού, τους μετατρέπει στο είδος που είδαμε προηγουμένως και μετά κατά την ανάγνωση κάνει εσωτερικά μόνος του την μετατροπή.

4.5 Περιγραφή “Διαχείρισης Στατικού Περιεχομένου”

Σε αυτό το μέρος χρησιμοποιήθηκαν συναρτήσεις της βιβλιοθήκης fileUtils για την διαχείριση των αρχείων στο server. Επίσης χρησιμοποιήθηκαν συναρτήσεις από την βιβλιοθήκη oreilly, για την διαχείριση των αρχείων που γίνονταν upload στον server, στις επιλογές Προσθήκη και αντικατάσταση του κυρίως μενού.

Το μέρος αποτελείται από ένα html frameset, το fileManager.jsp, που έχει τις σελίδες pdfList.jsp(αριστερά) docList.jsp(δεξιά) και την pdfDocMenu.jsp στο κάτω μέρος που περιέχει το μενού.

Οι δυο σελίδες αριστερά και δεξιά στην ουσία παρουσιάζουν τα περιεχόμενα των καταλόγων pdfManager/pdfs και pdfManager/msdocs αντίστοιχα. Για αυτόν τον σκοπό χρησιμοποιήθηκαν οι συναρτήσεις getPdfDirContents και getDocDirContents της βιβλιοθήκης fileUtils αντίστοιχα.

Το αρχείο pdfDocMenu.jsp περιέχει τα απαραίτητα κουμπιά για την διαχείριση των αρχείων, καθώς και κάποιες συναρτήσεις javascript που εκτελούν με το πάτημα του αντίστοιχου κουμπιού μια σελίδα στον server.

5. Οδηγός χρήσης της εφαρμογής

Η πρόσβαση στην εφαρμογή γίνεται μόνο από εξουσιοδοτημένους χρήστες στην δικτυακή διεύθυνση <http://www.ece.ntua.gr/update>. Εκεί, αφού δώσουν τα απαραίτητα στοιχεία, εισέρχονται στο κεντρικό μενού της εφαρμογής, όπου επιλέγουν “Διαχείριση Οδηγού Σπουδών” και εισέρχονται στο κυρίως μενού της εφαρμογής.

Προσωπική σελίδα μέλους Δ.Ε.Π.

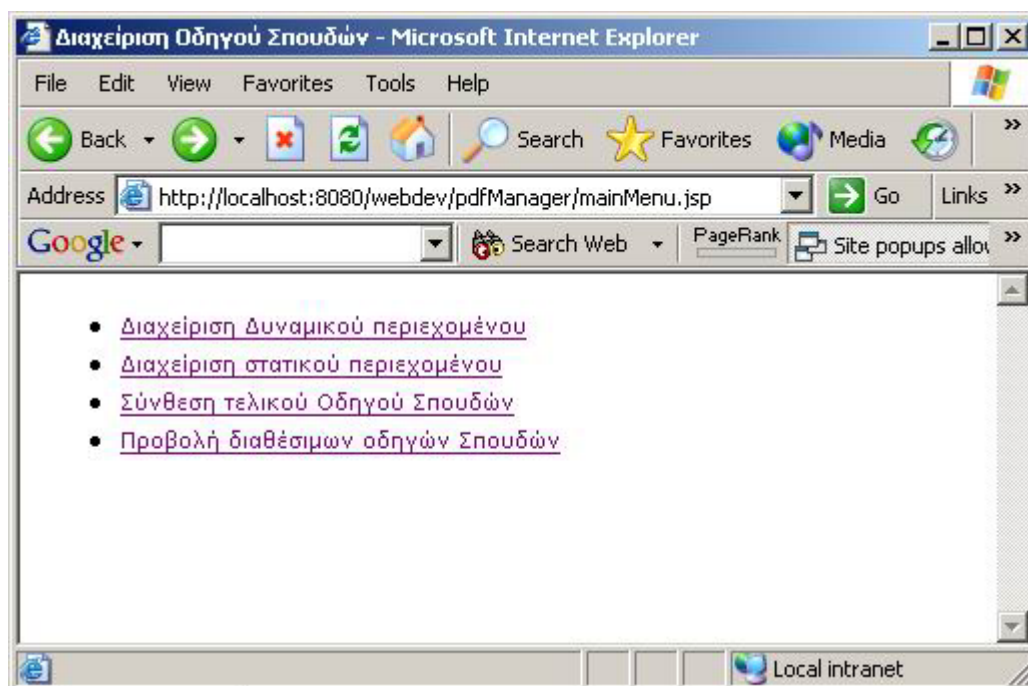
- [Προσωπικά Στοιχεία](#)
- [Ακαδημαϊκές Πληροφορίες](#)
- [Εργαστήρια](#)
- [Γραφεία](#)
- [Μαθήματα](#)
- [Ερευνητικά - Αναπτυξιακά Έργα](#)
- [Δημοσιεύσεις](#)
- [Πίνακας Ανακοινώσεων](#)
- [Εργασίες Μαθημάτων](#)
- [Αλλαγή Αναγνωριστικού/Κωδικού](#)
- [User & Group Management](#)
- [Διαχείριση Οδηγού Σπουδών](#)

[English](#)

[Αποσύνδεση](#)

5.1 Κυρίως Μενού

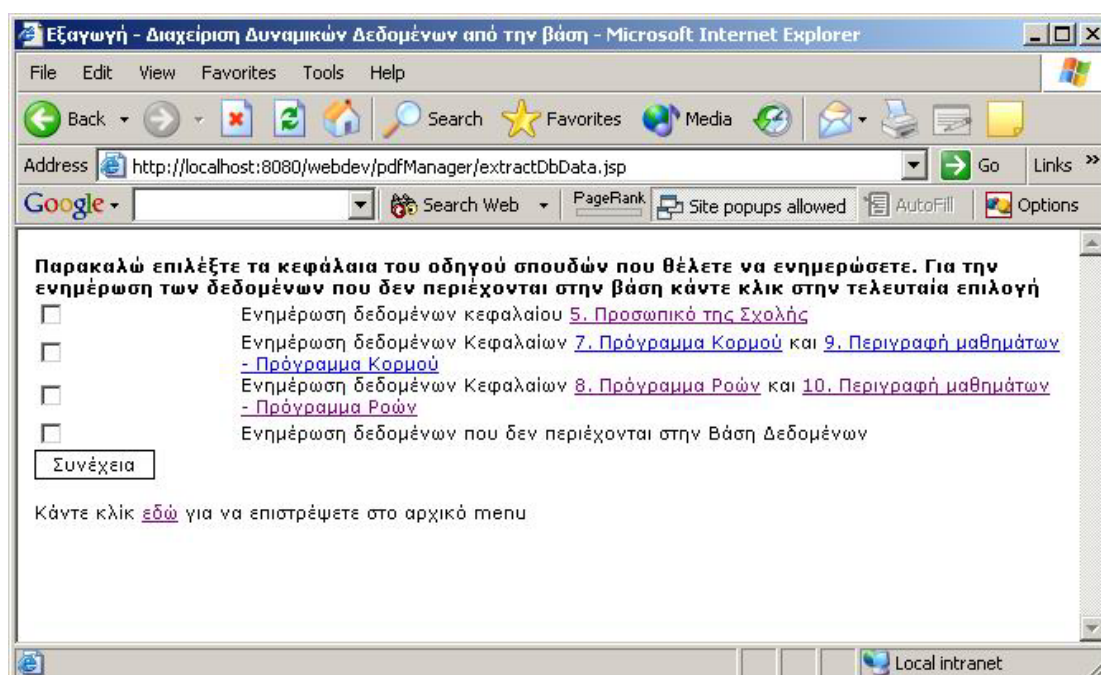
Σε αυτήν εδώ την φάση ο χρήστης έχει 4 επιλογές, όπως φαίνεται στην εικόνα 5.1A. Οι 4 αυτές επιλογές αποτελούν τα 4 μέρη από τα οποία αποτελείται η εφαρμογή, και αναπτύσσονται διεξοδικά στις παρακάτω υποενότητες του κεφαλαίου αυτού. Η τμηματοποίηση της εφαρμογής σε αυτά τα 4 μέρη έγινε για διευκόλυνση του χρήστη, έτσι ώστε οι δυνατότητές της να είναι προσβάσιμες σε αυτόν εύκολα και γρήγορα.



Εικόνα 5.1.A

5.2 Διαχείριση Δυναμικού Περιεχομένου

Σε αυτήν την ενότητα ο χρήστης έχει 4 επιλογές. Στις 3 πρώτες επιλογές ο χρήστης μπορεί να ενημερώσει τα ήδη υπάρχοντα αρχεία pdf των κεφαλαίων του οδηγού που φαίνονται στις αντίστοιχες υπερσυνδέσεις (links), και στην 4η επιλογή ο χρήστης μπορεί να αλλάξει κάποια στοιχεία απαραίτητα για την δημιουργία των παραπάνω κεφαλαίων που όμως δεν περιέχονται στην βάση δεδομένων της σχολής (πχ Πρόεδρος Σχολής, Προσωπικό γραμματείας, κα)

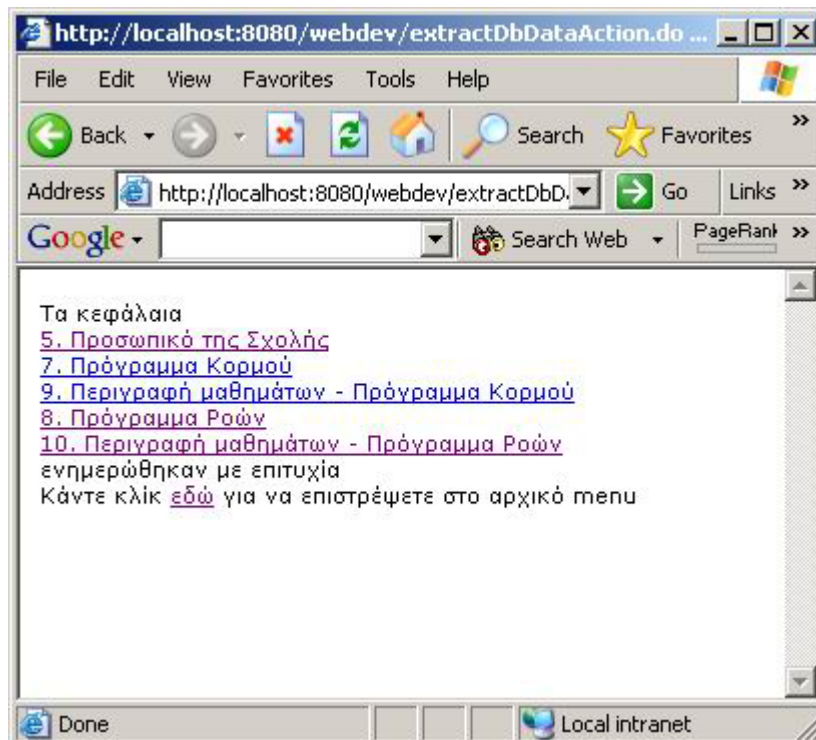


Σχήμα 5.2.A

5.2.1 Ενημέρωση δεδομένων κεφαλαίων 5,7,8,9,10

Επιλέγοντας τις αντίστοιχες επιλογές του Σχήματος 5.2.A τα αντίστοιχα .pdf ενημερώνονται με τις αλλαγές, αποθηκεύονται στον Server της Σχολής, και είναι διαθέσιμα για προεπισκόπηση από τον χρήστη, κάνοντας κλικ στις αντίστοιχες υπερσυνδέσεις. Ο τρόπος που γίνεται αυτή η ανανέωση προγραμματιστικά θα αναλυθεί σε επόμενο κεφάλαιο.

Κατ' αυτόν τον τρόπο, ο χρήστης κάθε φορά που γίνεται μια αλλαγή στα στοιχεία της βάσης δεδομένων, πχ προστίθεται ένα νέο μάθημα, ένας νέος καθηγητής, ή κάτι άλλο, το μόνο που έχει να κάνει είναι να ενημερώσει τα αντίστοιχα κεφάλαια με ένα απλό κλικ στις επιλογές του Σχήματος 5.2.A.

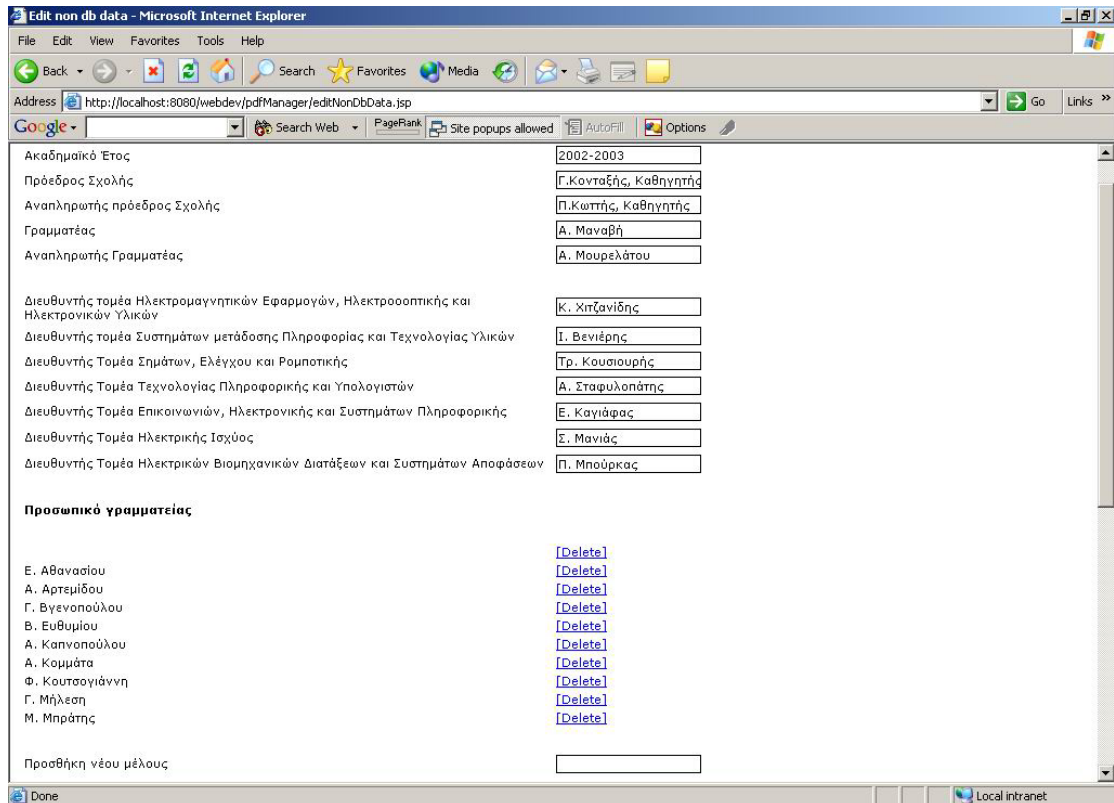


5.2.2 Ενημέρωση δεδομένων που δεν περιέχονται στην βάση δεδομένων

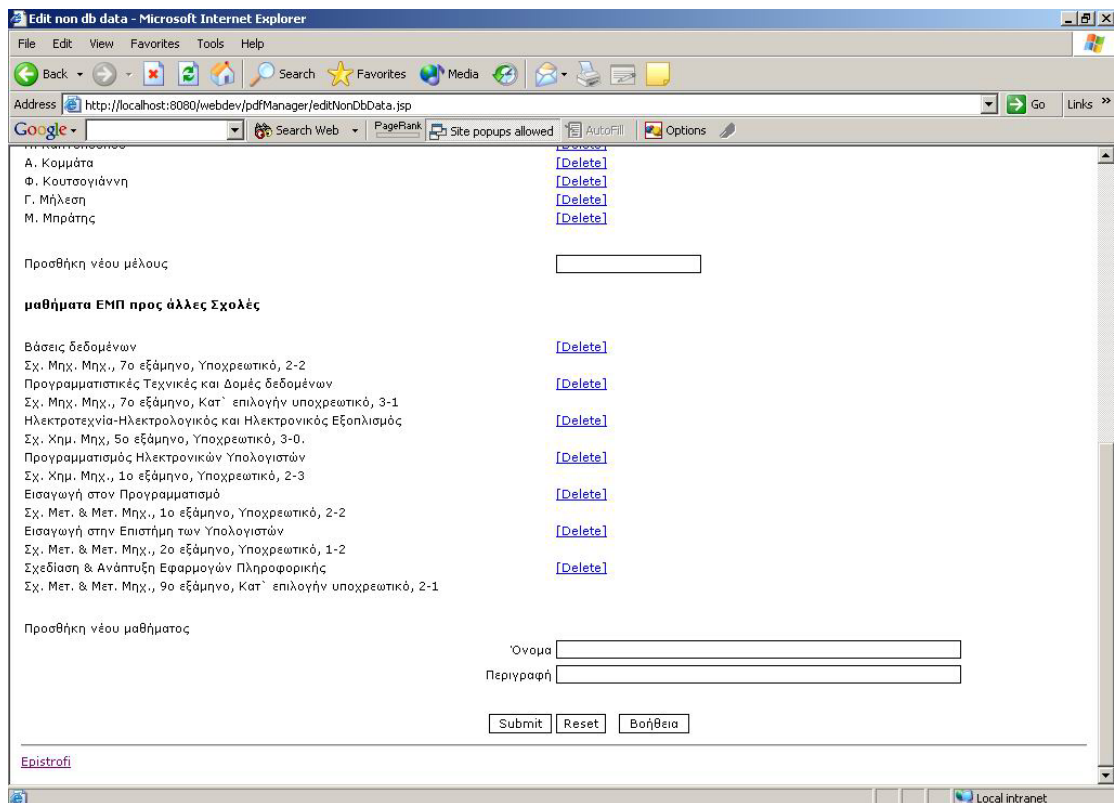
Σε αυτό το μέρος ο χρήστης διαχειρίζεται κάποια δεδομένα που είναι απαραίτητα για την δυναμική δημιουργία των παραπάνω κεφαλαίων, και τα οποία δεν υπάρχουν καταχωρημένα στην βάση της σχολής. Πιο συγκεκριμένα, σε αυτή την σελίδα ο χρήστης μπορεί να αλλάξει :

- Το Ακαδημαϊκό έτος του Οδηγού Σπουδών.
- Το όνομα του Προέδρου της Σχολής.
- Το όνομα του Αντιπροέδρου της Σχολής.
- Το όνομα του Γραμματέα της Σχολής.
- Το όνομα του Αναπληρωτή Γραμματέα της Σχολής.
- Τους Διευθυντές των τομέων της Σχολής.
- Το προσωπικό της Γραμματείας.
- Τα μαθήματα ΕΜΠ προς άλλες Σχολές.

Για να αλλάξει ο χρήστης κάποιο από τα παραπάνω στοιχεία το μόνο που έχει να κάνει είναι να συμπληρώσει το αντίστοιχο πεδίο και να κάνει κλικ στο κουμπί Submit. Εάν δεν θέλει να κάνει αλλαγές μπορεί να κάνει κλικ στο κουμπί Reset, και εάν θέλει βοήθεια μπορεί να κάνει κλικ στο κουμπί Βοήθεια.



5.2.2.A



5.2.2.B

5.3 Διαχείριση Στατικού Περιεχομένου

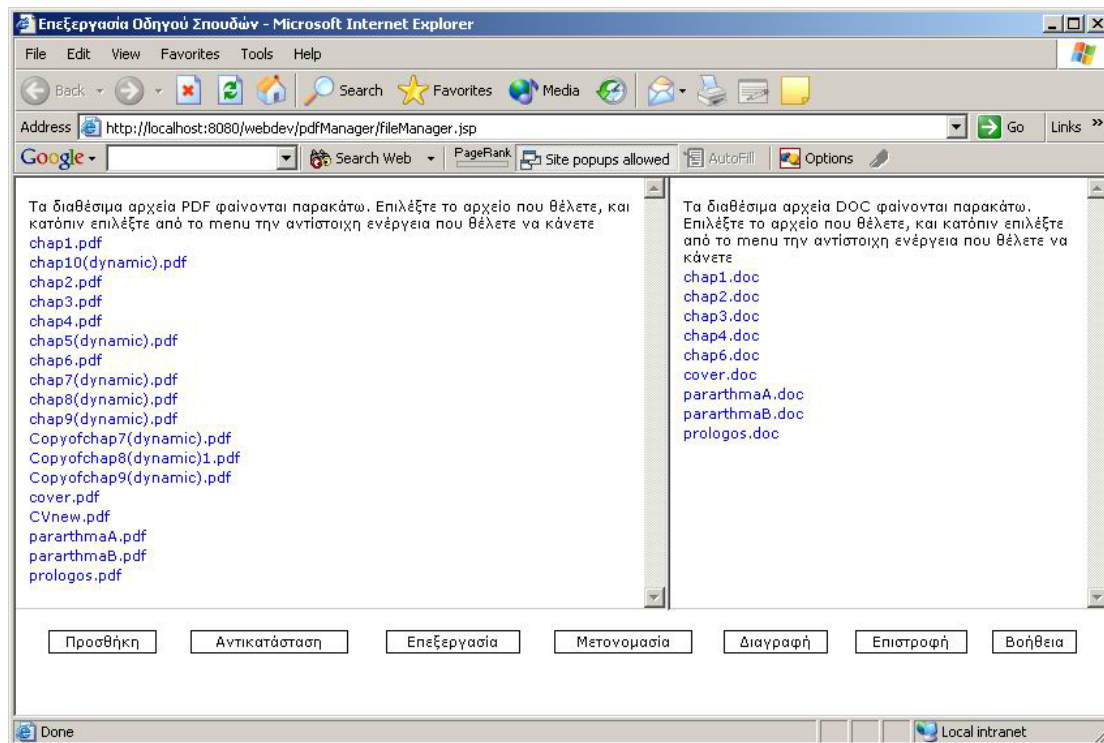
Σε αυτή τη σελίδα ο χρήστης μπορεί να έχει πλήρη έλεγχο και να διαχειριστεί το στατικό περιεχόμενο του οδηγού Σπουδών. Το στατικό περιεχόμενο χωρίζεται σε 2 μέρη, στα αρχεία .doc και στα .pdf, και γι' αυτό άλλωστε διαχωρίζεται το παράθυρο σε 2 μέρη.

Εδώ ο χρήστης μπορεί να μετονομάσει - διαγράψει - αντικαταστήσει - επεξεργαστεί όλα τα αρχεία που υπάρχουν στον Server της σχολής με ένα σχετικά εύκολο τρόπο. Το μόνο που ο χρήστης έχει να κάνει, είναι να επιλέξει το αρχείο που θέλει, (κάνοντας κλικ γίνεται κόκκινο) και κατόπιν να επιλέξει από το μενού την ενέργεια που θέλει να κάνει.

Την σελίδα αυτή ο χρήστης θα την χρησιμοποιήσει περισσότερο για να αλλάξει κάποια κείμενα που υπάρχουν στον server σε μορφή doc, και να δημιουργήσει τα αντίστοιχα .pdf. Σε παρακάτω κεφάλαιο θα περιγράψουμε αναλυτικότερα ένα τέτοιο σενάριο, βήμα - βήμα και έτσι θα γίνει ποιο κατανοητή η λειτουργικότητά της.

Ένα πράγμα που ο χρήστης πρέπει να έχει κατά νου, και το οποίο δεν μπορεί να ελεγχθεί από την εφαρμογή είναι να συμβαδίζουν τα .doc αρχεία με τα αντίστοιχα .pdf που υπάρχουν στον server, αλλιώς η σελίδα αυτή δεν έχει νόημα. Πρέπει δηλαδή ένας χρήστης όταν ενημερώνει ένα .doc αρχείο να ενημερώνει και το αντίστοιχο .pdf στον server (με αντικατάσταση του παλαιότερου με το καινούριο αρχείο).

Παρακάτω περιγράψουμε τις λειτουργίες των κουμπιών του μενού.



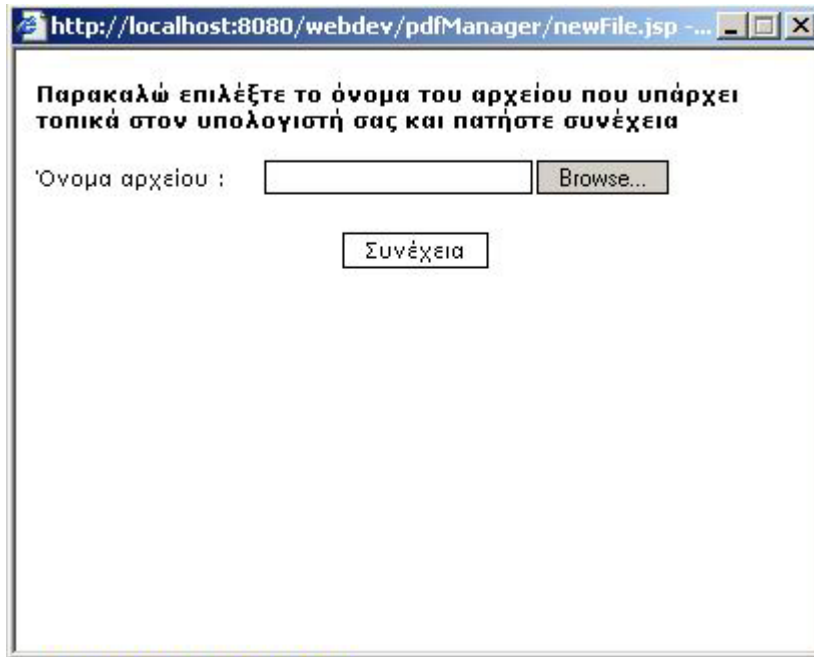
5.3.1 Προσθήκη

Με αυτήν την επιλογή ο χρήστης έχει την δυνατότητα να καταχωρήσει (upload) στον server της σχολής ένα νέο αρχείο από τον υπολογιστή του. Κάνοντας κλικ στο κουμπί Browse ο χρήστης μπορεί να επιλέξει το αρχείο που θέλει και να πατήσει συνέχεια.

Η εφαρμογή ανάλογα με την κατάληξη του αρχείου τοποθετεί το αρχείο αυτό σε αντίστοιχο κατάλογο στον server (ανάλογα δηλαδή στον κατάλογο των pdf ή των doc).

Καλό θα ήταν εδώ να αναφέρουμε πως η διαδικασία της προσθήκης μπορεί να είναι ιδιαίτερα χρονοβόρα διαδικασία, ανάλογα με το μέγεθος του αρχείου που θα ανεβάσουμε, και το είδος σύνδεσης που έχουμε στο Internet. Η εφαρμογή θέτει ένα όριο στο μέγεθος των αρχείων που μπορούν να ανέβουν στα 10 MB το καθένα.

Βέβαια, εάν ο χρήστης συνδεθεί στην εφαρμογή με το ταχύτατο τοπικό δίκτυο του πολυτεχνείου δεν θα υπάρξουν ιδιαίτερα προβλήματα. Σε περίπτωση όμως που κάτι τέτοιο δεν είναι εφικτό, μπορεί να γίνει και σύνδεση και με dial-up με μειωμένη όμως ταχύτητα.



Σχήμα 5.3.1A

5.3.2 Αντικατάσταση

Με αυτήν την επιλογή ο χρήστης μπορεί να αντικαταστήσει ένα συγκεκριμένο αρχείο στον server (pdf ή doc). Έστω πχ ότι ο χρήστης θέλει να αντικαταστήσει το αρχείο char4.doc με την ανανεωμένη έκδοσή του. Τότε, το μόνο που πρέπει να κάνει είναι να επιλέξει το αρχείο αυτό (το όνομά του τότε γίνεται κόκκινο) και να κάνει κλικ στο Αντικατάσταση.

Τότε εμφανίζεται ένα παράθυρο σαν αυτό του σχήματος 5.3.1.A όπου ο χρήστης υποδεικνύει που βρίσκεται η ανανεωμένη έκδοση του char4.doc στον τοπικό του υπολογιστή, και κάνει κλικ στο συνέχεια.

Προσοχή πρέπει να δώσει ο χρήστης στο γεγονός ότι ακόμα και αν τοπικά στον υπολογιστή του το αρχείο έχει διαφορετικό όνομα (πχ char4updated.doc) στον server θα αποθηκευτεί με το όνομα του προς αντικατάσταση αρχείου (στην προκειμένη περίπτωση char4.doc).

5.3.3 Επεξεργασία

Με αυτήν εδώ την επιλογή ο χρήστης έχει την δυνατότητα να “κατεβάσει” τοπικά στον υπολογιστή του αρχεία από τον server και να τα επεξεργαστεί.

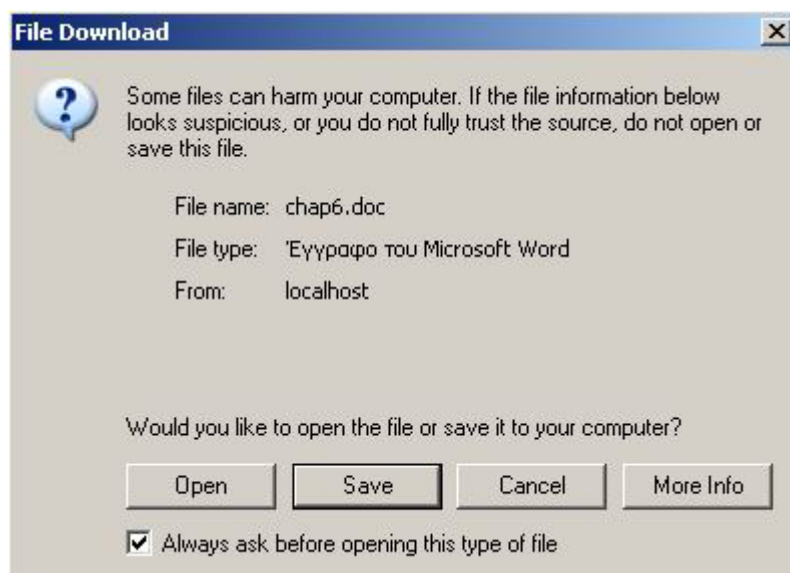
Παρόλο που δίνεται η δυνατότητα και για κατέβασμα pdf αρχείων, δεν έχει ιδιαίτερο νόημα αυτό. Αυτή η επιλογή είναι για τα αρχεία doc, τα οποία μπορούν να

υποστούν επεξεργασία, σε αντίθεση με τα pdf που μπορεί κάποιος μόνο να τα διαβάσει και να τα τυπώσει.

Ο χρήστης κάνει κλικ στο αρχείο doc που θέλει να επεξεργαστεί και κατόπιν στο κουμπί “Επεξεργασία”. Εδώ πρέπει να αναφέρουμε πως ο χρήστης πρέπει να έχει εγκατεστημένο στον υπολογιστή του το πρόγραμμα MS Word της Microsoft (μέρος του MS OFFICE) για να μπορέσει να επεξεργαστεί τα αρχεία αυτά.

Κάνοντας κλικ ο χρήστης, ανοίγει ένα νέο παράθυρο με ένα μήνυμα ίδιο ή παραπλήσιο με αυτό του σχήματος 5.3.3A (ανάλογα με την έκδοση της εφαρμογής MS Word του χρήστη και τις ρυθμίσεις ασφαλείας του Internet Explorer).

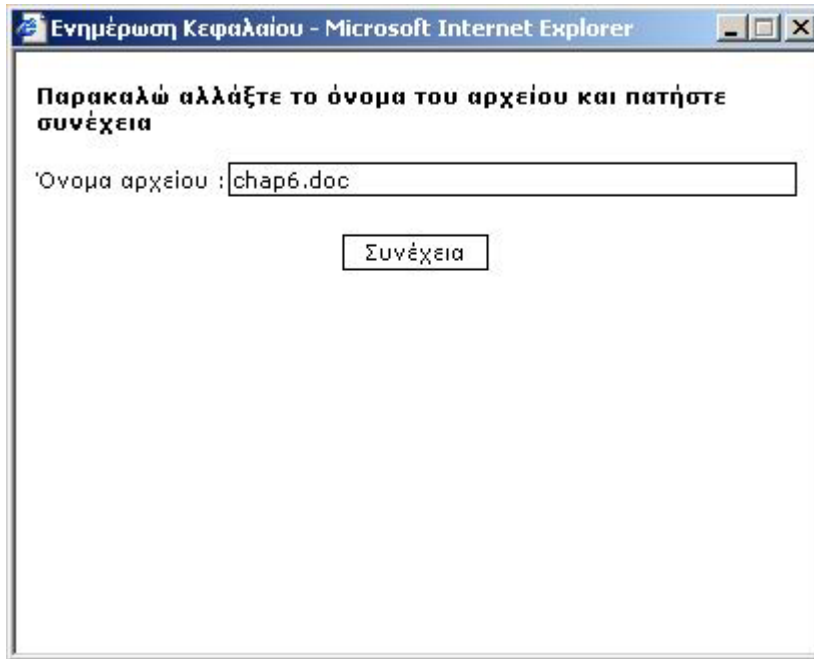
Εδώ ο χρήστης πρέπει να επιλέξει “Save” ή μια παραπλήσια επιλογή που θα του εμφανιστεί, για να το αποθηκεύσει στον τοπικό του δίσκο, όπου μετά θα μπορέσει να το επεξεργαστεί με το προαναφερθέν πρόγραμμα.



Σχήμα 5.3.3A

5.3.4 Μετονομασία

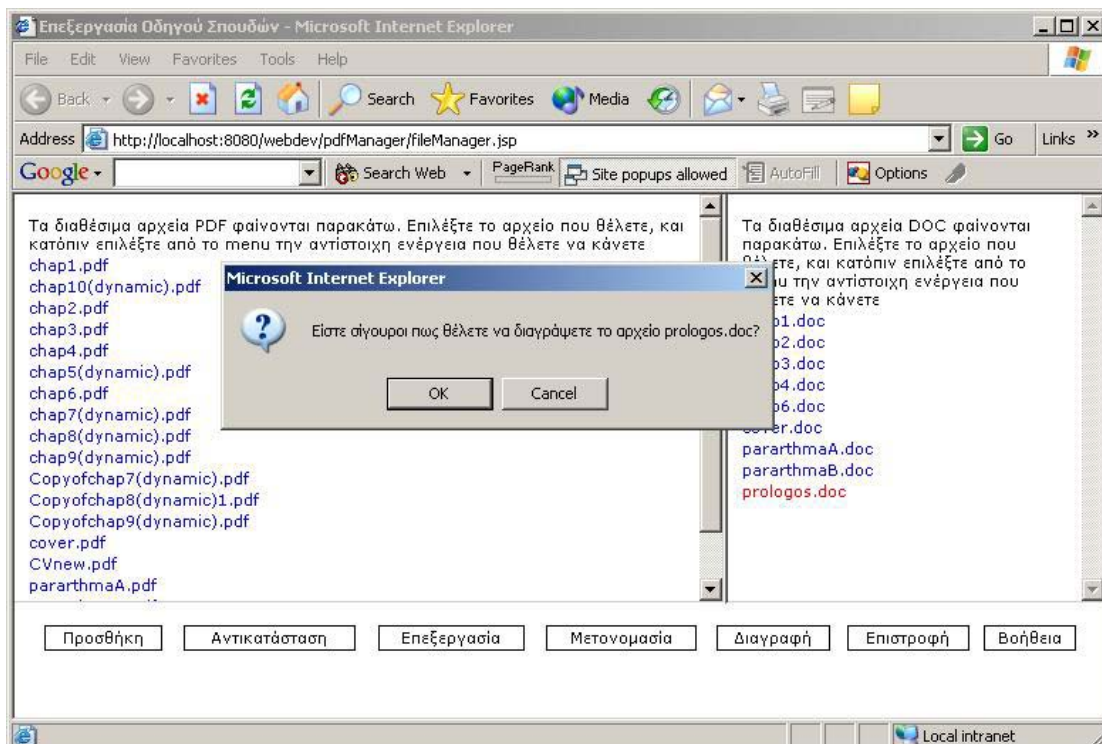
Εδώ ο χρήστης μπορεί να μετονομάσει ένα αρχείο που είναι αποθηκευμένο στον server. Επιλέγοντας το αρχείο που θέλει να μετονομάσει και κάνοντας κλικ στο “Μετονομασία” εμφανίζεται ένα παράθυρο παραπλήσιο με το παρακάτω σχήμα. Εκεί ο χρήστης δίνει το νέο όνομα του αρχείου και πατάει “Συνέχεια”.



Σχήμα 5.3.4.A

5.3.5 Διαγραφή

Με αυτή την επιλογή δίνεται η δυνατότητα στον χρήστη να διαγράψει ένα αρχείο που υπάρχει στον server. Ο χρήστης επιλέγει το προς διαγραφή αρχείο, κάνει κλικ στο κουμπί “Διαγραφή” και επιλέγει “OK” στο παράθυρο που του εμφανίζεται, όπως φαίνεται στο παρακάτω σχήμα.



Σχήμα 5.3.5.A

5.3.6 Επιστροφή

Με αυτή την επιλογή ο χρήστης κλείνει το παράθυρο αυτό και επιστρέφει στο αρχικό μενού της εφαρμογής, αφού το επαναεπιβεβαιώσει ο χρήστης (σε περίπτωση που το πάτησε κατά λάθος).

5.3.7 Βοήθεια

Εδώ εμφανίζεται ένα παράθυρο με κάποιες οδηγίες για τον χρήστη.

5.4 Σύνθεση Τελικού Οδηγού Σπουδών

Σε αυτό το μέρος της εφαρμογής γίνεται η τελική σύνθεση του οδηγού σε pdf από τα κομμάτια που δημιουργήθηκαν στα προηγούμενα στάδια. Όπως βλέπουμε στο σχήμα 5.4.A, αυτό το μέρος έχει διαιρεθεί σε 3 πλαίσια (frames). Το κάτω πλαίσιο, όπου εκεί είναι το μενού, το αριστερό πλαίσιο, όπου εκεί είναι τα περιεχόμενα του οδηγού σπουδών που προβάλλονται σε δενδρική μορφή, έτσι ακριβώς όπως θα αποθηκευτούν στο τελικό αρχείο, και τέλος δεξιά ο χρήστης μπορεί να έχει μια προεπισκόπηση σε pdf του επιλεγμένου από τα περιεχόμενα κεφαλαίου.

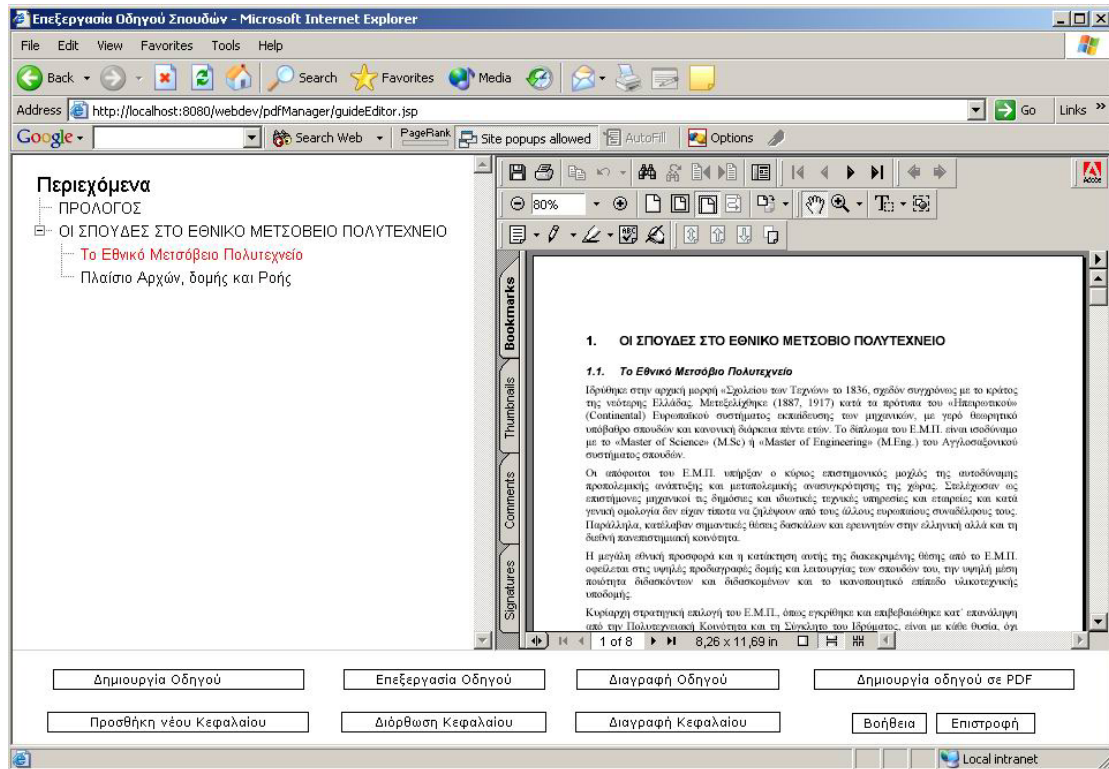
Γενικά, μέσω των λειτουργιών που παρέχονται σε αυτό το μέρος, ο χρήστης έχει την δυνατότητα να δημιουργήσει έναν οδηγό από την αρχή, και να αποθηκεύσει την “δομή” του στον server, τον τρόπο δηλαδή που ενώνονται τα επιμέρους κεφάλαια μεταξύ τους. Επομένως, εάν γίνουν αλλαγές στα περιεχόμενα των επιμέρους κεφαλαίων αλλά όχι στην σειρά με την οποία ενώνονται, για την δημιουργία του οδηγού σε αυτή τη φάση δεν χρειάζεται καμία αλλαγή.

Η δομή αυτή έπρεπε να χρησιμοποιηθεί, καθώς η επεξεργασία του οδηγού δεν μπορεί να γίνει απευθείας στο pdf. Το pdf είναι αρκετά κλειστό πρότυπο, χρησιμοποιείται κυρίως για την απεικόνιση πληροφορίας εφόσον αυτή έχει πάρει την τελική της μορφή, και όχι για την επεξεργασία αυτής.

Ο χρήστης μπορεί να διατηρεί πολλές διαφορετικές δομές του οδηγού αποθηκευμένες στον server, και να τις επεξεργάζεται μέσω της επιλογής “Επεξεργασία Οδηγού”.

Η φιλοσοφία που διέπει την χρήση αυτού του μέρους της εφαρμογής είναι η ίδια με τις προηγούμενες ενότητες : Ο χρήστης επιλέγει τον κόμβο στο δέντρο του πίνακα περιεχομένων, ο κόμβος αυτός κοκκινίζει, και η αντίστοιχη ενέργεια που επιλέγει από το μενού γίνεται στον κόμβο αυτό.

Παρακάτω εξηγούμε αναλυτικότερα τις διαθέσιμες επιλογές του μενού που ο χρήστης μπορεί να χρησιμοποιήσει.



Σχήμα 5.4A

5.4.1 Δημιουργία Οδηγού

Εδώ ο χρήστης μπορεί να δημιουργήσει τη δομή τους νέου οδηγού σπουδών. Εδώ πρέπει να τονίσουμε πως ο οδηγός δεν δημιουργείται σε pdf από αυτήν την επιλογή.

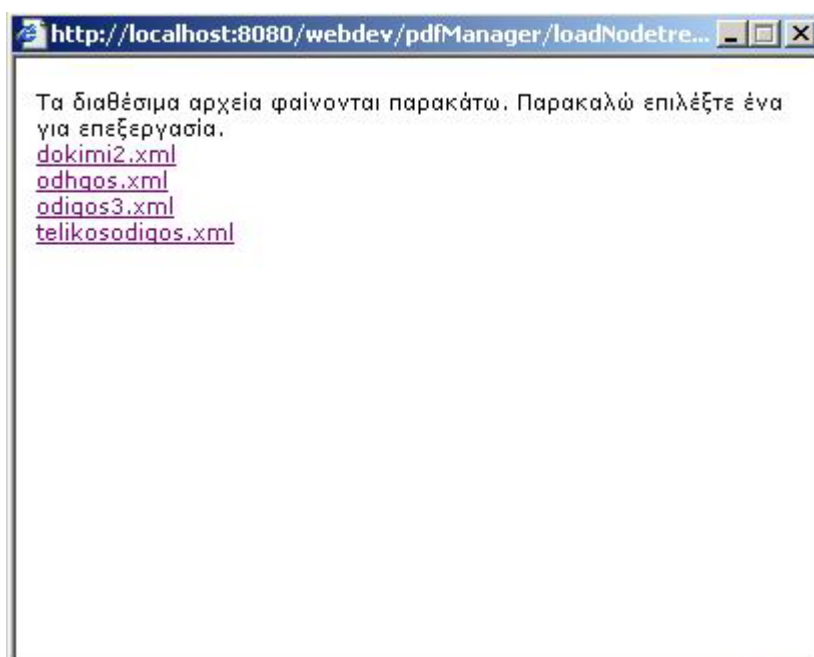
Ο χρήστης εδώ πρέπει να δώσει το όνομα του οδηγού που θέλει. Το όνομα πρέπει να αποτελείται από λατινικούς χαρακτήρες χωρίς κενά μεταξύ τους.



Σχήμα 5.4.1^A

5.4.2 Επεξεργασία Οδηγού

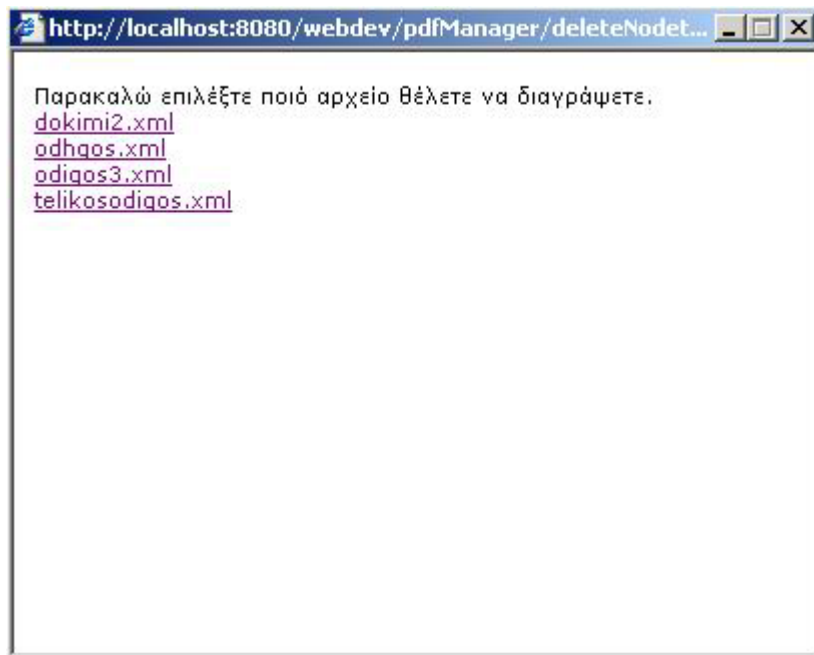
Ο χρήστης εδώ βλέπει την λίστα των διαθέσιμων δομών του οδηγού σπουδών που υπάρχουν στον server. Επιλέγοντας έναν οδηγό για επεξεργασία, τα περιεχόμενά του παρουσιάζονται στο αριστερό παράθυρο αυτός εφαρμογής και ο χρήστης μπορεί να κάνει όποιες αλλαγές αυτός θέλει.



Σχήμα 5.4.2^A

5.4.3 Διαγραφή Οδηγού

Αυτή η επιλογή δίνει στο χρήστη τη δυνατότητα να διαγράψει την δομή της οδηγού που είχε δημιουργήσει παλιότερα. Σημειωτέον ότι με την διαγραφή της δομής δεν διαγράφονται και τα αντίστοιχα pdf αρχεία που σχετίζονται με την δομή αυτή.



Σχήμα 5.4.3^A

5.4.4 Προσθήκη νέου Κεφαλαίου

Η προσθήκη μιας οποιαδήποτε υποενότητας του οδηγού (είτε κεφάλαιο είτε υποκεφάλαιο) γίνεται από αυτό το παράθυρο.

Εάν ο χρήστης θέλει να προσθέσει ένα κεφάλαιο στον οδηγό, πρέπει να κάνει κλικ στα "Περιεχόμενα" (με μαύρα έντονα γράμματα στην κορυφή του αριστερού μέρους). Έτσι θα προστεθεί κεφάλαιο στο τέλος του οδηγού.

Εάν τώρα θέλει να προσθέσει ένα υποκεφάλαιο, θα πρέπει να επιλέξει το αντίστοιχο κεφάλαιο στο οποίο θέλει να προσθέσει το υποκεφάλαιο αυτό (τότε το κεφάλαιο αυτό γίνεται κόκκινο) και κατόπιν να κάνει κλικ στο "Προσθήκη κεφαλαίου".

Εάν πχ θέλει να προσθέσει το κεφάλαιο 3, θα κάνει κλικ στα "Περιεχόμενα" και κατόπιν κλικ στο "Προσθήκη κεφαλαίου".

Εάν θέλει να προσθέσει την υποενότητα 1.1 του κεφαλαίου 1, θα πρέπει να επιλέξει το κεφάλαιο 1, και κατόπιν να επιλέξει "Προσθήκη κεφαλαίου".

Στο παράθυρο που παρουσιάζεται τώρα για την προσθήκη νέου κεφαλαίου (ή υποκεφαλαίου, δεν παίζει ουσιαστικό ρόλο) ο χρήστης πρέπει να δώσει το όνομα αυτού, το όνομα δηλαδή με το οποίο θα παρουσιάζεται στα περιεχόμενα του τελικού οδηγού, και στο δέντρο που θα δημιουργηθεί στο τέλος.

Μετά ο χρήστης έχει 2 επιλογές :

A) Να προσθέσει ένα αρχείο pdf, το οποίο το επιλέγει από την λίστα.

B) Να προσθέσει ένα σελιδοδείκτη. Ο σελιδοδείκτης είναι ένας αριθμός που δείχνει σε μια συγκεκριμένη σελίδα ενός pdf. Το pdf κάθε φορά θα είναι το pdf του “πατέρα” της συγκεκριμένης ενότητας, δηλαδή της ενότητας ενός επιπέδου πιο πάνω.

Πιο συγκεκριμένα, έστω ότι ο χρήστης θέλει να τοποθετήσει έναν σελιδοδείκτη που δείχνει στην ενότητα 1.1 του κεφαλαίου 1 που βρίσκεται στην σελίδα 3 (του κεφαλαίου 1 και όχι του συνολικού οδηγού). Έστω επίσης ότι το κεφάλαιο 1 είναι το chap1.pdf. Ο σελιδοδείκτης θα δείχνει στην σελίδα 3 του chap1.pdf.

Σε παρακάτω κεφάλαιο θα παρουσιάσουμε αναλυτικά παραδείγματα που θα περιγράψουν διεξοδικά τέτοιου είδους θέματα.

Σε περίπτωση που ο χρήστης δεν δώσει όνομα κεφαλαίου, ή συμπληρώσει και τα 2 παρακάτω στοιχεία (και όνομα αρχείου και σελιδοδείκτη) η εφαρμογή δεν επιτρέπει την καταχώρηση αυτή, διότι δεν έχει νόημα.

Προσθήκη νέου Κεφαλαίου - Microsoft Internet Explorer

Δώστε το όνομα του κεφαλαίου που θέλετε

Όνομα κεφαλαίου :

Παρακαλώ επιλέξτε το αρχείο από την παρακάτω λίστα και πατήστε Συνέχεια.

Όνομα αρχείου :

Εάν θέλετε να βάλετε έναν σελιδοδείκτη σε ένα ήδη υπάρχον κεφάλαιο, παρακαλώ δώστε τον αριθμό της σελίδας και πατήστε Συνέχεια. Ο αριθμός της σελίδας που θα δώσετε πρέπει να είναι μεγαλύτερος ή ίσος από τον προηγούμενο σελιδοδείκτη.

Αριθμός σελίδας :

Done Local intranet

Σχήμα 5.4.4A

5.4.5 Διόρθωση Κεφαλαίου

Στην περίπτωση που ο χρήστης επιλέξει διόρθωση του επιλεγμένου κεφαλαίου, παρουσιάζονται 2 διαφορετικά παράθυρα, ανάλογα με το είδος του κεφαλαίου που έχει επιλέξει.

Εάν έχει επιλέξει κεφάλαιο που αντιστοιχεί σε αρχείο pdf εμφανίζεται το παράθυρο του σχήματος 5.4.5A. Εδώ ο χρήστης μπορεί να διορθώσει τον τίτλο που έχει δώσει στο κεφάλαιο αυτό, να αλλάξει το αρχείο pdf με το οποίο συνδέει το κεφάλαιο αυτό, και τέλος να επεξεργαστεί το αρχείο κειμένου από το οποίο προέκυψε το συγκεκριμένο pdf. Η επιλογή αυτή όπως είναι φυσικό εμφανίζεται μόνο εάν το pdf προέκυψε από doc, και όχι εάν δημιουργήθηκε αυτόματα από δεδομένα της βάσης της σχολής (όπως τα κεφάλαια 5,8κλπ). Η επεξεργασία γίνεται ακριβώς με τον ίδιο τρόπο που περιγράφεται στην παράγραφο 5.3.3.

Εάν έχει επιλέξει κεφάλαιο που αντιστοιχεί σε σελιδοδείκτη, εμφανίζεται το παράθυρο του σχήματος 5.4.5B. Εδώ ο χρήστης μπορεί να αλλάξει το τίτλο της ενότητας, και τον αριθμό της σελίδας που δείχνει αυτός ο σελιδοδείκτης. Παρέχεται επίσης η δυνατότητα στον χρήστη να επεξεργαστεί το αρχείο κειμένου από το οποίο δημιουργήθηκε το pdf.

Ενημέρωση Κεφαλαίου - Microsoft Internet Explorer

Δώστε το όνομα του κεφαλαίου που θέλετε

Όνομα κεφαλαίου :

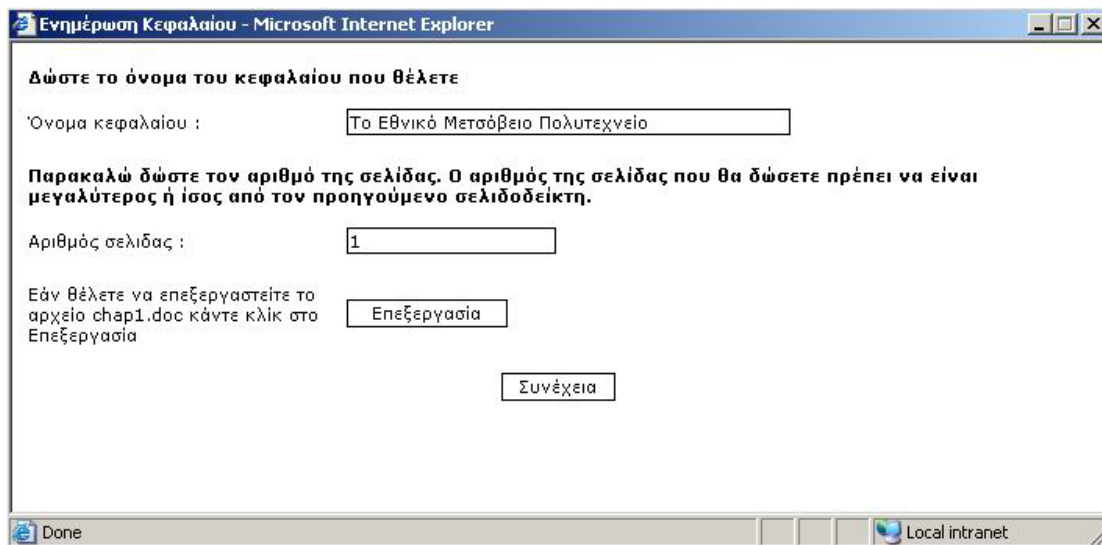
Παρακαλώ επιλέξτε το αρχείο PDF από την παρακάτω λίστα και πατήστε Συνέχεια.

Αρχεία ήδη ανεβασμένα :

Εάν θέλετε να επεξεργαστείτε το αρχείο chap1.doc κάντε κλικ στο

Done Local intranet

Σχήμα 5.4.5A

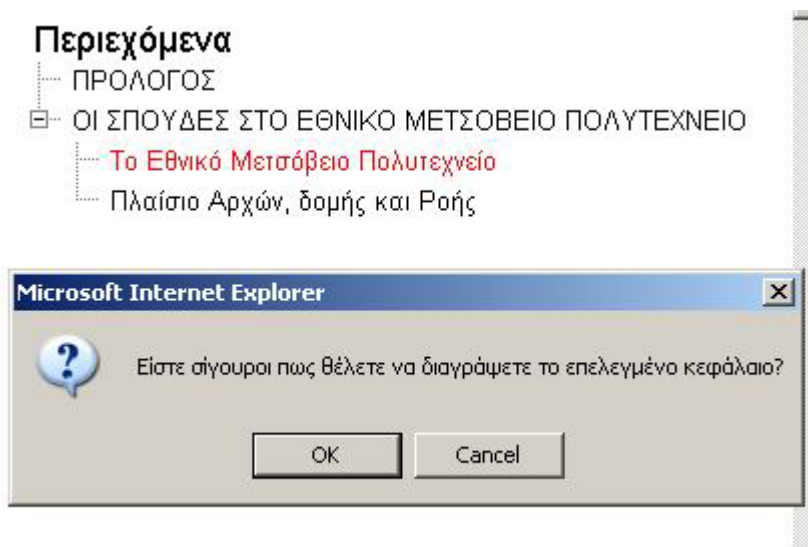


Σχήμα 5.4.5B

5.4.6 Διαγραφή Κεφαλαίου

Με αυτή την επιλογή ο χρήστης μπορεί να διαγράψει το επιλεγμένο κεφάλαιο ή υποκεφάλαιο από την δομή του οδηγού σπουδών. Με την διαγραφή δεν διαγράφεται και το αντίστοιχο αρχείο pdf από τον server, διαγράφεται μόνο η σύνδεσή του στην συγκεκριμένη δομή. Για να διαγράψει το αρχείο αυτό ο χρήστης θα πρέπει από το αρχικό μενού της εφαρμογής να επιλέξει “Διαχείριση στατικού περιεχομένου” και να το διαγράψει από εκεί με την διαδικασία που περιγράφεται στην παράγραφο 5.3.5

Μόλις ο χρήστης απαντήσει θετικά στην ερώτηση που του τίθεται, το συγκεκριμένο κεφάλαιο μαζί με όλες τις υποενότητες του διαγράφονται.



Σχήμα 5.4.6A

5.4.7 Δημιουργία οδηγού σε PDF

Αφού ο χρήστης δημιουργήσει την επιθυμητή δομή που θέλει να έχει ο οδηγός σπουδών, με αυτή την επιλογή δημιουργεί τον τελικό οδηγό στην μορφή pdf, αφού δώσει το όνομα που θέλει (λατινικοί χαρακτήρες, χωρίς κενά).

Στις παρακάτω παραγράφους περιγράφουμε αναλυτικά τις επιλογές που έχει ο χρήστης για το είδος του παραγόμενου pdf αρχείου.

Εξαγωγή PDF - Microsoft Internet Explorer

Σχόλια στην κεφαλίδα - υποσέλιδο και αρίθμηση σελίδων

Σχόλια κεφαλίδας Αρίθμηση σελίδας στην κεφαλίδα

Σχόλια υποσέλιδου Αρίθμηση σελίδας στο υποσέλιδο

Επιλεγμένη γλώσσα περιεχομένων Δημιουργία περιεχομένων

Προσθήκη εξώφυλλου

Εμφάνιση σελίδων

Μία σελίδα Μία στήλη Δύο στήλες, σελίδες αριστερά Δύο στήλες, σελίδες δεξιά

Τρόπος προβολής εγγράφου

Κανένας Προβολή χάρτη εγγράφου Προβολή εικόνων thumbnail Προβολή πλήρους οθόνης

Απόκρυψη εργαλειοθήκης Απόκρυψη μενού

Κωδικοποίηση

Επιτρεπτή αντιγραφή Επιτρεπτή αλλαγή περιεχομένων Επιτρεπτή εκτύπωση Επιτρεπτή αλλαγή σχολίων

Επιτρεπτή αντικατάσταση Επιτρεπτός αναγνώστης οθόνης Επιτρεπτή αλλαγή αρχείου Προσθήκη κωδικών

Κωδικός χρήστη Κωδικός κατόχου

Περίληψη

Τίτλος Θέμα Συγγραφέας

Δημιουργός Λέξεις κλειδιά

Δώστε όνομα αρχείου PDF για δημιουργία :

Done Local intranet

Σχήμα 5.4.7A

5.4.7.1 Σχόλια

Εδώ ο χρήστης επιλέγει εάν το τελικό αρχείο pdf θα έχει σχόλια στην κεφαλίδα και στο υποσέλιδο, και εάν θα αριθμούνται οι σελίδες αντίστοιχα στην κεφαλίδα ή στο υποσέλιδο.

Επίσης, επιλέγει εάν η εφαρμογή μαζί με τον οδηγό θα δημιουργήσει (αυτόματα) τον πίνακα περιεχομένων, τη γλώσσα αυτών, και εάν θα τοποθετηθεί και εξώφυλλο. Το εξώφυλλο τοποθετείται πριν από τον πίνακα περιεχομένων στην αρχή του pdf εγγράφου.

5.4.7.2 *Εμφάνιση σελίδων*

Εδώ επιλέγουμε πως θα φαίνονται οι σελίδες του pdf όταν το ανοίγουμε.

5.4.7.3 *Τρόπος Προβολής εγγράφου*

Εδώ επιλέγουμε ποια στοιχεία από τα στοιχεία του Acrobat θα εμφανίζονται στην αριστερή πλευρά του κειμένου.

5.4.7.4 *Κωδικοποίηση*

Σε αυτό το μέρος μπορούμε να θέσουμε κάποιους περιορισμούς στους χρήστες που θα διαβάσουν το αρχείο, εάν θέλουμε να το προστατέψουμε από αντιγραφή ή αλλοίωση του περιεχομένου του. Μπορούμε να απαγορέψουμε κάποιες ενέργειες στον χρήστη όπως πχ αντιγραφή στο clipboard, να θέσουμε κωδικούς, και άλλα.

5.4.7.5 *Περίληψη*

Εδώ δίνουμε κάποια ενημερωτικά στοιχεία για το αρχείο όπως τίτλο, θέμα, συγγραφέα, δημιουργό χωρίς να παίζουν κάποιο ιδιαίτερο λόγο για την δημιουργία του.

5.4.7.6 *Όνομα αρχείου*

Εδώ πρέπει να δώσουμε το όνομα του αρχείου pdf που θέλουμε να δημιουργήσουμε. Το όνομα πρέπει να αποτελείται από λατινικούς χαρακτήρες, χωρίς κενά (πχ OdhgosSproudon.pdf και όχι Οδηγός Σπουδών.pdf). Μετά την δημιουργία του, δίνεται η δυνατότητα στον χρήστη να δει το αρχείο που δημιούργησε. Εάν θέλει να το δει σε κάποια άλλη στιγμή, είναι διαθέσιμο από το βασικό μενού της εφαρμογής, στην επιλογή Προβολή “Διαθέσιμων οδηγών Σπουδών”.

5.4.8 Επιστροφή

Η επιλογή αυτή επιστρέφει τον χρήστη στο αρχικό μενού της εφαρμογής, αφού απαντήσει θετικά για επιβεβαίωση της ενέργειας αυτής (σε περίπτωση που την επιλέξει κατά λάθος).

5.4.9 Βοήθεια

Εδώ παρουσιάζονται κάποιες γενικές πληροφορίες για την σωστή χρήση της εφαρμογής.

5.5 Προβολή διαθέσιμων Οδηγών Σπουδών

Σε αυτή την σελίδα ο χρήστης μπορεί να επεξεργαστεί τους διαθέσιμους οδηγούς σπουδών σε pdf που έχουν δημιουργηθεί μέχρι τώρα και είναι αποθηκευμένοι στον server.

Η σελίδα είναι χωρισμένη σε 2 μέρη, στο πάνω μέρος παρουσιάζονται οι οδηγοί, και στο κάτω μέρος υπάρχει το μενού με τις διαθέσιμες επιλογές του χρήστη. Η λειτουργικότητά της είναι ίδια με αυτή των άλλων σελίδων : Ο χρήστης επιλέγει το αρχείο που θέλει και κατόπιν την ενέργεια που θέλει να του κάνει, από το μενού.

Αυτή η σελίδα θα μπορούσε να χρησιμοποιηθεί για να διατηρούνται οι οδηγοί σπουδών που δημιουργούνται κάθε χρόνο στη Σχολή, για λόγους αρχειοθέτησης (archiving).

Οι διαθέσιμες επιλογές του χρήστη περιγράφονται αναλυτικότερα παρακάτω.



Σχήμα 5.5A

5.5.1 Προβολή

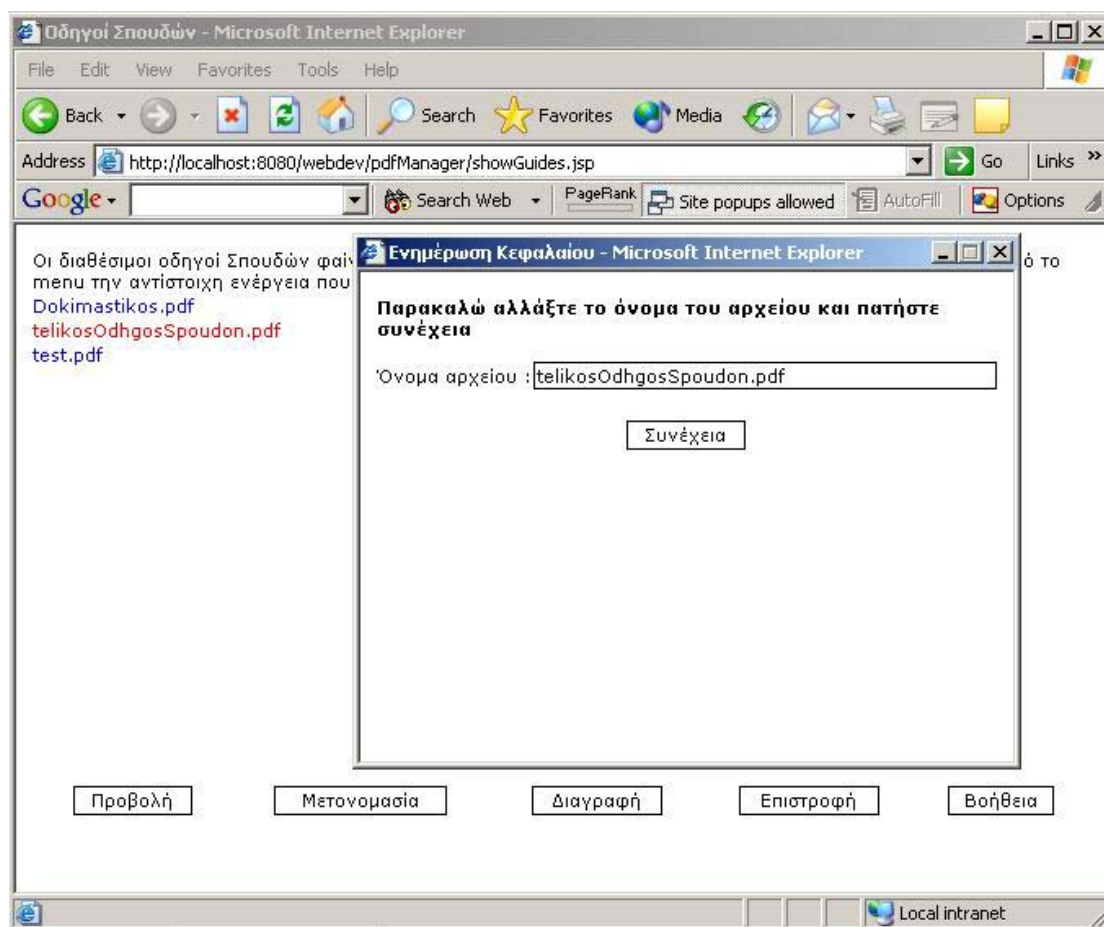
Εφόσον ο χρήστης επιλέξει έναν συγκεκριμένο οδηγό και πατήσει "Προβολή" ανοίγει ένα παράθυρο με τον συγκεκριμένο οδηγό σε pdf. Για να λειτουργήσει η συγκεκριμένη δυνατότητα, θα πρέπει στον υπολογιστή του χρήστη να είναι

εγκαταστημένη κάποια έκδοση του Adobe Acrobat Reader, ή κάποιο άλλο πρόγραμμα ικανό να προβάλει αρχεία pdf.

Επίσης, λόγω του σχετικά μεγάλου όγκου του οδηγού σπουδών, εάν ο χρήστης επιλέξει αυτή την λειτουργία και είναι συνδεδεμένος στο διαδίκτυο με αρκετά αργή σύνδεση (π.χ. από το σπίτι με dial up) θα παρατηρήσει μια καθυστέρηση της τάξεως 4-5 λεπτών. Για αυτό συνιστάται ο χρήστης να χρησιμοποιεί την εφαρμογή εφόσον είναι συνδεδεμένος διαδίκτυο μέσω του τοπικού δικτύου του πολυτεχνείου.

5.5.2 Μετονομασία

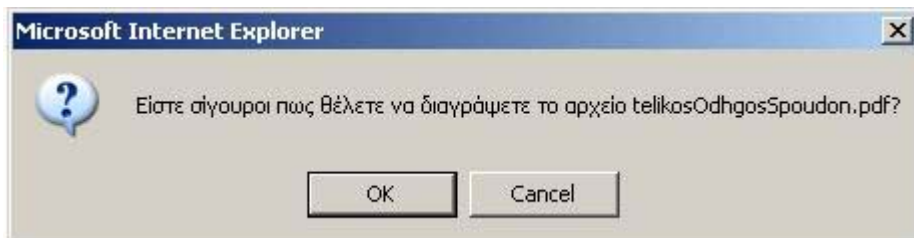
Εδώ δίνεται η δυνατότητα στον χρήστη να μετονομάσει ένα αρχείο που έχει ήδη δημιουργηθεί. Ο χρήστης θα πρέπει να έχει κατά νου τον περιορισμό που θέτει η εφαρμογή για τα ονόματα αρχείων : Πρέπει να αποτελούνται από λατινικούς χαρακτήρες, χωρίς κενά.



Σχήμα 5.5.2Α

5.5.3 Διαγραφή

Εδώ ο χρήστης μπορεί να διαγράψει ένα αρχείο που έχει επιλέξει μετά από επιβεβαίωση της ενέργειάς του (προς αποφυγή διαγραφής αρχείου κατά λάθος).



Σχήμα 5..5.3A

5.5.4 Επιστροφή

Εδώ ο χρήστης πάλι μετά από επιβεβαίωση της ενέργειάς του κλείνει το συγκεκριμένο παράθυρο και επιστρέφει στο κεντρικό μενού της εφαρμογής.

5.5.5 Βοήθεια

Εδώ παρουσιάζονται κάποιες γενικές πληροφορίες για την σωστή χρήση της εφαρμογής.

6. Σενάρια πρακτικών εφαρμογών

Σε αυτή την ενότητα θα παρουσιάσουμε βήμα - βήμα κάποιες λειτουργίες της εφαρμογής που ο χρήστης πρόκειται να χρησιμοποιήσει κατά κόρον.

Σε όλες τις περιπτώσεις θα θεωρήσουμε πως ο οδηγός που θα χρησιμοποιήσουμε είναι ο `telikosodigos.xml`. Για να τον επεξεργαστούμε ακολουθούμε την διαδικασία που περιγράφεται στην παράγραφο 5.4.4

6.1 Ενημέρωση οδηγού για αλλαγές σε στατικό περιεχόμενο

Σε αυτή την περίπτωση θα παρουσιαστεί βήμα – βήμα πως ο χρήστης μπορεί να αλλάξει ένα στατικό αρχείο που υπάρχει στον server, να δημιουργήσει το αντίστοιχο pdf, και να δημιουργήσει τον τελικό οδηγό.

Για να μπορέσει να το επιτύχει αυτό ο χρήστης, χρειάζεται να είναι εγκατεστημένες στον υπολογιστή του και 2 ανεξάρτητες εφαρμογές :

A) Η εφαρμογή Microsoft Word, απαραίτητη για την επεξεργασία του αρχείου κειμένου που θα κατεβάσει από τον Server.

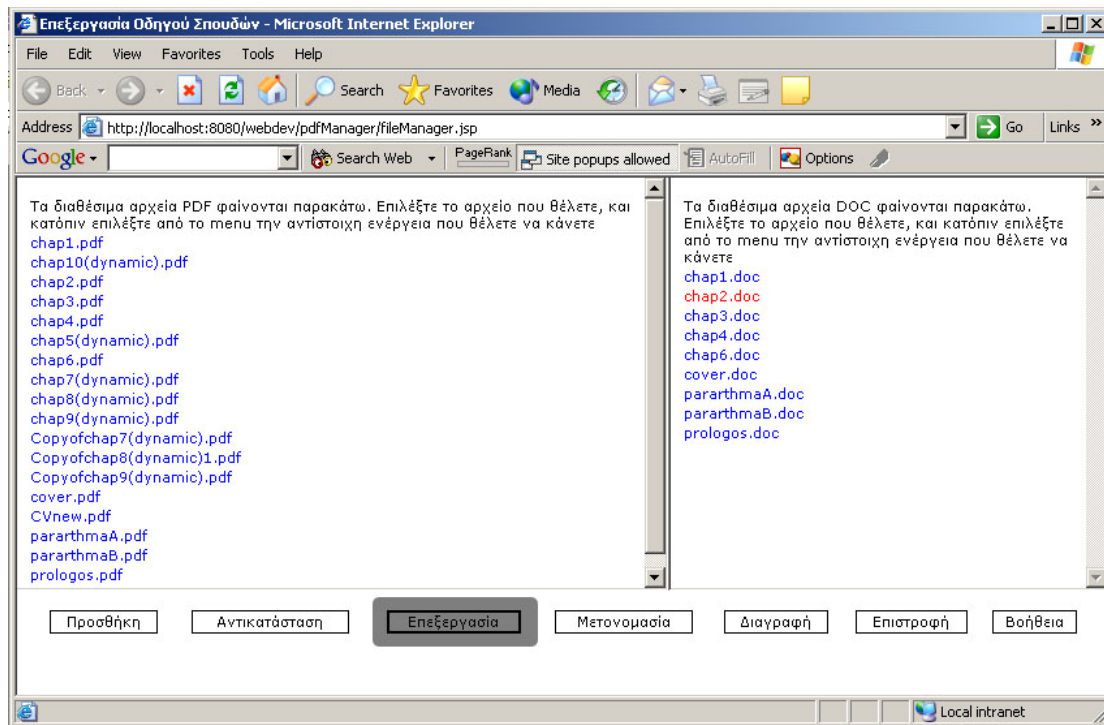
B) Μια εφαρμογή που θα μετατρέψει το doc αρχείο σε pdf. Τέτοιες εφαρμογές υπάρχουν πολλές στο εμπόριο, στο συγκεκριμένο παράδειγμα όμως θα χρησιμοποιήσουμε την εφαρμογή Adobe Acrobat 5, η οποία περιέχεται και στο CD της διπλωματικής.

Με την εγκατάσταση αυτή εγκαθίσταται στο σύστημα και ένας “εικονικός” εκτυπωτής, ο Acrobat Distiller. Έτσι, επιλέγοντας να εκτυπώσουμε το αρχείο doc σε αυτόν τον εκτυπωτή, εμφανίζεται ένα μενού μέσω του οποίου δημιουργούμε το αντίστοιχο pdf.

Ας υποθέσουμε τώρα πως ο χρήστης θέλει να αλλάξει το κείμενο του κεφαλαίου 2 που υπάρχει στον server. Ο χρήστης θα κάνει τις αντίστοιχες ενέργειες :

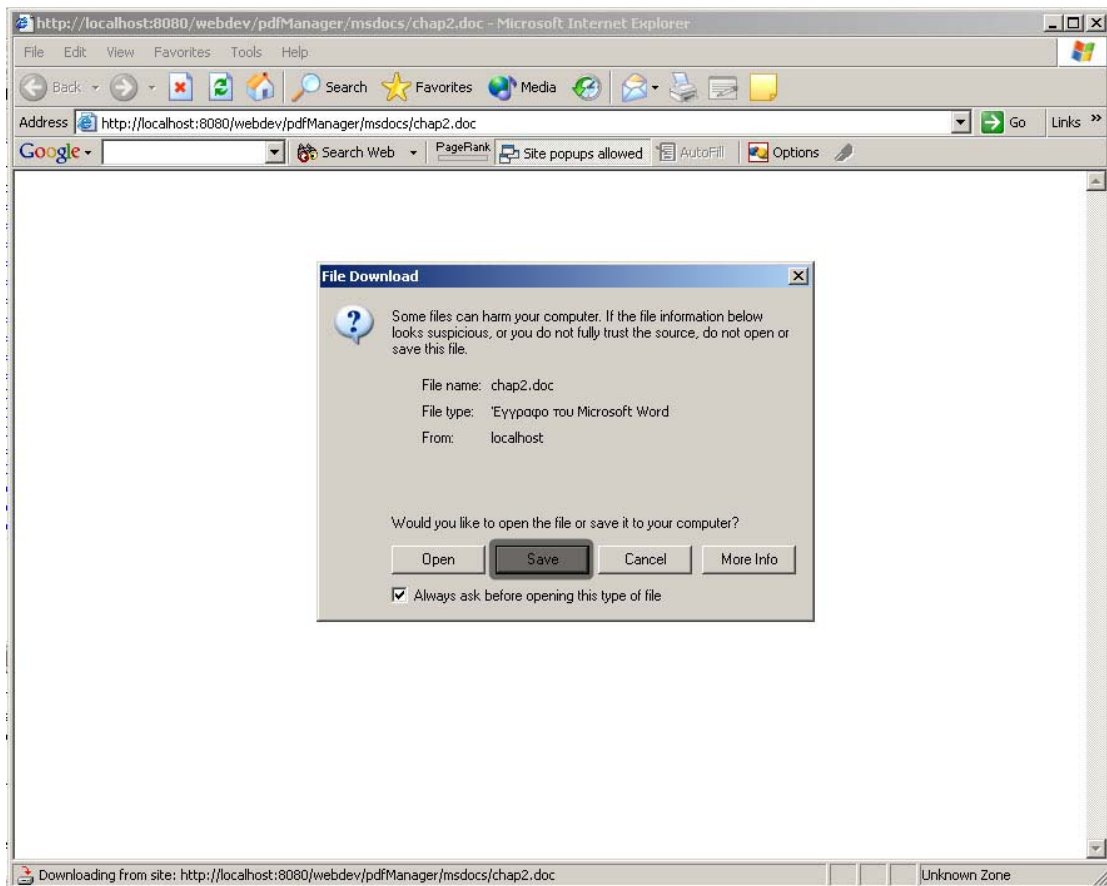
A) Επιλογή “Διαχείριση Στατικού περιεχομένου” από το αρχικό μενού.

B) Στο παράθυρο που θα εμφανιστεί, ο χρήστης επιλέγει το αρχείο `char2.doc` και κατόπιν επιλέγει Επεξεργασία, όπως φαίνεται στο παρακάτω σχήμα.



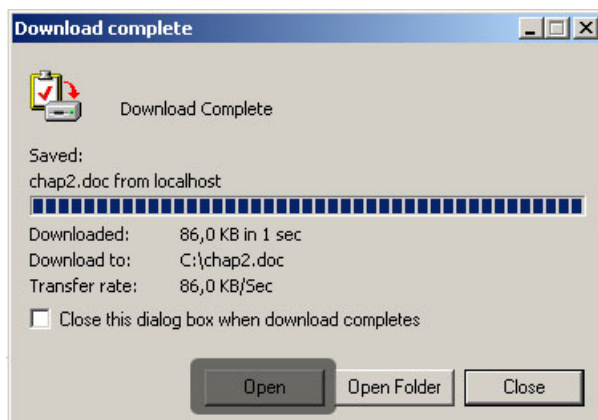
Σχήμα 6.1A

Γ) Κατόπιν, ανάλογα με τις ρυθμίσεις του υπολογιστή του, θα του εμφανιστεί ένα παραπλήσιο μήνυμα με αυτό που υπάρχει στο παρακάτω σχήμα. Εκεί ο χρήστης επιλέγει “Save” και αποθηκεύει το αρχείο τοπικά στον υπολογιστή του.



Σχήμα 6.1B

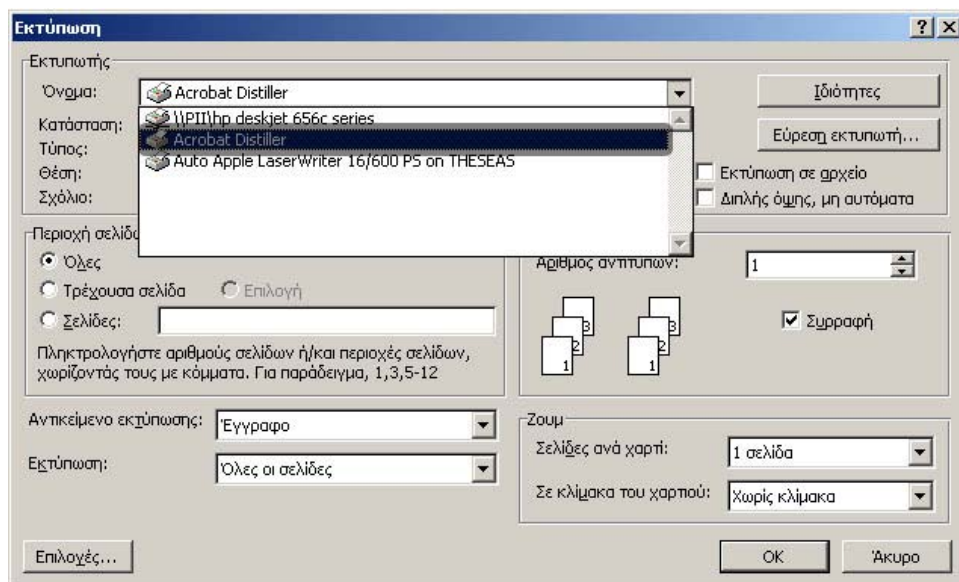
Δ) Μόλις το αρχείο αποθηκευτεί, ένα μήνυμα σαν αυτό του σχήματος 6.1Γ θα εμφανιστεί. Ο χρήστης εδώ επιλέγει “Open” και το παράθυρο της εφαρμογής MS Word ανοίγει, και έτσι ο χρήστης μπορεί να κάνει τις επιθυμητές αλλαγές.



Σχήμα 6.1Γ

Ε) Μόλις ο χρήστης τελειώσει με τις αλλαγές, επιλέγει αποθήκευση του κειμένου. Κατόπιν, θα πρέπει να δημιουργήσει το αντίστοιχο pdf. Σύμφωνα με αυτά

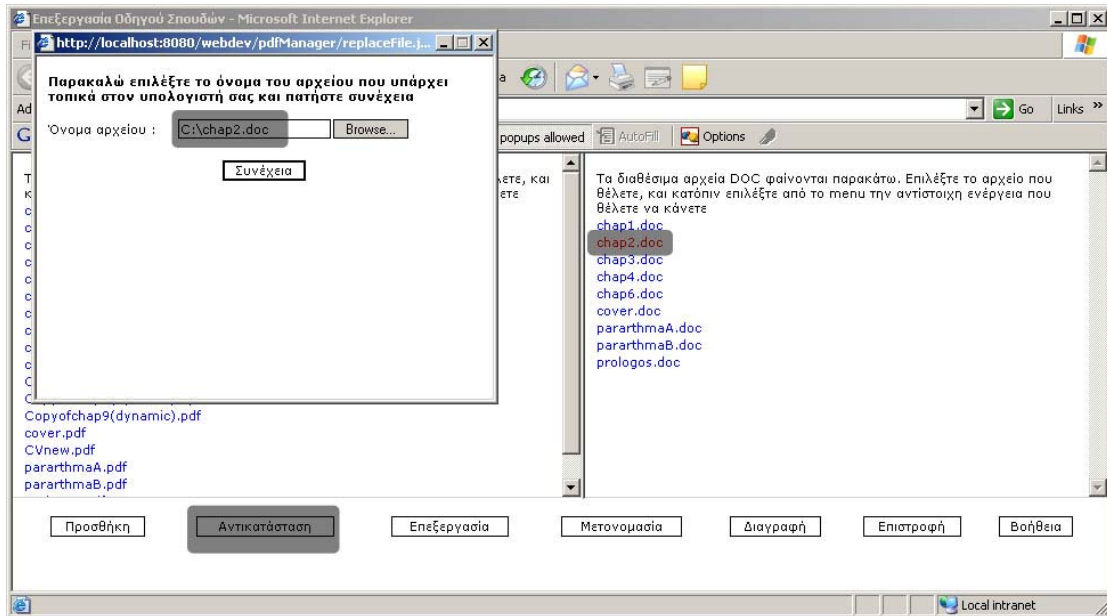
που είπαμε παραπάνω, εδώ ο χρήστης από την εφαρμογή MS Word επιλέγει Αρχείο - > Εκτύπωση, και στο παράθυρο που θα του εμφανιστεί επιλέγει για εκτυπωτή τον Acrobat Distiller. Κατόπιν ακολουθεί τις οδηγίες του προγράμματος και παράγει το αρχείο pdf.



Σχήμα 6.1Δ

ΣΤ) Τώρα ο χρήστης έχει δημιουργήσει και τα 2 απαραίτητα αρχεία (doc και pdf) αλλά βρίσκονται τοπικά στον υπολογιστή του. Πρέπει τώρα να αντικαταστήσει τις παλιές εκδόσεις που υπάρχουν στον server με τις καινούριες δικές του.

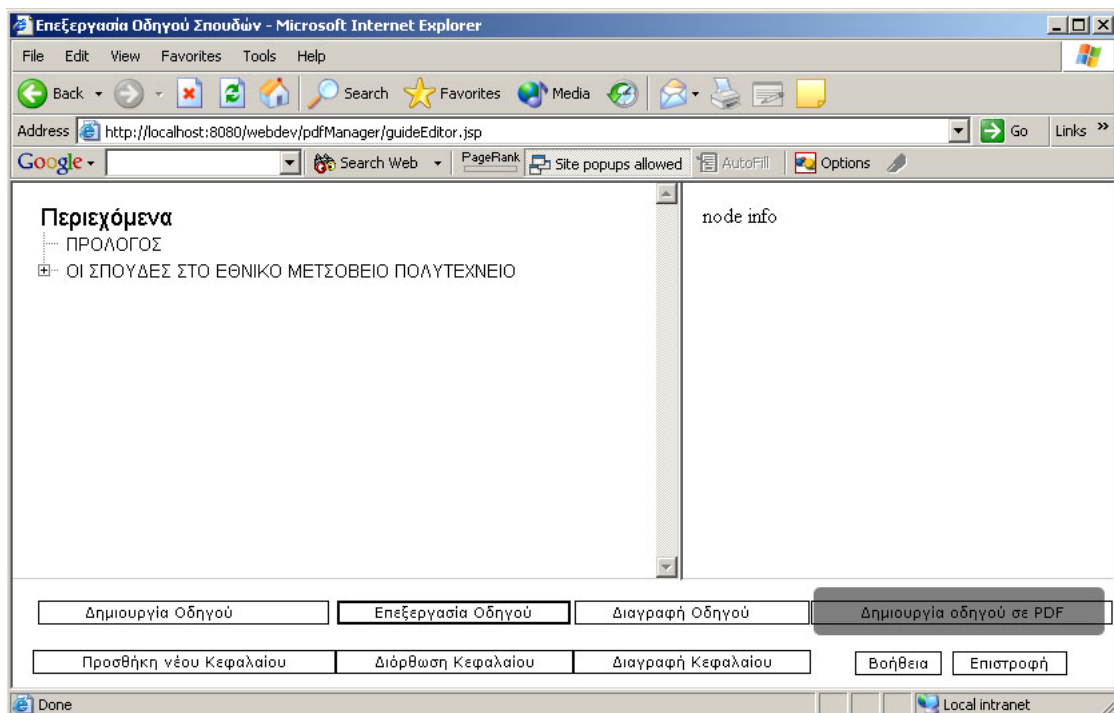
Ζ) Τώρα ο χρήστης πρέπει να επιστρέψει στην εφαρμογή στην Διαχείριση Στατικού περιεχομένου. Εκεί, όπως φαίνεται στο σχήμα 6.1Ε ο χρήστης επιλέγει το αρχείο που θέλει να αντικαταστήσει (το char2.doc) και κατόπιν επιλέγει αντικατάσταση. Στο παράθυρο που εμφανίζεται, ο χρήστης υποδεικνύει στο σύστημα την θέση του αρχείου τοπικά στον υπολογιστή του και κατόπιν κάνει κλικ στο Συνέχεια.



Σχήμα 6.1Ε

Η) Το ίδιο με το βήμα Ζ πρέπει να κάνει ο χρήστης και για το chap2.pdf που υπάρχει στον υπολογιστή του.

Θ) Αφού τελειώσει με τις αλλαγές ο χρήστης επιλέγει από το αρχικό μενού Σύνοψη τελικού Οδηγού σπουδών, επιλέγει τον κατάλληλο οδηγό με το κουμπί Επεξεργασία Οδηγού, και κατόπιν επιλέγει Δημιουργία οδηγού σε PDF όπως φαίνεται στο παρακάτω σχήμα.



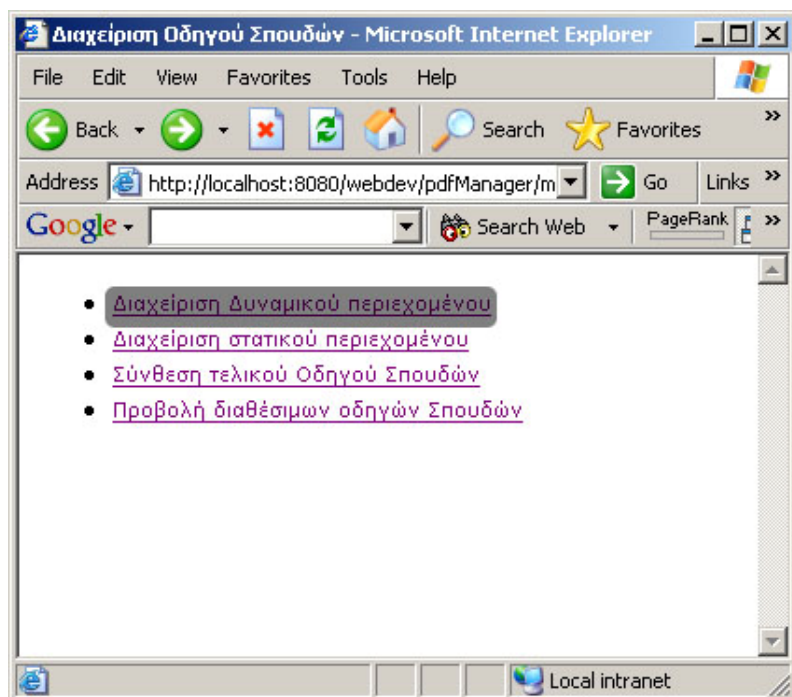
Σχήμα 6.1ΣΤ

6.2 Ενημέρωση οδηγού για αλλαγές σε δυναμικό περιεχόμενο

Σε αυτή τη περίπτωση θα περιγράψουμε τα βήματα που πρέπει να ακολουθήσει ο χρήστης για να ενημερώσει τον οδηγό Σπουδών σε περίπτωση που αλλάξουν κάποια δυναμικά δεδομένα που υπάρχουν στην βάση της σχολής (πχ προστέθηκε κάποιο νέο μάθημα Κορμού, και έτσι πρέπει να αλλάξει το κεφάλαιο 7).

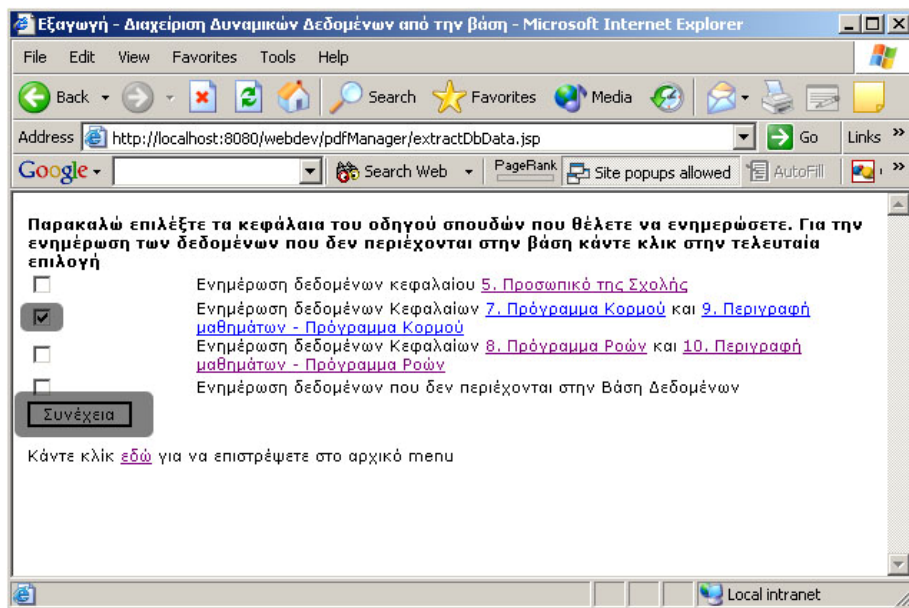
Ο χρήστης θα ακολουθήσει τα παρακάτω βήματα :

A) Από το αρχικό μενού της εφαρμογής ο χρήστης επιλέγει Διαχείριση Δυναμικού Περιεχομένου, όπως φαίνεται στο παρακάτω σχήμα.



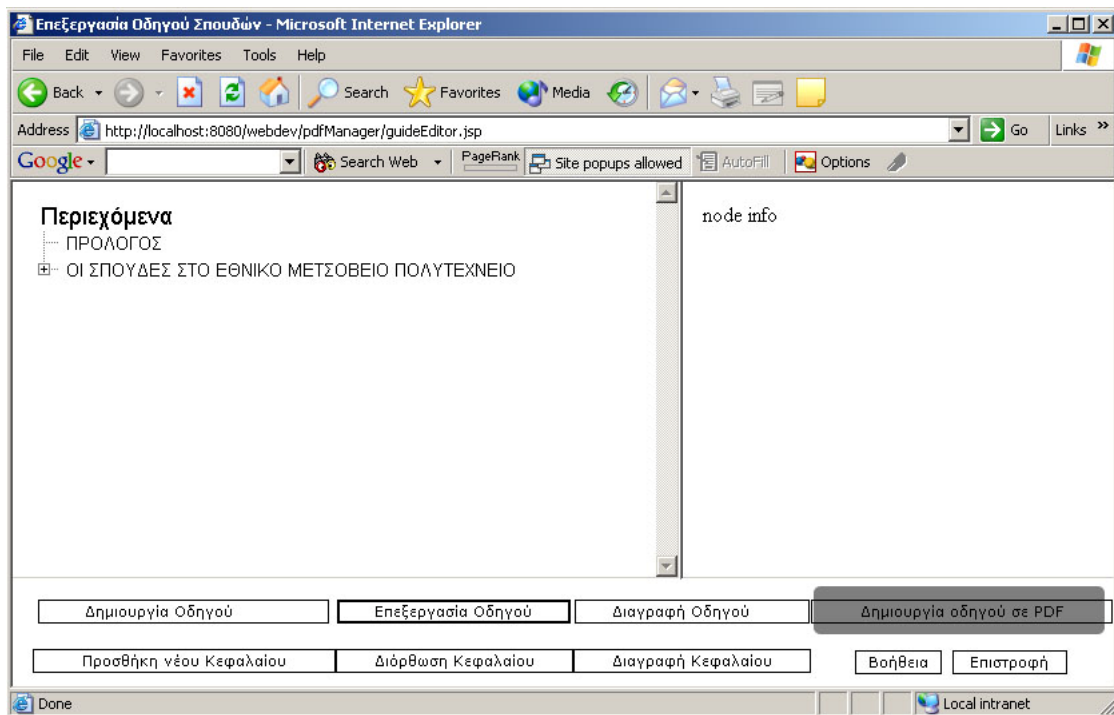
Σχήμα 6.2A

B) Κατόπιν, επιλέγει τη δεύτερη επιλογή στην επόμενη σελίδα, και κάνει κλικ στο Συνέχεια. Η εφαρμογή συνδέεται στην βάση δεδομένων της σχολής, διαβάζει τα ανανεωμένα στοιχεία και κατόπιν συνθέτει το συγκεκριμένο pdf από την αρχή, ανανεώνοντάς το.



Σχήμα 6.2B

Γ) Τέλος, ο χρήστης Επιλέγει από το αρχικό μενού “Σύνθεση τελικού οδηγού Σπουδών”, επιλέγει τον συγκεκριμένο οδηγό από το μενού “Επεξεργασία οδηγού” και κατόπιν επιλέγει “Δημιουργία Οδηγού σε PDF”, όπως βλέπουμε στο παρακάτω σχήμα.

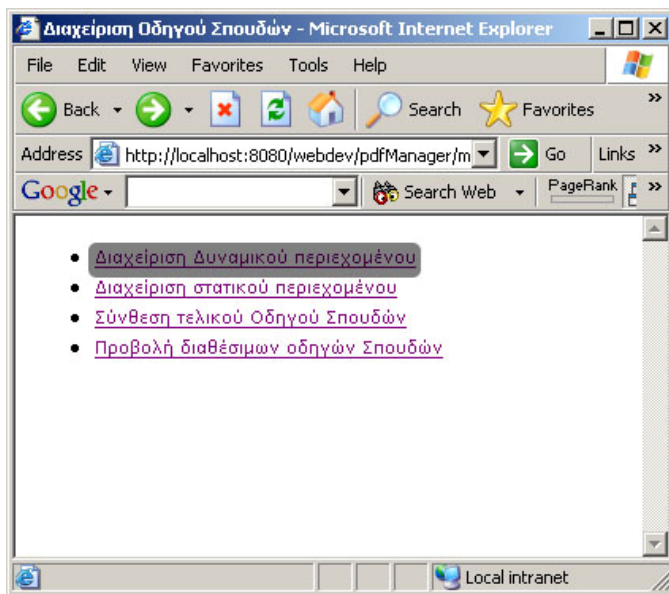


Σχήμα 6.2Γ

6.3 Ενημέρωση οδηγού σε δυναμικό περιεχόμενο που δεν περιέχεται στην βάση δεδομένων

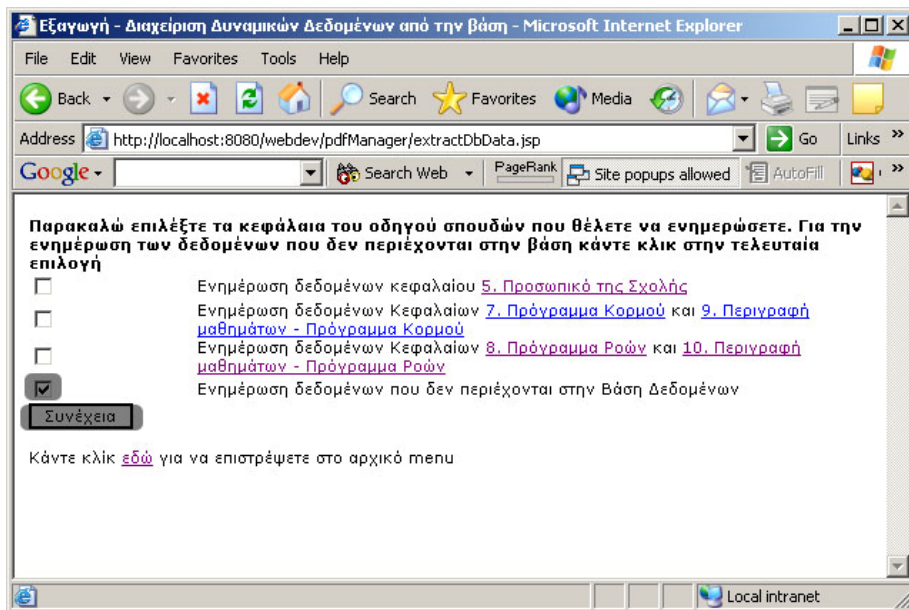
Όπως έχουμε προαναφέρει στην ενότητα 5.2.2, κάποια δεδομένα απαραίτητα για την δημιουργία κάποιων κεφαλαίων δεν υπάρχουν στην βάση της σχολής, επομένως δημιουργήθηκε μια φόρμα διαχείρισης των δεδομένων αυτών. Παρακάτω περιγράφεται τα βήματα που πρέπει να ακολουθήσει ο χρήστης για να αλλάξει αυτά τα δεδομένα και να ενημερώσει τον Οδηγό Σπουδών.

A) Ο χρήστης επιλέγει από το βασικό μενού της εφαρμογής “Διαχείριση Δυναμικού Περιεχομένου”.



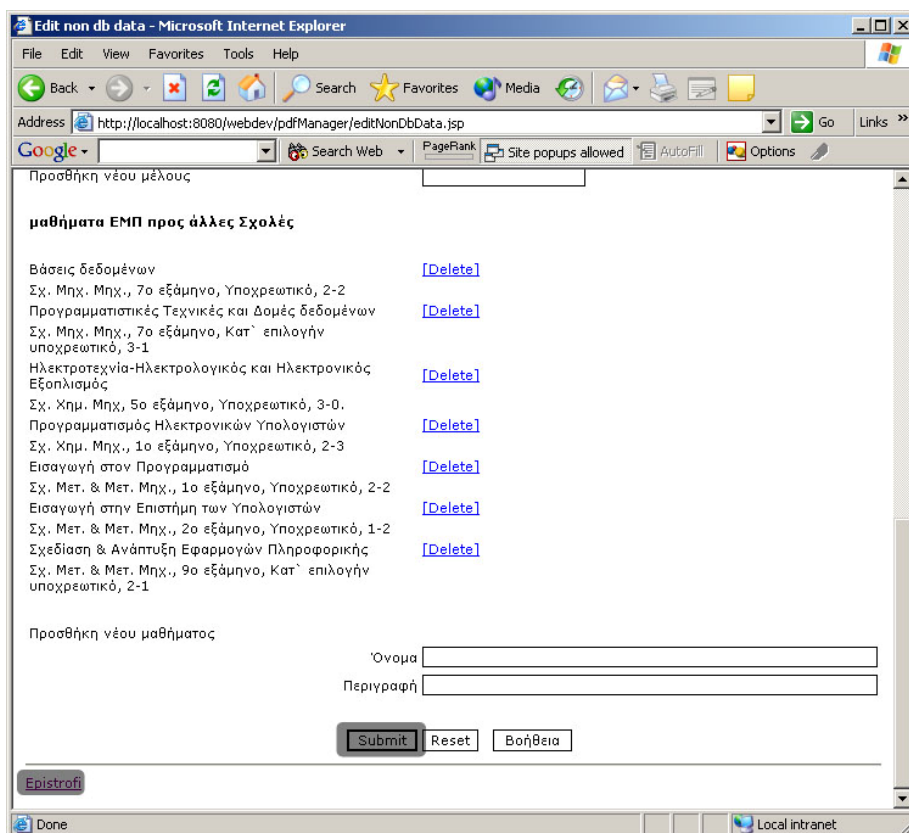
Σχήμα 6.3Α

B) Κατόπιν επιλέγει την τελευταία επιλογή “Ενημέρωση δεδομένων που δεν περιέχονται στην Βάση Δεδομένων” και κατόπιν κάνει κλικ στο “Συνέχεια”, όπως μπορούμε να δούμε στο παρακάτω Σχήμα.



Σχήμα 6.3B

Γ) Η επόμενη σελίδα που ο χρήστης βλέπει, περιέχει όλες αυτές τις πληροφορίες. Εδώ ο χρήστης αλλάζει τις πληροφορίες που θέλει, και κατόπιν επιλέγει Submit. Μόλις οι πληροφορίες καταχωρηθούν, ο χρήστης μπορεί να επιλέξει επιστροφή, και να επιστρέψει στην προηγούμενη σελίδα.



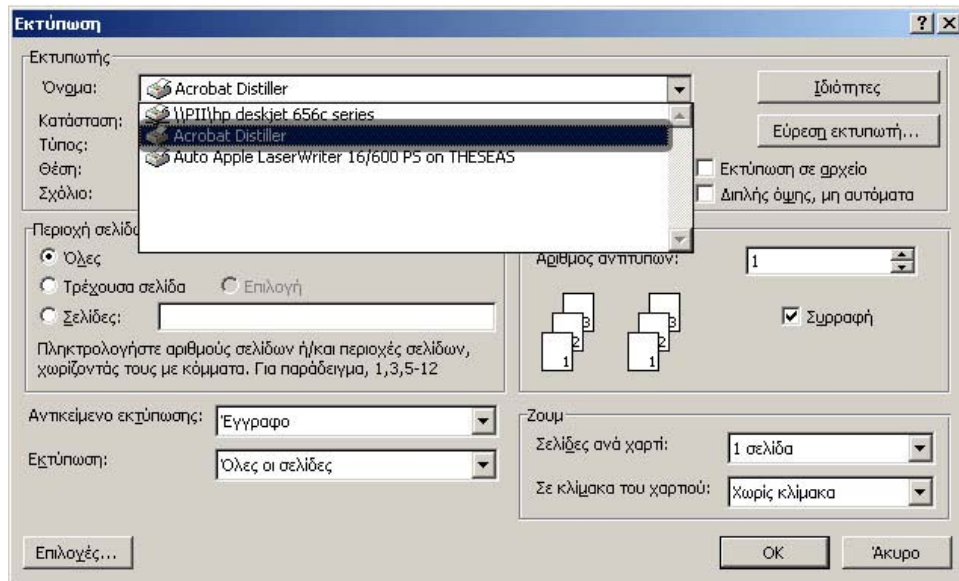
Σχήμα 6.3Γ

Δ) Εκεί, με τρόπο παρόμοιο με αυτόν του σχήματος 6.2B επιλέγει ενημέρωση των κεφαλαίων στα οποία περιέχεται η πληροφορία που άλλαξε, και τέλος από το “Σύνθεση τελικού οδηγού σπουδών”, με τον τρόπο που έχει προαναφερθεί, ο χρήστης δημιουργεί τον τελικό οδηγό σπουδών.

6.4 Προσθήκη νέου κεφαλαίου στον οδηγό.

Σε αυτό το σημείο θα περιγράψουμε αναλυτικά τα βήματα που πρέπει να κάνει ένας χρήστης για να προσθέσει ένα νέο στατικό κεφάλαιο στον οδηγό σπουδών, από την αρχή, από την στιγμή δηλαδή που θα έχει το αρχείο doc τοπικά στον υπολογιστή του, μέχρι να το προσαρτήσει στον οδηγό σε pdf μορφή και να τον ξαναδημιουργήσει.

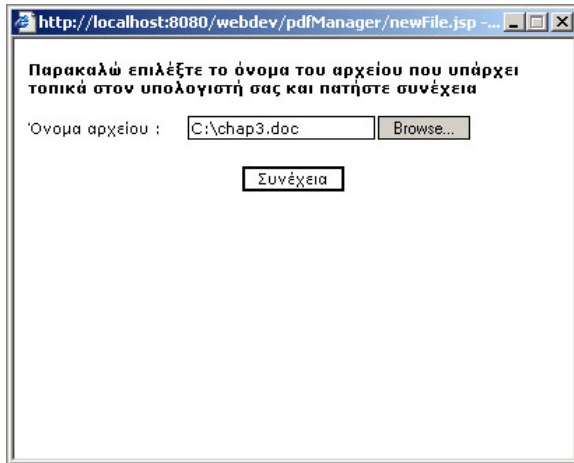
A) Έστω λοιπόν ότι ο χρήστης έχει δημιουργήσει το απαραίτητο αρχείο doc (έστω chap3.doc), και το έχει αποθηκεύσει στον υπολογιστή του. Τώρα, πρέπει αυτό το αρχείο να το μετατρέψει σε pdf, έτσι ώστε να ανεβάσει και τα 2 αρχεία στον server. Η μετατροπή του doc σε pdf γίνεται από το πρόγραμμα MS Word. Ο χρήστης ανοίγει το αρχείο doc και από το κυρίως μενού επιλέγει Αρχείο -> Εκτύπωση. Από εκεί επιλέγει για εκτυπωτή τον Acrobat Distiller (Εφόσον το πρόγραμμα Adobe Acrobat είναι εγκατεστημένο στον υπολογιστή του) και ακολουθώντας τα απαραίτητα βήματα, θα δημιουργηθεί το pdf αρχείο, τοπικά στον υπολογιστή του (με όνομα chap3.pdf).



Σχήμα 6.4A

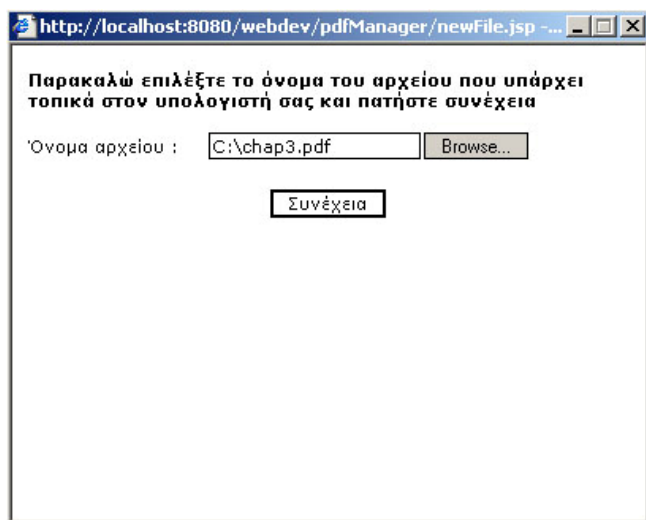
Β) Το επόμενο βήμα του χρήστη είναι να ανεβάσει τα αρχεία που δημιούργησε (doc και pdf) στον server. Η σειρά που ο χρήστης θα ακολουθήσει δεν έχει ιδιαίτερη σημασία. Στο παράδειγμα αυτό θα ανεβάσουμε πρώτα το doc. Ο χρήστης τώρα πρέπει να συνδεθεί με την δικτυακή εφαρμογή, να επιλέξει από το κεντρικό μενού της εφαρμογής “Διαχείριση Στατικού Περιεχομένου” και κατόπιν Προσθήκη, όπως φαίνεται στο παρακάτω σχήμα.

Γ) Τώρα, στο παράθυρο που εμφανίζεται στον χρήστη, αυτός επιλέγει το char3.doc από τον τοπικό υπολογιστή του (κάνοντας κλικ στο Browse εμφανίζονται τα περιεχόμενα του υπολογιστή του, από όπου επιλέγει και το αρχείο). Κάνοντας κλικ στο συνέχεια το αρχείο τοποθετείται στον server. Ο χρόνος για αυτή την ενέργεια ποικίλει ανάλογα την ταχύτητα της σύνδεσης και το μέγεθος του αρχείου.



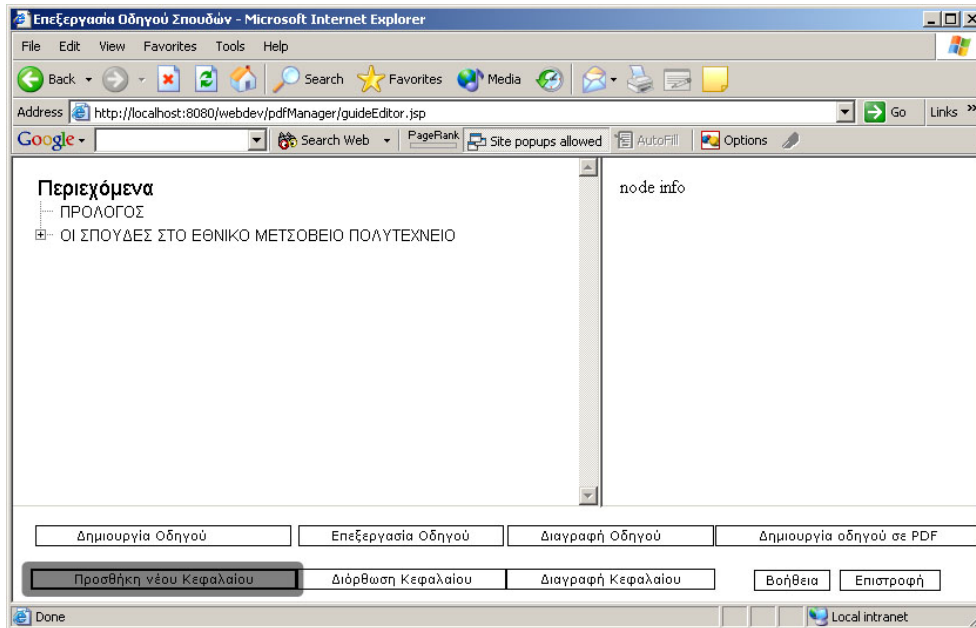
Σχήμα 6.4B

Δ) Κατόπιν ο χρήστης ακολουθεί την ίδια διαδικασία για να τοποθετήσει και το αρχείο pdf στον server (Σχήμα 6.4Γ)



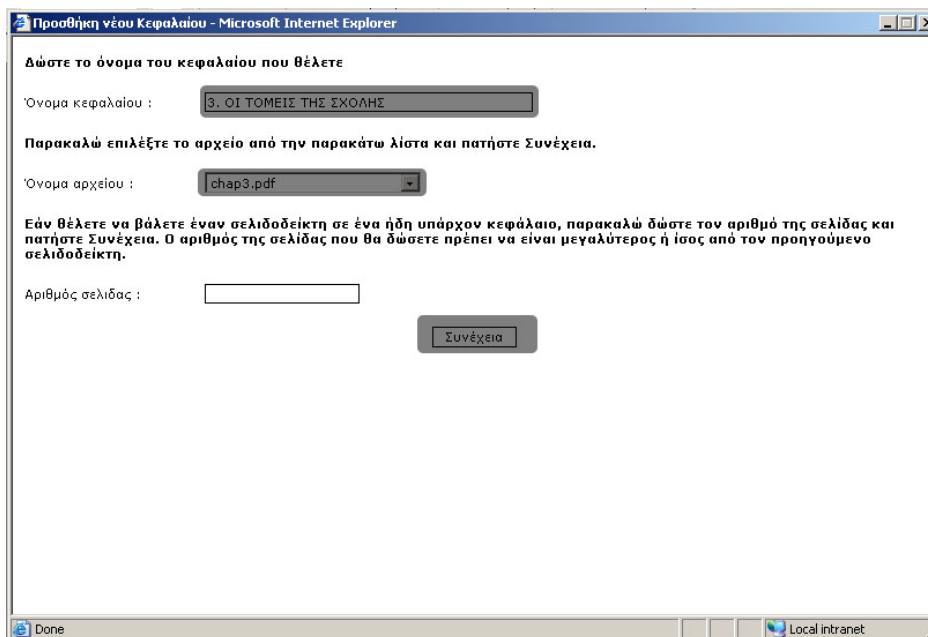
Σχήμα 6.4Γ

Ε) Αφού έχουν τοποθετηθεί και τα 2 αρχεία στον server, τώρα ο χρήστης πρέπει να τα ενώσει με τα άλλα αρχεία για να δημιουργήσει τον τελικό οδηγό σπουδών. Έτσι, επιλέγει από το κεντρικό μενού “Σύνθεση τελικού οδηγού Σπουδών” και επιλέγει τον οδηγό σπουδών (telikosodigos.xml στο παράδειγμά μας) από το κουμπί Επεξεργασία οδηγού. Κατόπιν, ο χρήστης επιλέγει προσθήκη νέου κεφαλαίου (Σχήμα 6.4Δ).



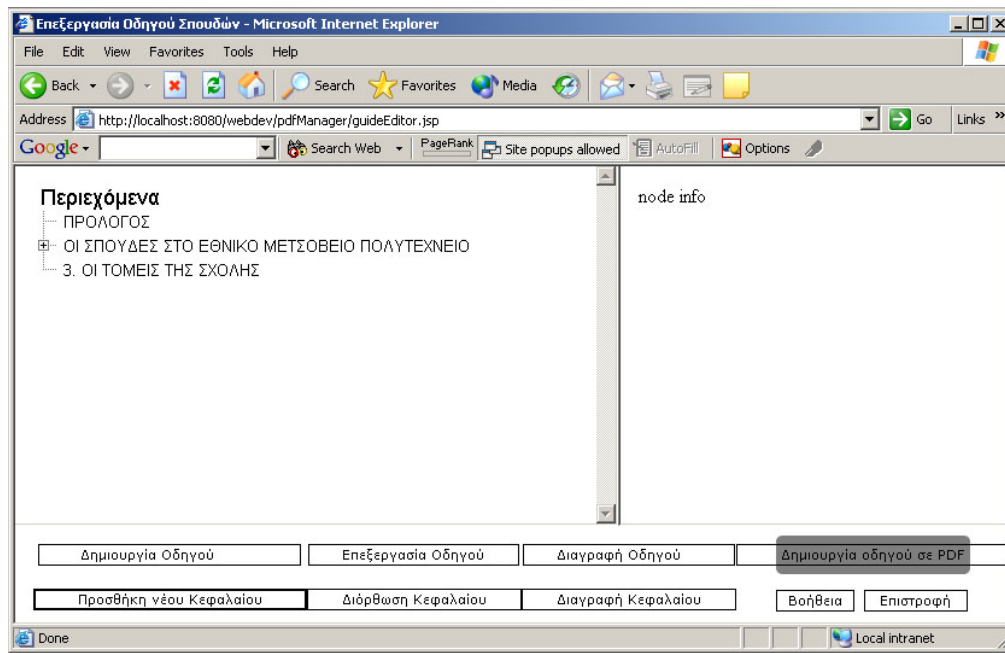
Σχήμα 6.4Δ

ΣΤ) Στο παράθυρο που εμφανίζεται στον χρήστη, αυτός βάζει το όνομα του κεφαλαίου (3. ΟΙ ΤΟΜΕΙΣ ΤΗΣ ΣΧΟΛΗΣ), επιλέγει το chap3.pdf από την αντίστοιχη επιλογή, και κατόπιν επιλέγει συνέχεια.



Σχήμα 6.4Ε

Ζ) Η προσάρτηση πλέον έγινε. Τώρα ο χρήστης επιλέγει Δημιουργία οδηγού σε PDF (βλέπε παρακάτω σχήμα) και ο οδηγός δημιουργείται πάλι, ανανεωμένος αυτή τη φορά.



Σχήμα 6.4ΣΤ

7. Συμπεράσματα

Στόχος της εφαρμογής αυτής ήταν η διαχείριση περιεχομένου του οδηγού σπουδών της σχολής με έναν εύκολο τρόπο, έτσι ώστε να μπορεί να την κάνει κάποιος χειριστής Η/Υ που δεν χρειάζεται να έχει ιδιαίτερες γνώσεις. Έγινε μεγάλη προσπάθεια έτσι ώστε το περιβάλλον χρήσης να είναι ιδιαίτερα φιλικό και κατανοητό, και υπάρχει βοήθεια on – line για κάθε βήμα που ο χρήστης θέλει να κάνει.

Το γεγονός ότι ο οδηγός σπουδών για να δημιουργηθεί θέλει την συνεργασία πολλών ατόμων, όπου ο καθένας είναι υπεύθυνος για το περιεχόμενο ενός συγκεκριμένου μέρους προβάλλει την αναγκαιότητα η πληροφορία να είναι προσβάσιμη σε πολλά άτομα ταυτόχρονα. Επίσης, οι αλλαγές που κάθε άτομο κάνει πρέπει να είναι ορατές και στους άλλους.

Η εφαρμογή δημιουργήθηκε έχοντας κατά νου τις γενικότερες αρχές που πρέπει να διέπει μια εφαρμογή διαχείρισης περιεχομένου (content management). Έτσι, έγινε προσπάθεια διατήρησης της πληροφορίας σε ένα μέρος μόνο, και να μην υπάρχουν ταυτόχρονες διαφορετικές εκδόσεις της ίδιας πληροφορίας. Το γεγονός ότι η εφαρμογή είναι πλήρως δικτυακή έχει κάποια θετικά στοιχεία που αναφέρονται παρακάτω :

- Η εφαρμογή είναι προσβάσιμη σε οποιονδήποτε έχει σύνδεση στο Internet και έχει κωδικό πρόσβασης στο admin site των ηλεκτρολόγων. Κατ' αυτόν τον τρόπο αποφεύγονται προβλήματα που δημιουργούνται κατά την εγκατάσταση της εφαρμογής σε διαφορετικά συστήματα, καθώς τώρα η εφαρμογή χρειάζεται εγκατάσταση μόνο στον server της σχολής.
- Τα δεδομένα υπάρχουν μόνο σε ένα μέρος στον server της σχολής, αποφεύγοντας έτσι το πρόβλημα των ταυτόχρονων εκδόσεων που θα προέκυπτε αλλιώς.
- Μπορεί εύκολα ο διαχειριστής του server να καταργήσει ή να δώσει πρόσβαση στα δεδομένα αυτά σε έναν συγκεκριμένο χρήστη.
- Εάν η εφαρμογή ανανεωθεί και αλλάξει, η εγκατάστασή της θα γίνει μόνο στο μηχάνημα του server και όχι σε κάθε μηχάνημα του χρήστη ξεχωριστά.
- Ο διαχειριστής του site μπορεί να κρατά αντίγραφα ασφαλείας της πληροφορίας εύκολα και χωρίς αυτό να φαίνεται και να ενοχλεί τον απλό χρήστη.

Βέβαια, πολλά μπορούν να γίνουν ακόμα για να κάνουν την εφαρμογή πιο λειτουργική.

Κατ' αρχάς, αυτό που θα μπορούσε να ρυθμιστεί, θα ήταν τα αρχεία doc που ο χρήστης αλλάζει να δημιουργούνται αυτόματα σε pdf στον server, και όχι να χρειάζεται να τα δημιουργεί ο χρήστης τοπικά στον υπολογιστή του. Για να γίνει όμως αυτό, θα πρέπει να αλλαχθεί εντελώς η τεχνολογία του server, και από λύσεις Linux - Resin, πρέπει να μεταφερθεί σε πλατφόρμα Windows – IIS. Αυτό γιατί πρέπει να εγκατασταθεί σε αυτή την περίπτωση MS Word στον server, έτσι ώστε να μπορεί να επεξεργάζεται τα αρχεία αυτά και να δημιουργεί τα pdf αυτόματα.

Επιπλέον, καλό θα ήταν να προστεθεί και μια λειτουργικότητα τύπου logging των ενεργειών των χρηστών. Δηλαδή καταγραφή στον server των δραστηριοτήτων των χρηστών πάνω στα αρχεία. Πχ ποιος ανέβασε και πότε ένα συγκεκριμένο αρχείο, πότε και από ποιον άλλαξε ένα ήδη υπάρχον, κα.

Θα μπορούσε ακόμα να ρυθμιστούν τα επίπεδα πρόσβασης των χρηστών πάνω στην πληροφορία. Δηλαδή ποιοι χρήστες μπορούν να αλλάξουν την πληροφορία, και ποιοι μπορούν μόνο να την προβάλλουν.

Τέλος, καλό θα ήταν να γίνουν οι απαραίτητες αλλαγές στην αρχιτεκτονική της εφαρμογής έτσι ώστε η πληροφορία να βρίσκεται μόνο στην βάση δεδομένων της σχολής, και ο οδηγός σπουδών να δημιουργείται ολόκληρος δυναμικά σε pdf από xml με τις συναρτήσεις for. Θα πρέπει να δημιουργηθούν οι απαραίτητοι πίνακες στην βάση που θα περιέχουν τα δεδομένα, οι σελίδες jsp που θα διαχειρίζονται τα δεδομένα αυτά και τα αρχεία xsl που θα περιγράφουν πως αυτά θα μετασχηματιστούν.

Πιστεύω πως η εφαρμογή αυτή παρά τις ατέλειές της μπορεί να αποτελέσει αρκετά χρήσιμο εργαλείο για την δημιουργία και την διαχείριση του οδηγού σπουδών της σχολής, καθώς και οποιουδήποτε άλλου εγγράφου που χρειάζεται την ταυτόχρονη συνεργασία πολλών ατόμων για την δημιουργία του.

ΠΑΡΑΡΤΗΜΑ - ΚΩΔΙΚΑΣ

Εδώ παραθέτουμε τον κώδικα σε java που δημιουργήσαμε για τις ανάγκες της διπλωματικής, τόσο τις βιβλιοθήκες όσο και τις σελίδες jsp που δημιουργήσαμε.

Π.1 *gr.ntua.ece.util.NavigateXmlUtils.java*

```
/*
 * Created on 15 Σεπ 2004
 *
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
package gr.ntua.ece.util;
import java.io.BufferedInputStream;
import java.io.ByteArrayInputStream;
import java.io.DataInputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.io.StringBufferInputStream;
import java.io.StringReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;
import org.dom4j.Document;
import org.dom4j.Element;
import org.dom4j.Node;
import org.dom4j.io.SAXReader;
import com.lowagie.text.pdf.PdfReader;
/**
 * @author Constantinou
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class NavigateXmlUtils {
    Document doc;
    String webApPath;
    String imageI = "<img src=\"images/I.gif\"
align=\"absmiddle\">";
    String imageT = "<img src=\"images/T.gif\"
align=\"absmiddle\">";
    String imageL = "<img src=\"images/L.gif\"
align=\"absmiddle\">";
    String imageWhite = "<img src=\"images/white.gif\"
align=\"absmiddle\">";
    String imageHtmlIcon = "<img src=\"images/htmlicon.gif\"
align=\"absmiddle\" class=\"icon\">";
    public NavigateXmlUtils(Document doc){
        this.doc = doc;
    }
    public NavigateXmlUtils(String xmlPath,String webApPath){
```

```

        this.webApPath = webApPath;
        SAXReader reader = new SAXReader();
        try {
            doc = reader.read(new File(xmlPath));
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    public int getPageNumber(String pdfFileName){
        int pageNum=0;
        String pdfFilePath = webApPath + "pdfManager" +
File.separator + "pdfs" + File.separator + pdfFileName;
        System.out.println("pdffilePath : " + pdfFilePath);
        try{// we retrieve the total number of pages
            //if the file is somewhere else
            PdfReader reader = new PdfReader(pdfFilePath);
            pageNum = reader.getNumberOfPages();

        }
        catch (Exception ex)
        {
            ex.printStackTrace();
        }
        return pageNum;
    }

    public List getDataNeeded(){
        //this method returns a List of arrays needed from the
Concat Class to
        //construct the pdf file. The arrays are filled from the
//xmltree which holds the information
        ArrayList tempList = new ArrayList();
        List xpathList = getXpathList();
        String args[] = new String[xpathList.size()+1];
        String title[] = new String[xpathList.size()];
        int page[] = new int[xpathList.size()];
        boolean untitled[] = new boolean[xpathList.size()];
        int level[] = new int[xpathList.size()];
        int x=1;
        int y=0;
        int allpage=1;
        int fatherpage=2;
        for (Iterator iter = xpathList.iterator();
iter.hasNext();) {
            String xpathStr = (String) iter.next();
            Element curElem =
(Element)doc.selectSingleNode(xpathStr);
            String chapterTitle =
curElem.element("chaptertitle").getText();
            String chapterType = curElem.valueOf("@type");
            boolean isLeaf = (curElem.nodeCount()==1);
            int depth = findDepth(xpathStr);
            level[y]= depth;
            if (chapterType.equals("file")){
                //einai file...
                page[y]=allpage;
                fatherpage=allpage;

            }

            allpage=allpage+getPageNumber(curElem.valueOf("@source"));

```

```

pdf //edo o synolikos arithmos tvn selidvn toy

//ayksanetai oso einai o arithmos toy
//sygkekrimenoy pdf poy prostitetai
title[y]=chapterTitle;
y++;
args[x]=webApPath + "pdfManager" +
File.separator + "pdfs" + File.separator +
curElem.valueOf("@source");
x++;
//System.out.println("numOfPages : " +
getPageNumber(curElem.valueOf("@source")) );
}else{
//einai bookmark
//Take information from a subchapter
//Set page for the subchapter in the new pdf
Integer bookmarkPageNum = new
Integer(curElem.valueOf("@page"));
page[y] = fatherpage +
bookmarkPageNum.intValue() - 1;
//System.out.println("pagey " + page[y] );
title[y]=chapterTitle;
y++;
x++;
}
}
tempList.add(args);
tempList.add(title);
tempList.add(page);
tempList.add(untitled);
tempList.add(level);
return (List)tempList;
}
public List getXpathList(){
Element rootElem = doc.getRootElement();
String xpath="";
ArrayList xpathList = new ArrayList();
ArrayList xpathList2 = new ArrayList();
List elems = doc.selectNodes("//chapter");
for ( Iterator iter3 = elems.iterator();
iter3.hasNext(); ) {
Node currnode = (Node)iter3.next();
Element currElem = (Element)currnode;
xpath = currnode.getUniquePath();
char[] tempArr = new char[xPath.length()+200];

int i = 0;
int j = 0;
while ( i < xpath.length() ) {
if ((xpath.charAt(i) == '[')){

char endChar=' ';
int k=i;
while (endChar!=']'){
endChar = xpath.charAt(k);
k++;
};
int numLength = k-i-2;
//System.out.println("numLength : " +
numLength);

if (numLength==1){

```

```

        tempArr[j]=xPath.charAt(i);
        tempArr[j+1]='0';
        tempArr[j+2]=xPath.charAt(i+1);
        i++;
    }
    else if(numLength==2){
        tempArr[j]=xPath.charAt(i);
        tempArr[j+1]=xPath.charAt(i+1);
        tempArr[j+2]=xPath.charAt(i+2);
        i=i+2;
    }
    j=j+2;

    }else{
        tempArr[j]=xPath.charAt(i);
    }
    i++;
    j++;
}
xpath = String.valueOf(tempArr);
xpath=xPath.trim();
//System.out.println("xpath : " + xpath);
xpathList.add(xpath);
//System.out.println("xpathTrimmed : " +
xpath);
//System.out.println("childIndex : " +
currElem.valueOf( "string(position())"));
}
Collections.sort(xpathList);
//edo dimiourgitai mia lista me ta chapter nodes.
//to unique xpath toy kathe node prostithetai stin xpathList.
//i xpath list ginetai alphabetic sort me tin Collections.sort
//kai etsi to kathe paidi einai kato apo ton patera toy
String a;
Node nd;
return (List)xpathList;
}

public String printFather(String name,int depth, boolean
islast, String titlos , boolean[] depthArray, String xpath){
String tempStr = "";
for (int i = 0; i < depth-1; i++) {
tempStr = tempStr + "\t";
}
tempStr = "<div ID=\"\" + name + \"Parent\"
class=\"parent\">";
for (int i = 0; i < depth - 1; i++) {
if (depthArray[i+1]){
tempStr+=imageWhite;
}else
{
tempStr+=imageI;
}
}
if (islast){
tempStr+= "<font color=\"black\" size=\"2\"
class=\"item\" face=\"Arial\"><a name=\"\" +name + \"a\" \" +
printLink(xpath) + \" onClick=\"FolderExpand('\" + name +
\", 'last',this)\">";
}else{

```

```

        tempStr+= "<font color=\"black\" size=\"2\"
class=\"item\" face=\"Arial\"><a name=\"\" +name + "a\" \" \" +
printLink(xpath) + \" onClick=\"FolderExpand('\" + name +
\"', '', this)\">";
    }
    if (islast){
        tempStr+= "<img name=\"\" + name+ \"Tree\"
src=\"images/Lplus.gif\" align=\"absmiddle\" border=\"0\">";
    }else{
        tempStr+= "<img name=\"\" + name+ \"Tree\"
src=\"images/Tplus.gif\" align=\"absmiddle\" border=\"0\">";
    }
    tempStr+= "<img name=\"\"+ name + \"Folder\"
src=\"images/foldericon.gif\" class=\"icon\" align=\"absmiddle\"
border=\"0\">"+ titlos + "</a></font></div>\n";

    for (int i = 0; i < depth-1; i++) {
        tempStr = tempStr + "\t";
    }
    tempStr+= "<div ID=\"\" +name+ \"Child\" CLASS=\"child\">";
    //to <div> poy anigei apo edo, prepei na kleisei kapoy
    //tha to klisei to teleytaio paidi toy father... (me tin
printChild)
    return tempStr;
}

public String printChild(String name,int depth, boolean islast,
String titlos , boolean[] depthArray, String xpath){
    String tempStr="";
    for (int i = 0; i < depth; i++) {
        tempStr = tempStr + "\t";
    }
    for (int i = 0; i < depth - 1; i++) {
        if (depthArray[i+1]){
            tempStr+=imageWhite;
        }else
        {
            tempStr+=imageI;
        }
    }
    if (islast){
        tempStr+= imageL;
    }else{
        tempStr+= imageT;
    }
    tempStr+=imageHtmlIcon;
    tempStr+= "<font color=\"black\" size=\"2\"
face=\"Arial\">";
    tempStr+= "<a class=\"item\" name = \"\" + name + \"a\"
onClick = \"javascript:setSelectedNode(this)\">\" + printLink(xpath) +
\">\" + titlos+ "</a></font><br>\n";
    if (islast){
        for (int i = 0; i < depth-1; i++) {
            //tempStr = tempStr + "\t";
            //tempStr+= "</div>";
        }
        tempStr+= "</div>";
    }
    return tempStr;
}

```

```

public String printLink(String xPath){
    String tempStr="";
    Element curElem = (Element)doc.selectSingleNode(xPath);
    String elemType = curElem.attributeValue("type");
    if (elemType.equals("file")){
        tempStr = " href=\"pdfs/";
        tempStr+= curElem.attributeValue("source") + "\"";
    }
    if (elemType.equals("bookmark")){
        tempStr = " href=\"pdfs/";
        tempStr+=curElem.attributeValue("father")+ "#page="
+ curElem.attributeValue("page") + "\"";
    }
    tempStr+= " target= \"mainFrame\"";
    return tempStr;
}

public int findDepth(String xPath){
    int depth=0;
    for (int i = 0; i < xPath.length()-1; i++) {
        if (xPath.charAt(i)=='/' ){
            depth++;
        }
    }
    return depth - 1;
}
}
}

```

Π.2 gr.ntua.ece.util.XmlUtils.java

```

/*
 * Created on 14 Σεπ 2004
 *
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
package gr.ntua.ece.util;
import java.io.File;
import java.io.FileWriter;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.DataOutputStream;
import java.io.FileOutputStream;
import java.io.StringReader;
import java.io.ByteArrayInputStream;
import java.io.DataInputStream;
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;

```

```

import java.util.List;
import java.util.Vector;
import org.apache.turbine.util.db.pool.DBConnection;
import org.dom4j.Document;
import org.dom4j.DocumentException;
import org.dom4j.DocumentHelper;
import org.dom4j.Element;
import org.dom4j.Node;
import org.dom4j.XPath;
import org.dom4j.io.OutputFormat;
import org.dom4j.io.SAXReader;
import org.dom4j.io.XMLWriter;
/**
 * @author Constantinou
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class XmlUtils {
    private String filename;
    private String query;
    private String dir2write;
    Connection con = null;
    Vector Tables = new Vector();
    private String webApPath;
    private String seperatorChar;
    public XmlUtils(String webApPath) {
        this.webApPath = webApPath;
        this.seperatorChar = File.separator;
        dir2write = webApPath + "pdfManager" + seperatorChar +
"data" + seperatorChar;
    } //end constructor

    public void update5() {
        DBConnection datasource = null;
        try {
            DBPoolAdministration.pool.init();
            datasource =
DBPoolAdministration.pool.getConnection("struts");
            con = datasource.getConnection();
            DatabaseMetaData dmd = con.getMetaData();
            ResultSet rs=dmd.getTables(null,null,"%",null);
            String tblname;
            boolean ok;
            while (rs.next()) {
                Tables.addElement(rs.getString("TABLE_NAME"));
            }
            if (Tables.isEmpty()) throw new SQLException("No
tables found in database");
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        File chap5xmlFile = new File(webApPath + File.separator +
"pdfManager" + File.separator + "data" + File.separator +
"chap5.xml");
        Document document = null;
        Element root = null;
        if (chap5xmlFile.exists()){
            //yparxei to xml, ara fortose to se document kai
proxora

```

```

        SAXReader reader = new SAXReader();
        try {
            document = reader.read(chap5xmlFile);
            Node detachedNode
document.selectSingleNode("/database/database_data").detach();
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        root
(Element)document.selectSingleNode("database");
    }else {
        //den yparxei, ftaksto apo tin arxi
        document = DocumentHelper.createDocument();

        root = document.addElement( "database" );
        Element addinfo = root.addElement("addinfo");
    }
    Element dbData = root.addElement("database_data");
    Element division;
    Element profRank;
    Element row;
    Element nameGr;
    Element surnameGr;
    Iterator ITables = Tables.iterator();
    try {
        Statement stmt = con.createStatement ();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    String tablename,data,typestr;
    ResultSet prkrs;
    ResultSet rs;
    Statement stmt = null;
    try {
        stmt = con.createStatement();
    } catch (SQLException e2) {
        // TODO Auto-generated catch block
        e2.printStackTrace();
    }
    ResultSetMetaData rsmd;
    Vector keys = new Vector();
    Vector columns = new Vector();

    ITables = Tables.iterator();
//edo to xml gemizei me dedomena
    try {
        for(int y=1;y<=7;y++)
        {
            division = dbData.addElement("Division" + y);
            for (int g=1;g<=5;g++)
            {
                profRank
division.addElement(returnRank(g));
                query="SELECT          NameGR,SurnameGR          FROM
AcademicC,Staff          where          AcademicC.StaffID=Staff.StaffID          AND
Division="+y
                +" AND Rank=\""+returnRank(g)+"\" ";
                rs = stmt.executeQuery(query);
                rsmd = rs.getMetaData();
            }
        }
    }

```



```

        while (rs.next()) {
            row = profRank.addElement("row");
            for (int i = 1;
i<=rsmd.getColumnCount(); i++) {
                nameGr =
row.addElement(rsmd.getColumnName(i));

                nameGr.setText(rs.getString(i));
            }
        }
    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
//the xml is saved in the hdd
FileWriter out;
OutputFormat format = new OutputFormat(" ", true);
format.setEncoding("ISO-8859-7");
try {
    out = new FileWriter(dir2write + "chap5.xml");
    XMLWriter writer = new XMLWriter( out, format );
    writer.setMaximumAllowedCharacter(255);
    writer.write( document );
    out.close();
} catch (IOException e3) {
    // TODO Auto-generated catch block
    e3.printStackTrace();
}

PdfCreator creator = new PdfCreator(webApPath);
creator.setXmlFile(dir2write + "chap5.xml");
creator.setXsltFile(dir2write + "chap5.xsl");
creator.setOutputFile(webApPath + "pdfManager" +
File.separator + "pdfs" + File.separator + "chap5(dynamic).pdf");
System.out.println("dir2write" + dir2write);
try {
    creator.transform();
} catch (Exception e4) {
    e4.printStackTrace();
}
}

public void update79() throws Exception{
    DBConnection datasource = null;
    try {
        DBPoolAdministration.pool.init();
        datasource =
DBPoolAdministration.pool.getConnection("struts");
        con = datasource.getConnection();
        DatabaseMetaData dmd = con.getMetaData();
        ResultSet rs=dmd.getTables(null,null,"%",null);
        String tblname;
        boolean ok;
        while (rs.next()) {
            Tables.addElement(rs.getString("TABLE_NAME"));
        }
        if (Tables.isEmpty()) throw new SQLException("No
tables found in database");
    } catch (Exception e) {
        // TODO Auto-generated catch block

```

```

        e.printStackTrace();
    }
    Document document = DocumentHelper.createDocument();
    Element root = document.addElement("database");
    Element dbData = root.addElement("database_data");
    Element semester;
    Element row;
    Element dataElem;
    Element professors;
    Element professor;
    Iterator ITables = Tables.iterator();
    Statement stmt = con.createStatement();
    String tablename,data,typestr;
    ResultSet prkrs;
    ResultSet rs;
    ResultSetMetaData rsmd;
    Vector keys = new Vector();
    Vector columns = new Vector();
    ITables = Tables.iterator();
    for(int y=1;y<=5;y++)
    {
        query="Select
Courses.CourseID,Code,Courses.TitleGR,LectureHours,LabHours,CourseCat
egories.TitleGR,Courses.DescriptionGR "+
        "from Courses,CourseCategories where "+
        "CourseCategories.CourseCategoryID=Courses.Category
AND Semester="+y;
        rs = stmt.executeQuery(query);
        rsmd = rs.getMetaData();
        semester = dbData.addElement("semester_" + y);
        while (rs.next()) {
            row = semester.addElement("row");
            for (int i = 1; i<=rsmd.getColumnCount(); i++) {
                dataElem =
row.addElement(rsmd.getColumnName(i) + "_" +i);
                dataElem.setText(rs.getString(i));
            }
        }
        query="select CourseID,NameGR,SurnameGR from"+
        "          Staff,CourseLecturers          where
CourseLecturers.StaffID=Staff.Staffid";
        rs = stmt.executeQuery(query);
        rsmd = rs.getMetaData();
        professors = dbData.addElement("Professors");
        boolean first=true;
        while (rs.next()) {
            row = professors.addElement("row");

            row.addAttribute(rsmd.getColumnName(1),rs.getString(1));
            professor = row.addElement("professor");
            dataElem =
professor.addElement(rsmd.getColumnName(2));
            dataElem.setText(rs.getString(2));
            dataElem =
professor.addElement(rsmd.getColumnName(3));
            dataElem.setText(rs.getString(3));
        }
    }
    //the xml is saved in the hdd
    FileWriter out;
    OutputFormat format = new OutputFormat(" ", true);

```

```

format.setEncoding("ISO-8859-7");
try {
    out = new FileWriter(dir2write + "chap7-9.xml");
    XMLWriter writer = new XMLWriter( out, format );
    writer.setMaximumAllowedCharacter(255);
    writer.write( document );
    out.close();
} catch (IOException e3) {
    // TODO Auto-generated catch block
    e3.printStackTrace();
}

PdfCreator creator = new PdfCreator(webApPath);
creator.setXmlFile(dir2write + "chap7-9.xml");
creator.setXsltFile(dir2write + "chap7.xsl");
creator.setOutputFile(webApPath + File.separator +
"pdfs" + File.separator + "chap7(dynamic).pdf");
try {
    creator.transform();
} catch (Exception e4) {
    e4.printStackTrace();
}

PdfCreator creator2 = new PdfCreator(webApPath);
creator2.setXmlFile(dir2write + "chap7-9.xml");
creator2.setXsltFile(dir2write + "chap9.xsl");
creator2.setOutputFile(webApPath + File.separator +
"pdfs" + File.separator + "chap9(dynamic).pdf");
try {
    creator2.transform();
} catch (Exception e4) {
    e4.printStackTrace();
}
}

public void update810() throws Exception{
    DBConnection datasource = null;
    try {
        DBPoolAdministration.pool.init();
        datasource =
DBPoolAdministration.pool.getConnection("struts");
        con = datasource.getConnection();
        DatabaseMetaData dmd = con.getMetaData();
        ResultSet rs=dmd.getTables(null,null,"%",null);
        String tblname;
        boolean ok;
        while (rs.next()) {
            Tables.addElement(rs.getString("TABLE_NAME"));
        }
        if (Tables.isEmpty()) throw new SQLException("No
tables found in database");
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    File chap810xmlFile = new File(webApPath + File.separator
+ "pdfManager" + File.separator + "data" + File.separator + "chap8-
10.xml");
    Document document = null;
    Element root = null;
    if (chap810xmlFile.exists()){

```

```

//yparxei to xml, ara fortose to se document kai
proxora
SAXReader reader = new SAXReader();
try {
    document = reader.read(chap810xmlFile);
    Node detachedNode =
document.selectSingleNode("/database/database_data").detach();
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
root =
(Element)document.selectSingleNode("database");
}else {
    //den yparxei, ftaksto apo tin arxi
    document = DocumentHelper.createDocument();

    root = document.addElement( "database" );
    Element addinfo = root.addElement("addinfo");
}
Element dbData = root.addElement("database_data");
Element flow;
Element row;
Element dataElem;
Element dataElem2;
Element professors;
Element professor;
Iterator ITables = Tables.iterator();
Statement stmt = con.createStatement ();
String tablename,data,typestr;
ResultSet prkrs;
ResultSet rs;
ResultSetMetaData rsmd;
Vector keys = new Vector();
Vector columns = new Vector();
    ITables = Tables.iterator();
    for(int y=1;y<=14;y++)
    {
        query="Select
Courses.CourseID,Code,Courses.TitleGR,Semester,LectureHours,LabHours,
CourseCategories.TitleGR,Courses.DescriptionGR "+
"from
Courses,CourseCategories where "+
"CourseCategories.CourseCategoryID=Courses.Category AND
Flow="+y;
        rs = stmt.executeQuery(query);
        rsmd = rs.getMetaData();
        flow = dbData.addElement("flow_" + y);
        while (rs.next()) {
            row = flow.addElement("row");
            for (int i = 1; i<=rsmd.getColumnCount(); i++) {
                dataElem =
row.addElement(rsmd.getColumnName(i) +"_"+i);
                dataElem.setText(rs.getString(i));
            }
        }
    }
    query="select CourseID,NameGR,SurnameGR from"+
" Staff,CourseLecturers where
CourseLecturers.StaffID=Staff.Staffid";

```

```

        rs = stmt.executeQuery(query);
        rsmd = rs.getMetaData();
        professors = dbData.addElement("Professors");
        boolean first=true;
        while (rs.next()) {
            row = professors.addElement("row");

            row.addAttribute(rsmd.getColumnName(1),rs.getString(1));
            professor = row.addElement("professor");
            dataElem =
professor.addElement(rsmd.getColumnName(2));
            dataElem.setText(rs.getString(2));
            dataElem2 =
professor.addElement(rsmd.getColumnName(3));
            dataElem2.setText(rs.getString(3));
        }
//the xml is saved in the hdd
FileWriter out;
OutputFormat format = new OutputFormat(" ", true);
format.setEncoding("ISO-8859-7");
try {
    //System.out.println(document.asXML());
    out = new FileWriter(dir2write + "chap8-10.xml");
    XMLWriter writer = new XMLWriter( out, format );
    writer.setMaximumAllowedCharacter(255);
    writer.write( document );
    out.close();
} catch (IOException e3) {
    // TODO Auto-generated catch block
    e3.printStackTrace();
}
PdfCreator creator = new PdfCreator(webApPath);
creator.setXmlFile(dir2write + "chap8-10.xml");
creator.setXsltFile(dir2write + "chap8.xsl");
creator.setOutputFile(webApPath + File.separator +
"pdfs" + File.separator + "chap8(dynamic).pdf");
try {
    creator.transform();
} catch (Exception e4) {
    e4.printStackTrace();
}
PdfCreator creator2 = new PdfCreator(webApPath);
creator2.setXmlFile(dir2write + "chap8-10.xml");
creator2.setXsltFile(dir2write + "chap10.xsl");
creator2.setOutputFile(webApPath + File.separator +
"pdfs" + File.separator + "chap10(dynamic).pdf");
try {
    creator2.transform();
} catch (Exception e4) {
    e4.printStackTrace();
}
}

public void addChapterNode(String treeFileName, String
parentXPath ,String chapterName, String pdfFileName, String
bookmarkPage){
    SAXReader reader = new SAXReader();
    Document xml2load=null;
    String treeFileNamePath="";
    treeFileNamePath = webApPath + "pdfManager" +
seperatorChar + "xmlltrees" + seperatorChar + treeFileName;

```

```

        if (parentXpath.equals("m")) {
            parentXpath = "/root";
        }else{
            parentXpath = "/root/" + parentXpath;
        };
        try {
            xml2load = reader.read(new File(treeFileNamePath));
        } catch (DocumentException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        Node parentNode =
xml2load.selectSingleNode(parentXpath);
        Element parentElement = (Element)parentNode;
        Element chapterNode =
parentElement.addElement("chapter");
        Element chapterTitle =
chapterNode.addElement("chaptertitle");
        chapterTitle.setText(chapterName);
        if (bookmarkPage.equals("")){
            //den prokeitai gia bookmark
            //ara bazo type=file kai source=pdffilename sto chapter
node
            chapterNode.addAttribute("type","file");
            chapterNode.addAttribute("source",pdfFileName);
        }else{
            //prokeitai gia bookmark, ara bazotype=bookmark
            //father= to pdfonoma toy patera kai page= tin selida poy
edose o xristis
            chapterNode.addAttribute("type","bookmark");
            chapterNode.addAttribute("page",bookmarkPage);

            chapterNode.addAttribute("father",parentElement.attributeValue(
"source"));
        }
        // the xml is saved in the hdd
        OutputFormat format = new OutputFormat(" ", true);
        format.setEncoding("ISO-8859-7");
        FileWriter out;
        try {
            out = new FileWriter(treeFileNamePath);
            XMLWriter writer = new XMLWriter( out, format );
            writer.setMaximumAllowedCharacter(255);
            writer.write( xml2load );
            out.close();
        } catch (IOException e3) {
            // TODO Auto-generated catch block
            e3.printStackTrace();
        }
    }

    public void writeXml(Document xmlDoc, String xmlFileName){
        OutputFormat format = new OutputFormat(" ", true);
        format.setEncoding("ISO-8859-7");
        try {
            FileOutputStream fos = new FileOutputStream(new
File(xmlFileName));
            XMLWriter writer = new XMLWriter( fos, format );
            writer.setMaximumAllowedCharacter(255);
            writer.write( xmlDoc );
            writer.flush();
        }
    }

```

```

        fos.close();
    } catch (IOException e3) {
        // TODO Auto-generated catch block
        e3.printStackTrace();
    }
}

public void deleteChapter(String xPathStr, String xmlFileName) {
    SAXReader reader = new SAXReader();
    Document xmlDoc=null;
    try {
        xmlDoc = reader.read(new File(xmlFileName));
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    Element rootElem = xmlDoc.getRootElement();
    Node chapterNode = rootElem.selectSingleNode(xPathStr);
    Node detachedNode = chapterNode.detach();
    writeXml(xmlDoc, xmlFileName);
}

public String returnRank(int r)
{String value="";
    switch (r)
    {
        case 1:      value="Professor";
                    break;

        case 2:      value="Assistant";
                    break;

        case 3:      value="Associate";
                    break;

        case 4:      value ="Lecturer";
                    break;

        case 5:      value ="Emeritus";
                    break;

    }
    return value;
}
public static void main(String[] args) {
}
}

```

Π.3 gr.ntua.ece.util.fileUtils.java

```

/*
 * Created on 18 Σεπ 2004
 *
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Comments

```

```

*/
package gr.ntua.ece.util;
import java.io.File;
import java.io.FileWriter;
import java.io.FileNameFilter;
import java.io.IOException;
import org.dom4j.Document;
import org.dom4j.DocumentHelper;
import org.dom4j.Element;
import org.dom4j.io.OutputFormat;
import org.dom4j.io.XMLWriter;
/**
 * @author Constantinou
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class fileUtils {
    class dirFilter implements FileNameFilter {
        String afn;
        dirFilter(String afn) {
            this.afn = afn;
        }
        public boolean accept(File dir, String name) {
            boolean b = false;
            String f = new File(name).getName(); //strip
filename
            b = (f.indexOf(afn) == f.length() - 3) &&
(f.indexOf(" ") == -1);
            return b;
        }
    }

    public String[] getXmlDirContents(String dirName){
        File path = new File(dirName);
        String[] list;
        list = path.list(new dirFilter("xml"));
        return list;
    }
    public String[] getPdfDirContents(String dirName){
        File path = new File(dirName);
        String[] list;
        list = path.list(new dirFilter("pdf"));
        return list;
    }
    public String[] getDocDirContents(String dirName){
        File path = new File(dirName);
        String[] list;
        list = path.list(new dirFilter("doc"));
        return list;
    }
    public void createEmptyXmlFile(String filePath) throws
Exception{
        Document document = DocumentHelper.createDocument();
        Element root = document.addElement( "root" );
        OutputFormat format = new OutputFormat(" ", true);
        FileWriter out;
        out = new FileWriter(filePath);
        XMLWriter writer = new XMLWriter( out, format );
        writer.write( document );
        out.close();
    }
}

```



```

    }
    public boolean deleteFile (String filePath){
        File forDeletion = new File(filePath);
        return forDeletion.delete();
    }
    public boolean renameFile (String fromFilePath, String
toFilePath){
        File forRen = new File(fromFilePath);
        return forRen.renameTo(new File(toFilePath));
    }
}

```

7.4 addNewChapter.jsp

```

<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "java.io.File" %>
<%@ taglib uri="/WEB-INF/app.tld" prefix="app" %>

<%
fileUtils fUtils = new fileUtils();
String dirName
pageContext.getServletContext().getRealPath("/pdfManager/pdfs");
dirName = dirName + File.separator;
%>
<script language="javascript" type="text/JavaScript">
function getContents(){
    //get the values from the hidden fields in the tree.jsp
    //process them
    //and store them in hidden fields in this jsp page...
    selectedTree = opener.parent.frames[0].tempForm.floded.value
    parentNode
opener.parent.frames[0].tempForm.selectednode.value
    parentNode = parentNode.slice(0,parentNode.length-1)
    tempStr = ''
    re1 = new RegExp("chapter","g")
    re2 = new RegExp("_","g")
    re3 = new RegExp("\\[\\]", "g")
    parentNode = parentNode.replace(re1,'chapter[')
    parentNode = parentNode.replace(re2,']_')
    parentNode = parentNode.replace(re2,'/')
    parentNode = parentNode.replace(re3,'')
    lastCharAscii = parentNode.charCodeAt(parentNode.length-1)

    if ((lastCharAscii>=48) && (lastCharAscii<=57)){
        parentNode = parentNode + ']'
    }
    //edo an to teleytaio psifio einai digit, tote prostese kai
ena ]
    //gia na kleisei to bracket poy anoikse
    if (parentNode.charAt(parentNode.length-1)=='['){
        parentNode = parentNode.substr(0,parentNode.length-1)
    }
    addNewChapterForm.selectedTree.value=selectedTree
    addNewChapterForm.parentNode.value=parentNode
}

function formValidate(){
with (addNewChapterForm){

```

```

if (chapterName.value==''){
    alert('Παρακαλώ δώστε όνομα κεφαλαίου')
}else {
    if(((uploadedFileName.value=='
')&&(bookmarkPage.value==''))||((!(uploadedFileName.value=='
'))&&(!(bookmarkPage.value=='')))){
        alert('Παρακαλώ συμπληρώστε MONO ένα από τα παρακάτω
στοιχεία : \nΌνομα Αρχείου\nΑριθμός Σελίδας');
    }else{
        submit()
    }
}
}
}

</script>
<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
<head>
<title>Προσθήκη νέου Κεφαλαίου</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-
7">
</head>
<body bgcolor="white" onLoad = "getContents()">
<form action="chapterAdded.jsp" method = "POST" name =
"addNewChapterForm">
<input type="hidden" name = "selectedTree"/>
<input type="hidden" name="parentNode"/>
<table width = "100%" cellspacing="0" cellpadding="0" border="0">
<tr>
<td colspan = "2" class="normalText"><b>Δώστε το όνομα του
κεφαλαίου που θέλετε</b></td>
</tr>
<tr>
<td colspan = "2">&nbsp;</td>
</tr>
<tr>
<td width="20%" class="normalText">Όνομα κεφαλαίου :</td>
<td width="80%"><input type = "text" name = "chapterName"
size="45" class="whiteButton"/></td>
</tr>
<tr>
<td colspan = "2">&nbsp;</td>
</tr>
<tr>
<td colspan = "2" class="normalText"><b>Παρακαλώ επιλέξτε το
αρχείο από την παρακάτω λίστα και πατήστε Συνέχεια.</b></td>
</tr>
<tr>
<td colspan = "2">&nbsp;</td>
</tr>
<tr>
<td width="20%" class="normalText">Όνομα αρχείου :</td>
<td width="80%"><select name = "uploadedFileName"
class="whiteButton">
<option value=" "> </option>
<%

```

```

String[] fnames = fUtils.getPdfDirContents(dirName);
    for (int i = 0; i < fnames.length; i++) {
        String fname = fnames[i];
%>
        <option value="<%=fname%>"><%=fname%></option>

<%
    }
%>
    </select>
    </td>
</tr>
<tr>
    <td colspan = "2">&nbsp;</td>
</tr>

<tr>
    <td colspan = "2" class="normalText"><b>Εάν θέλετε να βάλετε
έναν σελιδοδείκτη σε ένα ήδη υπάρχον κεφάλαιο, παρακαλώ δώστε τον
αριθμό της σελίδας και πατήστε Συνέχεια. Ο αριθμός της σελίδας που θα
δώσετε πρέπει να είναι μεγαλύτερος ή ίσος από τον προηγούμενο
σελιδοδείκτη.</b></td>
</tr>
<tr>
    <td colspan = "2">&nbsp;</td>
</tr>

<tr>
    <td width="20%" class="normalText">Αριθμός σελίδας :</td>
    <td width="80%"><input type="text" name="bookmarkPage"
class="whiteButton"/></td>
</tr>

<tr>
    <td colspan = "2">&nbsp;</td>
</tr>
<tr>
<td colspan=2><center><input name="next" type="button"
value="Συνέχεια" class = "whiteButton"
onClick="formValidate()" /></center></td>

</tr>

</table>

</form>

</body>
</html>

```

Π.5 bottomFrame.jsp

```

<html>
<head>
<title>Folder Navigation</title>

<script language="javascript" type="text/JavaScript">

function winopen(fname,foptions){

```

```

newWindow = window.open(fname, 'dialogWindow', foptions)
//, 'width=400,height=300'
newWindow.focus()
}
function editChapter(){
    //get the values from the hidden fields in the tree.jsp
    //process them
    //and store them in hidden fields in this jsp page...
    selectedTree = parent.frames[0].tempForm.floaded.value
    parentNode = parent.frames[0].tempForm.selectednode.value
    parentNode = parentNode.slice(0,parentNode.length-1)
    //re = /\d+/gi
    tempStr = ''
    re1 = new RegExp("chapter", "g")
    re2 = new RegExp("_", "g")
    re3 = new RegExp("\\[\\]", "g")
    //alert(parentNode)
    parentNode = parentNode.replace(re1, 'chapter[')
    parentNode = parentNode.replace(re2, ']'_')
    parentNode = parentNode.replace(re2, ']'_')
    parentNode = parentNode.replace(re3, '')
    lastCharAscii = parentNode.charCodeAt(parentNode.length-1)
    //alert(lastCharAscii)
    if ((lastCharAscii>=48) && (lastCharAscii<=57)){
        parentNode = parentNode + ']'
    }
    //edo an to teleytaio psifio einai digit, tote prosthese kai
ena ]
    //gia na kleisei to bracket poy anoikse

    if (parentNode.charAt(parentNode.length-1)=='['){
        parentNode = parentNode.substr(0,parentNode.length-1)
    }
    win2openStr = 'editChapter.jsp?selectedTree=' + selectedTree
+'&parentNode=' + parentNode
    newWindow
    window.open(win2openStr, 'dialogWindow', 'width=700,height=300,status')
}

function deleteChapter(){
    //get the values from the hidden fields in the tree.jsp
    //process them
    //and store them in hidden fields in this jsp page...
    selectedTree = parent.frames[0].tempForm.floaded.value
    parentNode = parent.frames[0].tempForm.selectednode.value
    parentNode = parentNode.slice(0,parentNode.length-1)
    tempStr = ''
    re1 = new RegExp("chapter", "g")
    re2 = new RegExp("_", "g")
    re3 = new RegExp("\\[\\]", "g")
    parentNode = parentNode.replace(re1, 'chapter[')
    parentNode = parentNode.replace(re2, ']'_')
    parentNode = parentNode.replace(re2, ']'_')
    parentNode = parentNode.replace(re3, '')
    lastCharAscii = parentNode.charCodeAt(parentNode.length-1)
    //alert(lastCharAscii)
    if ((lastCharAscii>=48) && (lastCharAscii<=57)){
        parentNode = parentNode + ']'
    }
}

```



```

        <tr><td colspan="4">&nbsp;&nbsp;&nbsp;&nbsp;</td></tr>
        <tr>
            <td align="center"><input name="newChap"
value="Προσθήκη νέου Κεφαλαίου" type="button" class="whiteButton"
onClick
=
"winopen('addNewChapter.jsp','width=800,height=500,status')"/><!--a
href="javascript:winopen('addNewChapter.jsp','width=800,height=500,st
atus')">Προσθήκη νέου Κεφαλαίου</a--></td>
            <td align="center"><input name="editChap"
value="Διόρθωση Κεφαλαίου" type="button" class="whiteButton" onClick
=
"editChapter()"/><!--a href="javascript:editChapter()">Διόρθωση
Κεφαλαίου</a--></td>
            <td align="center"><input name="delChap"
value="Διαγραφή Κεφαλαίου" type="button" class="whiteButton" onClick
=
"deleteChapter()"/><!--a href="javascript:deleteChapter()">Διαγραφή
Κεφαλαίου</a--></td>
            <td align="center"><input name="helpBut"
value="Βοήθεια" type="button" class="whiteButton" onClick
=
"helpMe()"/>&nbsp;&nbsp;&nbsp;&nbsp;<input name="goBack" value="Επιστροφή"
type="button" class="whiteButton" onClick
=
"goBack()"/><!--a href="javascript:goBack()">Επιστροφή</a--></td>
        </tr>
    </table>

</body>
</html>

```

Π.6 chapterAdded.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import="gr.ntua.ece.util.XmlUtils"%>
<%
        String bookmarkPage =
request.getParameter("bookmarkPage");
        String uploadedFileName
=request.getParameter("uploadedFileName");
        String parentNode = request.getParameter("parentNode");
        String selectedTree =
request.getParameter("selectedTree");
        String chapterName = request.getParameter("chapterName");
        //System.out.println("bookmarkPage : " + bookmarkPage + "
selectedTree : " + selectedTree + " parentNode: " + parentNode + "
chapterName : " + chapterName );
        String webApPath =
pageContext.getServletContext().getRealPath("/");
        XmlUtils xmlUtils = new XmlUtils(webApPath);
        xmlUtils.addChapterNode(
selectedTree,parentNode,chapterName,uploadedFileName,bookmarkPage);

%>
<app:checkLogon/>
<html:html locale="true">
<head>

```

```

<title></title>
<script language="javascript" type="text/JavaScript">

function refreshTree(){
    opener.parent.frames[0].location      =      'tree.jsp?fname='+
opener.parent.frames[0].tempForm.floaded.value
    self.close();
}
</script>
<html>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-
7">
</head>
<body bgcolor="white" onLoad = "refreshTree()">
<br>
Το κεφάλαιο προστέθηκε με επιτυχία. Κάντε κλικ <a
href="javascript:self.close()">εδώ</a> για να επιστρέψετε.
</body>
</html>

```

Π.7 chapterEdited.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page import="gr.ntua.ece.webapp.formbeans.extractDbDataForm"%>
<%@ page contentType = "text/html; charset=iso-8859-7" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/app.tld" prefix="app" %>
<%
extractDbDataForm          test          =
(extractDbDataForm)session.getAttribute("extractDbDataForm");
String str2write = "Τα κεφάλαια <BR>";
if (test.getUpdateChapter5().equals("on")){str2write = str2write + "
<a href=\"pdfs/chap5(dynamic).pdf\" target=\"_new\">5. Προσωπικό της
Σχολής</a><br> "; };
if (test.getUpdateChapter79().equals("on")){str2write = str2write + "
<a href=\"pdfs/chap7(dynamic).pdf\" target=\"_new\">7. Πρόγραμμα
Κορμού</a><br>          <a href=\"pdfs/chap9(dynamic).pdf\"
target=\"_new\">9. Περιγραφή μαθημάτων - Πρόγραμμα Κορμού</a><br> ";
};
if (test.getUpdateChapter810().equals("on")){str2write = str2write +
" <a href=\"pdfs/chap8(dynamic).pdf\" target=\"_new\">8. Πρόγραμμα
Ροών</a><br>          <a href=\"pdfs/chap10(dynamic).pdf\"
target=\"_new\">10. Περιγραφή μαθημάτων - Πρόγραμμα Ροών</a><br> ";
};
str2write = str2write.substring(0,str2write.length()-1);
str2write = str2write + " ενημερώθηκαν με επιτυχία";
if
((!(test.getUpdateChapter5().equals("on")))&&!(test.getUpdateChapter
79().equals("on")))&&!(test.getUpdateChapter810().equals("on"))){
str2write = "";};
%>

```

```

<app:checkLogon/>

<html:html locale="true">
<head>
<title></title>

<html:base/>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-7">
</head>
<link href="stylesheets/stylessheet.css" rel="stylesheet"
type="text/css">
<body bgcolor="white" class="normalText">
<%=str2write%>

<br>
Κάντε κλίκ <html:link page="/pdfManager/mainMenu.jsp">εδώ</html:link>
για να επιστρέψετε στο αρχικό menu

</body>
</html:html>

```

Π.8 deleteChapter.jsp

```

<%@ page contentType = "text/html; charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.XmlUtils" %>
<%@ page import = "org.dom4j.Document" %>
<%@ page import = "org.dom4j.Element" %>
<%@ page import = "org.dom4j.Node" %>
<%@ page import = "org.dom4j.io.SAXReader" %>
<%@ page import = "java.io.File" %>
<%
String selectedTree = request.getParameter("selectedTree");
String parentNode = request.getParameter("parentNode");
XmlUtils xmlUtils = new
XmlUtils(pageContext.getServletContext().getRealPath("/"));
String fileName =
pageContext.getServletContext().getRealPath("/pdfManager/xmltrees");
fileName = fileName + File.separator + selectedTree;
System.out.println(fileName);
xmlUtils.deleteChapter(parentNode, fileName);
%>
<script language="javascript" type="text/JavaScript">

function refreshTree() {
    //alert('tree.jsp?fname='+
opener.parent.frames[0].tempForm.floaded.value)

    opener.parent.frames[0].location = 'tree.jsp?fname='+
opener.parent.frames[0].tempForm.floaded.value
    self.close()
}
</script>
<html>
<head>
<title>Ενημέρωση Κεφαλαίου</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-7">

```



```

</head>
<body bgcolor="white" onLoad = "refreshTree()">
<br>
Το κεφάλαιο διαγράφηκε με επιτυχία. Κάντε κλικ <a
href="javascript:self.close()">εδώ</a> για να επιστρέψετε.
</body>
</html>

```

Π.9 deletecourse.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page language="java" contentType="text/html; charset=iso-8859-7"%>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/app.tld" prefix="app" %>
<%@ page import="org.apache.log4j.Category" %>
<%@ page import = "org.dom4j.Document" %>
<%@ page import = "org.dom4j.Element" %>
<%@ page import = "org.dom4j.Node" %>
<%@ page import = "org.dom4j.io.SAXReader" %>
<%@ page import = "org.dom4j.io.XMLWriter" %>
<%@ page import = "org.dom4j.io.OutputFormat" %>
<%@ page import="java.util.List" %>
<%@ page import="java.io.File" %>
<%@ page import="java.io.FileOutputStream" %>
<%@ page import="gr.ntua.ece.util.XmlUtils"%>
<app:checkLogon/>
<html:html locale='true'>
<%
    Category cat = Category.getInstance("deletecourse.jsp");
    XmlUtils xmlUtils = new
XmlUtils(pageContext.getServletContext().getRealPath("/"));
    String strCourse = (String) request.getParameter("num");
    if (strCourse != null) {
        SAXReader reader = new SAXReader();
        Document doc = reader.read(new
java.io.File(pageContext.getServletContext().getRealPath("/pdfManager
/data") + File.separator + "chap8-10.xml"));
        Element e = (Element)
doc.selectSingleNode("//database/addinfo/row[" + strCourse + "]");

        Node detachedNode = e.detach();
        Element e2 = (Element)
doc.selectSingleNode("//database/addinfo/row[" + strCourse + "]");
        detachedNode = e2.detach();

        xmlUtils.writeXml(doc,pageContext.getServletContext().getRealPa
th("/pdfManager/data") + File.separator + "chap8-10.xml");
    } //if
    response.sendRedirect("editNonDbData.jsp");
%>
</html:html>

```

Π.10 deleteFile.jsp

```
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "java.io.File" %>
<%@ page errorHandler="errorhandler.jsp"%>
<%
String fileName = request.getParameter("name");
String fileType="";
String dirName = "";
String filePath="";
fileUtils fUtils = new fileUtils();
fileType = fileName.substring(fileName.length()-3);
if (fileType.equals("pdf")){
dirName
pageContext.getServletContext().getRealPath("/pdfManager/pdfs");
}else if (fileType.equals("doc")){
dirName
pageContext.getServletContext().getRealPath("/pdfManager/msdocs");
};
filePath = dirName + File.separator + fileName;
fUtils.deleteFile(filePath);
%>
<script language="javascript" type="text/JavaScript">

function refreshMe(){
    opener.parent.frames[0].location = 'pdfList.jsp'
    opener.parent.frames[1].location = 'docList.jsp'
    opener.parent.frames[2].location = 'pdfDocMenu.jsp'
    self.close()
}
</script>
<html>
<head>
<title>Ενημέρωση Κεφαλαίου</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-7">
</head>
<body bgcolor="white" onLoad = "refreshMe()">
</body>
</html>
```

Π.11 deleteFileOdhgos.jsp

```
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "java.io.File" %>
<%@ page errorHandler="errorhandler.jsp"%>
<%
String fileName = request.getParameter("name");
String fileType="";
String dirName = "";
String filePath="";
fileUtils fUtils = new fileUtils();
dirName
pageContext.getServletContext().getRealPath("/pdfManager/odhgoi");
filePath = dirName + File.separator + fileName;
```

```

System.out.println(filePath);
fUtils.deleteFile(filePath);
%>

<script language="javascript" type="text/JavaScript">
function refreshMe() {
    opener.parent.frames[0].location = 'showGuidesMain.jsp'
    opener.parent.frames[1].location = 'showGuidesBottom.jsp'
    self.close()
}

</script>
<html>
<head>
<title>Ενημέρωση Κεφαλαίου</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-7">
</head>
<body bgcolor="white" onLoad = "refreshMe()">
</body>

</html>

```

Π.12 deleteNodetree.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html; charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page errorPage="errorhandler.jsp"%>
<%@ page import = "java.io.File" %>
<%
fileUtils fUtils = new fileUtils();
String dirName =
pageContext.getServletContext().getRealPath("/pdfManager/xmltrees");
dirName = dirName + File.separator;
String filePath;
%>
<html>
<script language="javascript" type="text/JavaScript">
function ok2continue(what2delete) {
if (confirm("Είστε σίγουροι ότι θέλετε να διαγράψετε αυτό το αρχείο?")) {
    document.form1.filename.value = what2delete
    document.form1.submit()
}

}
function closeMe() {
what2load = 'tree.jsp'
opener.parent.frames[0].location = what2load
self.close()
}
</script>
<%

```

```

if (request.getMethod().equals( "POST" )){
String fileName = request.getParameter("filename");
String str2write="";
filePath = dirName + fileName;
try {
    if (fUtils.deleteFile(filePath)){
        str2write = "";
    } else {
        str2write=" δεν ";
    };
} catch (Exception e3) {
    // TODO Auto-generated catch block
    e3.printStackTrace();
}
%>
<body onLoad="closeMe()">
To αρχείο <%=str2write%> διαγράφηκε με επιτυχία. Πατήστε<a href="#"
onClick = "javascript:self.close()"> εδώ </a>για να επιστρέψετε
πίσω.
<%

} //do your post stuff here
else {
%>
<body class="normalText">
<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
<form name = "form1" method = "post" action = "deleteNodetree.jsp">
<input type = "hidden" name = "filename">
Παρακαλώ επιλέξτε ποιο αρχείο θέλετε να διαγράψετε.<br>
<%
String[] fnames = fUtils.getXmlDirContents(dirName);
for (int i = 0; i < fnames.length; i++) {
    String fname = fnames[i];
%>
    <a href="#"
onClick="javascript:ok2continue('<%=fname%>');return
target="leftFrame"><%=fname%></a><br>

<%
    }
}
%>
</body>
</html>

```

Π.13 deletesecretary.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page language="java" contentType="text/html; charset=iso-8859-7"%>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/app.tld" prefix="app" %>
<%@ page import="org.apache.log4j.Category" %>
<%@ page import = "org.dom4j.Document" %>
<%@ page import = "org.dom4j.Element" %>

```

```

<%@ page import = "org.dom4j.Node" %>
<%@ page import = "org.dom4j.io.SAXReader" %>
<%@ page import = "org.dom4j.io.XMLWriter" %>
<%@ page import = "org.dom4j.io.OutputFormat" %>
<%@ page import="java.util.List" %>
<%@ page import="java.io.File" %>
<%@ page import="java.io.FileOutputStream" %>
<%@ page import="gr.ntua.ece.util.XmlUtils"%>
<app:checkLogon/>
<html:html locale='true'>
<%
    XmlUtils          xmlUtils          =          new
XmlUtils(pageContext.getServletContext().getRealPath("/"));
    String strSecretary = (String) request.getParameter("num");
    if (strSecretary != null) {
        SAXReader reader = new SAXReader();
        File          file          =          new
File(pageContext.getServletContext().getRealPath("/pdfManager/data")
+ File.separator + "chap5.xml");
        Document doc = reader.read(file);
        Element          e          =          (Element)
doc.selectSingleNode("//database/addinfo/secretary_staff/row["
+ strSecretary + "]");
        Node detachedNode = e.detach();

        xmlUtils.writeXml(doc,pageContext.getServletContext().getRealPa
th("/pdfManager/data") + File.separator + "chap5.xml");
    }
    response.sendRedirect("editNonDbData.jsp");
%>
</html:html>

```

Π.14 docList.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page errorPage="errorhandler.jsp"%>
<%@ page import = "java.io.File" %>
<%
fileUtils fUtils = new fileUtils();
String          dirName          =
pageContext.getServletContext().getRealPath("/pdfManager/msdocs");
dirName = dirName + File.separator;
String filePath;
%>
<html>
<style type="text/css">
<!--
a {
    text-decoration: none;
    color: blue;
}
-->
</style>
<script language="javascript" type="text/JavaScript">

```

```

function setSelected(item, fileName) {
eval (parent.frames[2].form1.selhrefname.value + ".style.color =
'blue'")
item.style.color = 'red'
parent.frames[2].form1.selhrefname.value = 'parent.frames[1].' +
item.name
parent.frames[2].form1.selFileName.value = fileName
}

function previewMe(fileName) {
alert(fileName)
}
</script>
<link href="stylesheets/stylesheets.css" rel="stylesheet"
type="text/css">
<body class="normalText">
<a href="#" name="dummy"></a>
<table width="100%" cellspacing="0" cellpadding="0" border="0">
<tr>
<td class="normalText">Τα διαθέσιμα αρχεία DOC φαίνονται παρακάτω.
Επιλέξτε το αρχείο που θέλετε, και κατόπιν επιλέξτε από το menu την
αντίστοιχη ενέργεια που θέλετε να κάνετε</td>
</tr>
<%
String[] docnames = fUtils.getDocDirContents(dirName);
for (int i = 0; i < docnames.length; i++) {
String docname = docnames[i];
%>
<tr>
<td></td>
</tr>
<tr>
<td class="normalText"><a href="#" name="<%=doc" + i%>"
onClick = "setSelected(this, '<%=docname%>') "><%=docname%></a></TD>
</tr>
<%
}
%>
</body>
</html>

```

Π.15 editChapter.jsp

```

<%@ page contentType = "text/html; charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "gr.ntua.ece.util.XmlUtils" %>
<%@ page import = "org.dom4j.Document" %>
<%@ page import = "org.dom4j.Element" %>
<%@ page import = "org.dom4j.Node" %>
<%@ page import = "org.dom4j.io.SAXReader" %>
<%@ page import = "java.io.File" %>
<%@ page import = "java.util.List" %>
<%@ page import = "java.util.Iterator" %>
<%@ page import = "java.util.Enumeration" %>
<%@ page import = "org.dom4j.io.OutputFormat" %>
<%@ page import = "org.dom4j.io.XMLWriter" %>
<%@ page import = "java.io.PrintWriter" %>
<%@ page import = "java.io.IOException" %>
<%@ taglib uri="/WEB-INF/app.tld" prefix="app" %>

```

```

<%
String selectedTree = request.getParameter("selectedTree");
String parentNode = request.getParameter("parentNode");
String selectedFile = "";
%>
<script language="javascript" type="text/JavaScript">
function getContents() {
form1.selectedTree.value='<%=selectedTree%>'
form1.parentNode.value='<%=parentNode%>'
}
function refreshTree() {
    opener.parent.frames[0].location      =      'tree.jsp?fname='+
opener.parent.frames[0].tempForm.floaded.value
}

function closeMeRefreshTree() {
    opener.parent.frames[0].location      =      'tree.jsp?fname='+
opener.parent.frames[0].tempForm.floaded.value
    self.close()
}

function editFile(filename) {
    newWindow = window.open('msdocs/'+filename, 'previewWin')
    self.close()
}

</script>
<link href="stylesheets/stylessheet.css" rel="stylesheet"
type="text/css">
<head>
<title>Ενημέρωση Κεφαλαίου</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-
7">
</head>
<%
XmlUtils xmlUtils = new
XmlUtils(pageContext.getServletContext().getRealPath("/"));
String fileName =
pageContext.getServletContext().getRealPath("/pdfManager/xmltrees") +
File.separator + selectedTree;
SAXReader reader = new SAXReader();
Document xmlDoc=null;
try {
    xmlDoc = reader.read(new File(fileName));
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
Element rootElem = xmlDoc.getRootElement();
Element chapterElement =
(Element)rootElem.selectSingleNode(parentNode);
String chapterType = chapterElement.attributeValue("type");
if (request.getMethod().equals("POST")){
    String chapterTitleToChange =
request.getParameter("chapterName");
    chapterElement.element("chaptertitle").setText(
chapterTitleToChange );
    if (chapterType.equals("file")){
        String fileNameToChange =
request.getParameter("uploadedFileName");

```

```

        chapterElement.setAttributeValue("source", fileNameToChange);
    };
    if (chapterType.equals("bookmark")) {
        String bookmarkToChange =
request.getParameter("bookmarkPage");

        chapterElement.setAttributeValue("page", bookmarkToChange);
    };

//        the xml is saved in the hdd
xmlUtils.writeXml(xmlDoc, fileName);
%>
<body bgcolor="white" onLoad = "closeMeRefreshTree()">
<br>
To κεφάλαιο προστέθηκε με επιτυχία. Κάντε κλικ <a
href="javascript:self.close()">εδώ</a> για να επιστρέψετε.
<%
}else {
fileUtils futils = new fileUtils();
String chapterTitle = chapterElement.elementText("chaptertitle");
//System.out.println("selectedTree : " + selectedTree + " chapterType
: " + chapterType + " chapterTitle : " + chapterTitle + "
chapterElement : " + chapterElement.asXML());
%>
<body bgcolor="white" onLoad = "getContents()" class="normalText">
<form action="editChapter.jsp" method = "POST" name = "form1">
<input type= "hidden" name = "selectedTree"/>
<input type="hidden" name="parentNode"/>
<table width = "100%" cellspacing="0" cellpadding="0" border="0">
<tr>
<td colspan = "2" class="normalText"><b>Δώστε το όνομα του
κεφαλαίου που θέλετε</b></td>
</tr>
<tr>
<td colspan = "2" class="normalText">&nbsp;</td>
</tr>
<tr>
<td width="30%" class="normalText">Όνομα κεφαλαίου :</td>
<td width="70%"><input type = "text" name = "chapterName" value
="<%=chapterTitle%>" size="45" class="whiteButton"/></td>
</tr>
<tr>
<td colspan = "2">&nbsp;</td>
</tr>

<%
if (chapterType.equals("file")){ //prokeitai gia kefalaio tyroy file,
ara toy bgazo list box me files
selectedFile = chapterElement.attributeValue("source");
%>
<tr>
<td colspan = "2" class="normalText"><b>Παρακαλώ επιλέξτε το
αρχείο PDF από την παρακάτω λίστα και πατήστε Συνέχεια.</b></td>
</tr>
<tr>
<td colspan = "2" class="normalText">&nbsp;</td>
</tr>
<tr>

```



```

        <td width="30%" class="normalText">Αρχεία ήδη ανεβασμένα :</td>
        <td width="70%" class="normalText"><select name =
"uploadedFileName" class="whiteButton" >
        <%
            String dirName =
pageContext.getServletContext().getRealPath("/pdfManager/pdfs")
File.separator;
            String selected = "";
            String[] fnames = futils.getPdfDirContents(dirName);
            for (int i = 0; i < fnames.length; i++) {
                String fname = fnames[i];
                if (fname.equals(selectedFile)){
                    selected = "selected";
                }
                else {
                    selected = "";
                }
            }
        %>
        <option value="<%=fname%>" <%=selected%>><%=fname%></option>
    <%
    }
    %>
    </select>
    </td>
</tr>
<%
} //end if chapterType.equals("file")
%>
<%
if (chapterType.equals("bookmark")){ //prokeitai gia kefalαιο tyroy
bookmark, ara toy bgazo mono text box me bookmark
    String bookmarkPage = chapterElement.attributeValue("page");
    selectedFile = chapterElement.attributeValue("father");
    %>
    <tr>
        <td colspan = "2" class="normalText"><b>Παρακαλώ δώστε τον
αριθμό της σελίδας. Ο αριθμός της σελίδας που θα δώσετε πρέπει να
είναι μεγαλύτερος ή ίσος από τον προηγούμενο σελιδοδείκτη.</b></td>
    </tr>
    <tr>
        <td colspan = "2" class="normalText">&nbsp;</td>
    </tr>
    <tr>
        <td width="30%" class="normalText">Αριθμός σελίδας :</td>
        <td width="70%" class="normalText"><input type="text"
name="bookmarkPage" value="<%=bookmarkPage%>"
class="whiteButton"/></td>
    </tr>
    <%
    } //end if chapterType.equals("bookmark")
    %>
    <tr>
        <td colspan = "2">&nbsp;</td>
    </tr>
    <%
    String selectedFileDocName =
selectedFile.substring(0,selectedFile.length()-4)+".doc";

```

```

String                selectedFileDocPath                =
pageContext.getServletContext().getRealPath("/pdfManager/msdocs") +
File.separator + selectedFileDocName;
File msdocFile = new File(selectedFileDocPath);
if (msdocFile.exists()){
%>
<tr>
    <td class="normalText">Εάν θέλετε να επεξεργαστείτε το αρχείο
<%=selectedFileDocName%> κάντε κλίκ στο Επεξεργασία</td>
    <td
        class="normalText"><input
            name="butEditFile"
value="Επεξεργασία" type="button" class="whiteButton" onClick =
"editFile('<%=selectedFileDocName%>')"/></td>
</tr>

<tr>
    <td colspan = "2">&nbsp;  </td>
</tr>
<%
} //end if msdocfile.exists
%>
<tr>
<td colspan=2><center><input type="submit" name="submit"
value="Συνέχεια" class="whiteButton"></center></td>
</tr>
</table>
</form>
<%
} //end else
%>
</body>
</html>

```

Π.16 editNonDbData.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page language="java" contentType="text/html; charset=iso-8859-7"%>

<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/app.tld" prefix="app" %>
<%@ page import="org.apache.log4j.Category" %>
<%@ page import = "org.dom4j.Document" %>
<%@ page import = "org.dom4j.Element" %>
<%@ page import = "org.dom4j.Node" %>
<%@ page import = "org.dom4j.io.SAXReader" %>
<%@ page import = "org.dom4j.io.XMLWriter" %>
<%@ page import = "org.dom4j.io.OutputFormat" %>
<%@ page import = "org.dom4j.CDATA" %>
<%@ page import="java.util.List" %>
<%@ page import="java.io.File" %>
<%@ page import="java.io.FileOutputStream" %>
<%@ page import="gr.ntua.ece.util.XmlUtils"%>
<app:checkLogon/>
<html>
    <head>
        <title>

```

```

        Edit non db data
    </title>
    <meta HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=iso-8859-7">
    </head>
<link href="stylesheets/stylesheets.css" rel="stylesheet"
type="text/css">
    <script type='text/javascript'>
        function confirmDeleteSecretary(num) {
            if(confirm('Είστε σίγουροι πως θέλετε να διαγράψετε
το επελεγμένο στοιχείο?')) {
                num = num + 1

                this.location.href='deletesecretary.jsp?num='+num
            }
        }

        function confirmDeleteCourse(num) {
            if(confirm('Είστε σίγουροι πως θέλετε να διαγράψετε
το επελεγμένο στοιχείο?')) {
                num = num + 1

                this.location.href='deletecourse.jsp?num='+num
            }
        }

function helpMe() {
}
    </script>

<%
    Category cat = Category.getInstance("editnondbdata.jsp");
    String strYear = request.getParameter("year");
    String strPresident = request.getParameter("president");
    String strAssPresident = request.getParameter("asspresident");
    String strSecretary = request.getParameter("secretary");
    String strAssSecretary = request.getParameter("assecretary");
    String strDivision1 = request.getParameter("division1");
    String strDivision2 = request.getParameter("division2");
    String strDivision3 = request.getParameter("division3");
    String strDivision4 = request.getParameter("division4");
    String strDivision5 = request.getParameter("division5");
    String strDivision6 = request.getParameter("division6");
    String strDivision7 = request.getParameter("division7");
    String strNewSecretary = request.getParameter("newsecretary");
    String strNewCourseName = request.getParameter("newcoursename");
    String strNewCourseDescription = request.getParameter("newcoursedescription");
    boolean bCheck = (strYear != null) &&
        (strPresident != null) &&
        (strAssPresident != null) &&
        (strSecretary != null) &&
        (strAssSecretary != null) &&
        (strDivision1 != null) &&
        (strDivision2 != null) &&
        (strDivision3 != null) &&
        (strDivision4 != null) &&
        (strDivision5 != null) &&
        (strDivision6 != null) &&
        (strDivision7 != null);

```

```

    if (bCheck) {
        SAXReader reader = new SAXReader();
        Document doc = reader.read(new
File(pageContext.getServletContext().getRealPath("/pdfManager/data")
+ File.separator + "chap5.xml"));
        Document doc2 = reader.read(new
File(pageContext.getServletContext().getRealPath("/pdfManager/data")
+ File.separator + "chap8-10.xml"));
        out.println("president" + strPresident);
        Element nYear =
(Element) doc.selectSingleNode("//database/addinfo/year");
        nYear.setText(strYear);
        Node nPresident =
doc.selectSingleNode("//database/addinfo/president");
        nPresident.setText(strPresident);
        Node nAssPresident =
doc.selectSingleNode("//database/addinfo/asspresident");
        nAssPresident.setText(strAssPresident);
        Node nSecretary =
doc.selectSingleNode("//database/addinfo/secretary");
        nSecretary.setText(strSecretary);
        Node nAssSecretary =
doc.selectSingleNode("//database/addinfo/asssecretary");
        nAssSecretary.setText(strAssSecretary);
        Node nDivision1 =
doc.selectSingleNode("//database/addinfo/division1");
        nDivision1.setText(strDivision1);
        Node nDivision2 =
doc.selectSingleNode("//database/addinfo/division2");
        nDivision2.setText(strDivision2);
        Node nDivision3 =
doc.selectSingleNode("//database/addinfo/division3");
        nDivision3.setText(strDivision3);
        Node nDivision4 =
doc.selectSingleNode("//database/addinfo/division4");
        nDivision4.setText(strDivision4);
        Node nDivision5 =
doc.selectSingleNode("//database/addinfo/division5");
        nDivision5.setText(strDivision5);
        Node nDivision6 =
doc.selectSingleNode("//database/addinfo/division6");
        nDivision6.setText(strDivision6);
        Node nDivision7 =
doc.selectSingleNode("//database/addinfo/division7");
        nDivision7.setText(strDivision7);
        if (strNewSecretary.length() > 0) {
            Element e =
doc.selectSingleNode("//database/addinfo/secretary_staff");
            Element newE = e.addElement("row");
            newE.setText(strNewSecretary);
        }
        if ((strNewCourseName.length() > 0) &&
(strNewCourseDescription.length() > 0)) {
            Element e =
doc2.selectSingleNode("//database/addinfo");
            Element newEname = e.addElement("row");
            newEname.addAttribute("bold", "true");
            newEname.setText(strNewCourseName);
            Element newEdescription = e.addElement("row");
            newEdescription.setText(strNewCourseDescription);
        }
    }
}

```

```

        XmlUtils          xmlUtils          =          new
XmlUtils(pageContext.getServletContext().getRealPath("/"));

        xmlUtils.writeXml(doc,pageContext.getServletContext().getRealPa
th("/pdfManager/data") + File.separator + "chap5.xml");

        xmlUtils.writeXml(doc2,pageContext.getServletContext().getRealP
ath("/pdfManager/data") + File.separator + "chap8-10.xml");
        response.sendRedirect("editNonDbData.jsp");
    } else {
        SAXReader reader = new SAXReader();
        Document      doc      =      reader.read(new
java.io.File(pageContext.getServletContext().getRealPath("/pdfManager
/data") + File.separator + "chap5.xml"));
        Document      doc2     =      reader.read(new
java.io.File(pageContext.getServletContext().getRealPath("/pdfManager
/data") + File.separator + "chap8-10.xml"));
        String sYear = doc.valueOf("//database/addinfo/year");
        String      sPresident      =
doc.valueOf("//database/addinfo/president");
        String      sAssPresident   =
doc.valueOf("//database/addinfo/asspresident");
        String      sSecretary     =
doc.valueOf("//database/addinfo/secretary");
        String      sAssSecretary  =
doc.valueOf("//database/addinfo/assecretary");
        String      sDivision1     =
doc.valueOf("//database/addinfo/division1");
        String      sDivision2     =
doc.valueOf("//database/addinfo/division2");
        String      sDivision3     =
doc.valueOf("//database/addinfo/division3");
        String      sDivision4     =
doc.valueOf("//database/addinfo/division4");
        String      sDivision5     =
doc.valueOf("//database/addinfo/division5");
        String      sDivision6     =
doc.valueOf("//database/addinfo/division6");
        String      sDivision7     =
doc.valueOf("//database/addinfo/division7");
        List        lSecretaryStaff =
doc.selectNodes("//database/addinfo/secretary_staff//row");
        for (int i = 0; i < lSecretaryStaff.size(); i++) {
            String      s          =
doc.valueOf("//database/addinfo/secretary_staff//row[" + new
Integer(i).toString() + "]");
            lSecretaryStaff.set(i, s);
        }
        List        lCourses       =
doc2.selectNodes("//database/addinfo//row");
        for (int i = 0; i < lCourses.size(); i++) {
            String      str        =      ((Element)
lCourses.get(i)).valueOf(".");
            lCourses.set(i, str);
        }
    }
%>
    <body bgcolor='white' class="normalText">
        <form action='editNonDbData.jsp'
            name='editnondbdata'
            method='post'
            >

```

```

<table border='0'>
  <tr>
    <td align='left' colspan='2'>
      <br/>
    </td>
  </tr>
  <tr>
    <td align='left' width='50%'
class="normalText">
      Ακαδημαϊκό Έτος
    </td>
    <td align='left' width='50%'
class="normalText">
      <input type='text'
value='<%=sYear%>' name='year' class="whiteButton"/>
    </td>
  </tr>
  <tr>
    <td align='left' width='50%'
class="normalText">
      Πρόεδρος Σχολής
    </td>
    <td align='left' width='50%'
class="normalText">
      <input type='text'
value='<%=sPresident%>' name='president' class="whiteButton"/>
    </td>
  </tr>
  <tr>
    <td align='left' width='50%'
class="normalText">
      Αναπληρωτής πρόεδρος Σχολής
    </td>
    <td align='left' width='50%'
class="normalText">
      <input type='text'
value='<%=sAssPresident%>' name='asspresident' class="whiteButton"/>
    </td>
  </tr>
  <tr>
    <td align='left' width='50%'
class="normalText">
      Γραμματέας
    </td>
    <td align='left' width='50%'
class="normalText">
      <input type='text'
value='<%=sSecretary%>' name='secretary' class="whiteButton"/>
    </td>
  </tr>
  <tr>
    <td align='left' width='50%'
class="normalText">
      Αναπληρωτής Γραμματέας
    </td>
    <td align='left' width='50%'
class="normalText">
      <input type='text'
value='<%=sAssSecretary%>' name='asssecretary' class="whiteButton"/>
    </td>
  </tr>
</table>

```

```

        <tr>
            <td colspan='2'>
                <br/>
            </td>
        </tr>
        <tr>
            <td align='left' width='50%'
class="normalText">
                Διευθυντής τομέα
                Ηλεκτρομαγνητικών Εφαρμογών, Ηλεκτροοπτικής και Ηλεκτρονικών Υλικών
            </td>
            <td align='left' width='50%'
class="normalText">
                <input type='text'
value='<%=sDivision1%>' name='division1' class="whiteButton"/>
            </td>
        </tr>
        <tr>
            <td align='left' width='50%'
class="normalText">
                Διευθυντής τομέα Συστημάτων
                μετάδοσης Πληροφορίας και Τεχνολογίας Υλικών
            </td>
            <td align='left' width='50%'
class="normalText">
                <input type='text'
value='<%=sDivision2%>' name='division2' class="whiteButton"/>
            </td>
        </tr>
        <tr>
            <td align='left' width='50%'
class="normalText">
                Διευθυντής Τομέα Σημάτων, Ελέγχου
                και Ρομποτικής
            </td>
            <td align='left' width='50%'
class="normalText">
                <input type='text'
value='<%=sDivision3%>' name='division3' class="whiteButton"/>
            </td>
        </tr>
        <tr>
            <td align='left' width='50%'
class="normalText">
                Διευθυντής Τομέα Τεχνολογίας
                Πληροφορικής και Υπολογιστών
            </td>
            <td align='left' width='50%'
class="normalText">
                <input type='text'
value='<%=sDivision4%>' name='division4' class="whiteButton"/>
            </td>
        </tr>
        <tr>
            <td align='left' width='50%'
class="normalText">
                Διευθυντής Τομέα Επικοινωνιών,
                Ηλεκτρονικής και Συστημάτων Πληροφορικής
            </td>
            <td align='left' width='50%'>

```

```

                                <input
                                type='text'
value='<%=sDivision5%>' name='division5' class="whiteButton"/>
                                </td>
</tr>
<tr>
<td align='left' width='50%'
class="normalText">
                                Διευθυντής Τομέα Ηλεκτρικής
Ισχύος
                                </td>
                                <td align='left' width='50%'>
                                <input
                                type='text'
value='<%=sDivision6%>' name='division6' class="whiteButton"/>
                                </td>
</tr>
<tr>
<td align='left' width='50%'
class="normalText">
                                Διευθυντής Τομέα Ηλεκτρικών
Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων
                                </td>
                                <td align='left' width='50%'>
                                <input
                                type='text'
value='<%=sDivision7%>' name='division7' class="whiteButton"/>
                                </td>
</tr>
<tr>
<td colspan='2'>
<br/>
</td>
</tr>
<tr>
<th align='left' colspan='2'
class="normalText">
                                Προσωπικό γραμματείας
</th>
</tr>
<tr>
<td colspan='2'>
<br/>
</td>
</tr>
<%
    for (int i = 0; i < lSecretaryStaff.size(); i++) { %>
<tr>
<td width='50%' class="normalText">
                                <%=lSecretaryStaff.get(i)%>
</td>
<td width='50%' class="normalText">
                                <a
href='javascript:confirmDeleteSecretary(<%=i%>)'>[Delete]</a>
                                </td>
</tr>
<%
        } //for %>
<tr>
<td colspan='2'>
<br/>
</td>
</tr>
<tr>
<td align='left' width='50%'
class="normalText">

```



```

                Προσθήκη νέου μέλους
            </td>
            <td align='left' width='50%'>
                <input name='newsecretary'
type='text' class="whiteButton" maxlength='50'>
            </td>
        </tr>
        <tr>
            <td colspan='2' class="normalText">
                <br/>
            </td>
        </tr>
        <tr>
            <th align='left' colspan='2'
class="normalText">
                μαθήματα ΕΜΠ προς άλλες Σχολές
            </th>
        </tr>
        <tr>
            <td colspan='2'>
                <br/>
            </td>
        </tr>
        <%
            for (int i = 0; i < lCourses.size(); i++) { %>
            <tr>
                <td width='50%' class="normalText">
                    <%=lCourses.get(i)%>
                </td>
                <td width='50%' class="normalText">
                    <% if (i % 2 == 0) { %><a
href='javascript:confirmDeleteCourse(<%=i%>)'>[Delete]</a> <% } %>
                </td>
            </tr>
            <%
                } //for %>
            <tr>
                <td colspan='2'>
                    <br/>
                </td>
            </tr>
            <tr>
                <td colspan='2' class="normalText">
                    Προσθήκη νέου μαθήματος
                </td>
            </tr>
            <tr>
                <td align='right' width='50%'
class="normalText">
                    Όνομα
                </td>
                <td align='left' width='50%'
class="normalText">
                    <input type='text'
name='newcoursename' maxlength='100' size='60'
onfocus='javascript:this.select()' class="whiteButton">
                </td>
            </tr>
            <tr>
                <td align='right' width='50%'
class="normalText">
                    Περιγραφή
                </td>
            </tr>
        </pre>

```

```

class="normalText">
        <td align='left' width='50%'
        <input type='text'
name='newcoursedescription' maxlength='100' size='60'
onfocus='javascript:this.select()' class="whiteButton">
        </td>
    </tr>
    <tr>
        <td colspan='2'>
            <br/>
        </td>
    </tr>
    <tr>
        <td align='right' width='50%'>
            <input type='submit'
name='submit' value='Submit' class="whiteButton"/>
        </td>
        <td align='left' width='50%'>
            <input type='reset' name='reset'
value='Reset' class="whiteButton"/>&nbsp;&nbsp;&nbsp;<input name="helpBut"
value="Βοήθεια" type="button" class="whiteButton" onClick =
"helpMe()" />
        </td>
    </tr>
</table>
<hr>
<a href='extractDbData.jsp'
class="normalText">Epistrofi</a>
</form>
</body>
<% } //if %>
</html>

```

Π.17 errorhandler.jsp

```

<%@ page isErrorPage="true"%>
<%@ page contentType = "text/html;charset=windows-1253" %>
<html>
<link href="stylesheets/stylessheet.css" rel="stylesheet"
type="text/css">
<body class="normalText">
<font color="red"><b>An error has occured!</b></font><br>
<%= exception.getMessage() %><br><br>
<a href="javascript:history.back()">Επιστροφή </a> στην αρχική σελίδα

</body>
</html>

```

Π.18 extractDbData.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>

```

```

<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/app.tld" prefix="app" %>
<app:checkLogon/>
<html:html locale="true">
<head>
<title>Εξάγωση - Διαχείριση Δυναμικών Δεδομένων από την βάση</title>
<html:base/>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-7">
</head>
<link href="stylesheets/styleSheet.css" rel="stylesheet" type="text/css">
<body bgcolor="white" class="normalText">
<html:errors/>
<html:form action="extractDbDataAction">
<table width = "100%" cellspacing="0" cellpadding="0" border="0" class="normalText">
<tr>
<td colspan = "2"><b>Παρακαλώ επιλέξτε τα κεφάλαια του οδηγού σπουδών που θέλετε να ενημερώσετε. Για την ενημέρωση των δεδομένων που δεν περιέχονται στην βάση κάντε κλικ στην τελευταία επιλογή</b></td>
</tr>
<tr>
<td width="20%"><html:checkbox property="updateChapter5"/></td>
<td width="80%">Ενημέρωση δεδομένων κεφαλαίου <a href="pdfs/chap5(dynamic).pdf" target="_new">5. Προσωπικό της Σχολής</a></td>
</tr>
<tr>
<td width="20%"><html:checkbox property="updateChapter79"/></td>
<td width="80%">Ενημέρωση δεδομένων Κεφαλαίων <a href="pdfs/chap7(dynamic).pdf" target="_new">7. Πρόγραμμα Κορμού</a> και <a href="pdfs/chap9(dynamic).pdf" target="_new">9. Περιγραφή μαθημάτων - Πρόγραμμα Κορμού</a> </td>
</tr>
<tr>
<td width="20%"><html:checkbox property="updateChapter810"/></td>
<td width="80%">Ενημέρωση δεδομένων Κεφαλαίων <a href="pdfs/chap8(dynamic).pdf" target="_new">8. Πρόγραμμα Ροών</a> και <a href="pdfs/chap10(dynamic).pdf" target="_new">10. Περιγραφή μαθημάτων - Πρόγραμμα Ροών</a> </td>
</tr>
<tr>
<td width="20%"><html:checkbox property="updateNonDbData"/></td>
<td width="80%">Ενημέρωση δεδομένων που δεν περιέχονται στην Βάση Δεδομένων</td>
</tr>
<tr>
<td colspan=2><input type="submit" value="Συνέχεια" class="whiteButton"/></td>
</tr>
<tr>
<td colspan=2>&nbsp;</td>
</tr>

```

```

<tr>
<td colspan=2>Κάντε κλικ <html:link
page="/pdfManager/mainMenu.jsp">εδώ</html:link> για να επιστρέψετε
στο αρχικό menu</td>

</tr>
</table>
</html:form>
</body>
</html:html>

```

Π.19 extractPdf.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html; charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "java.io.File" %>
<%
fileUtils futils = new fileUtils();
%>
<html>
<script language="javascript" type="text/JavaScript">
function setInitValues() {
    selectedTree = opener.parent.frames[0].tempForm.floded.value
    form1.selectedTree.value=selectedTree
}

function formValidate() {
with (form1){
if (pdffilename.value==''){
    alert('Παρακαλώ δώστε όνομα Αρχείου PDF')
}else {
    submit()
}
} //end with
}

function helpMe() {

}

</script>

<head>
<title>Εξαγωγή PDF</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-7">
</head>
<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
<body bgcolor="white" onLoad = "setInitValues()">
<form action="extractPdfAction.jsp" method = "POST" name = "form1">
<input type= "hidden" name = "selectedTree"/>

```

```

<table width = "100%" cellspacing="0" cellpadding="0" border="0"
class="normalText">
<tr>
<td colspan=4><b>Σχόλια στην κεφαλίδα - υποσέλιδο και αρίθμηση
σελίδων</b></td>
</tr>
<tr>
<td width="25%">Σχόλια κεφαλίδας</td>
<td width="25%"><input type = "text" name="HeaderComments"
maxlength="45" class="whiteButton"/></td>
<td width="25%">Αρίθμηση σελίδας στην κεφαλίδα</td>
<td width="25%"><input type = "checkbox" name="HeaderPageCount"
value="on" class="whiteCheck"/></td>
</tr>
<tr>
<td colspan = "4">&nbsp;</td>
</tr>
<tr>
<td width="25%">Σχόλια υποσέλιδου</td>
<td width="25%"><input type = "text" name="FooterComments"
maxlength="45" class="whiteButton"/></td>
<td width="25%">Αρίθμηση σελίδας στο υποσέλιδο</td>
<td width="25%"><input name="FooterPageCount" type = "checkbox"
class="whiteCheck" value="on"/></td>
</tr>
<tr>
<td colspan = "4">&nbsp;</td>
</tr>
<tr>
<td width="25%">Επελεγμένα γλώσσα περιεχομένων</td>
<td width="25%"><select name="contentLang" class="whiteButton">
<option selected value="gr">Ελληνικά</option>
<option value="en">Αγγλικά</option>
</select>
</td>
<td width="25%">Δημιουργία περιεχομένων</td>
<td width="25%"><input name="createContents" type = "checkbox"
class="whiteCheck" value="on" checked /></td>
</tr>
<tr>
<td>Προσθήκη εξώφυλλου</td>
<td>
<select name = "cover2Upload" class="whiteButton" >
<option value="" selected></option>
<%
String dirName
pageContext.getServletContext().getRealPath("/pdfManager/pdfs")
File.separator;
String selected = "";
String[] fnames = futils.getPdfDirContents(dirName);
for (int i = 0; i < fnames.length; i++) {
String fname = fnames[i];
%>
<option value="<%=fname%>"><%=fname%></option>
<%
}
%>
</select>

```

```

</td>
<td></td>
<td></td>
</tr>
<tr>
  <td colspan="4">
    <table width="100%" cellspacing="0" cellpadding="0" border=0
class="normalText">
      <tr>
        <td colspan="4">&nbsp;</td>
      </tr>
      <tr>
        <td colspan="4"><b>Εμφάνιση σελίδων</b></td>
      </tr>
      <tr>
        <td width="25%"><input name="pageLayout"
type="radio" class="whiteCheck" value = "single"
checked>&nbsp;&nbsp;&nbsp;Μία σελίδα</td>
        <td width="25%"><input name="pageLayout"
type="radio" class="whiteCheck" value = "one">&nbsp;&nbsp;&nbsp;Μία
σιτήλη</td>
        <td width="25%"><input name="pageLayout"
type="radio" class="whiteCheck" value = "twoleft">&nbsp;&nbsp;&nbsp;Δύο
σιτήλες, σελίδες αριστερά</td>
        <td width="25%"><input name="pageLayout"
type="radio" class="whiteCheck" value = "tworight">&nbsp;&nbsp;&nbsp;Δύο
σιτήλες, σελίδες δεξιά</td>
      </tr>
      <tr>
        <td colspan="4">&nbsp;</td>
      </tr>
      <tr>
        <td colspan="4"><b>Τρόπος προβολής
εγγράφου</b></td>
      </tr>
      <tr>
        <td width="25%"><input name="pageMode" type="radio"
class="whiteCheck" value = "none" checked>&nbsp;&nbsp;&nbsp;Κανένας</td>
        <td width="25%"><input name="pageMode" type="radio"
class="whiteCheck" value = "outline">&nbsp;&nbsp;&nbsp;Προβολή χάρτη
εγγράφου</td>
        <td width="25%"><input name="pageMode" type="radio"
class="whiteCheck" value = "thumb">&nbsp;&nbsp;&nbsp;Προβολή εικόνων
thumbnail</td>
        <td width="25%"><input name="pageMode" type="radio"
class="whiteCheck" value = "fullscreen">&nbsp;&nbsp;&nbsp;Προβολή πλήρους
οθόνης</td>
      </tr>
      <tr>
        <td colspan="4">&nbsp;</td>
      </tr>
      <tr>
        <td width="50%" colspan="2"><input
name="hideToolBar" type = "checkbox" class="whiteCheck"
value="on"/>&nbsp;&nbsp;&nbsp;Απόκρυψη εργαλειοθήκης</td>
        <td width="50%" colspan="2"><input
name="hideMenuBar" type = "checkbox" class="whiteCheck"
value="on"/>&nbsp;&nbsp;&nbsp;Απόκρυψη μενού</td>
      </tr>
    </table>
  </td>

```

```

        </td>
</tr>
<tr>
    <td colspan="4">
        <table width="100%" cellpadding="0" cellspacing="0" border="0"
class="normalText">
            <tr>
                <td colspan="4">&nbsp;</td>
            </tr>
            <tr>
                <td colspan="4"><b>Κωδικοποίηση</b></td>
            </tr>
            <tr>
                <td width="25%"><input name="allowCopy" type =
"checkbox" class="whiteCheck" value="on"/>&nbsp;&nbsp;&nbsp;Επιτρεπτή
αντιγραφή</td>
                <td width="25%"><input name="allowModContents" type
= "checkbox" class="whiteCheck" value="on"/>&nbsp;&nbsp;&nbsp;Επιτρεπτή
αλλαγή περιεχομένων</td>
                <td width="25%"><input name="allowPrint" type =
"checkbox" class="whiteCheck" value="on"/>&nbsp;&nbsp;&nbsp;Επιτρεπτή
εκτύπωση</td>
                <td width="25%"><input name="allowModAnnots" type =
"checkbox" class="whiteCheck" value="on"/>&nbsp;&nbsp;&nbsp;Επιτρεπτή αλλαγή
σχολίων</td>
            </tr>
            <tr>
                <td width="25%"><input name="allowFillIn" type =
"checkbox" class="whiteCheck" value="on"/>&nbsp;&nbsp;&nbsp;Επιτρεπτή
αντικατάσταση</td>
                <td width="25%"><input name="allowScreenReader"
type = "checkbox" class="whiteCheck"
value="on"/>&nbsp;&nbsp;&nbsp;Επιτρεπτός αναγνώστης οθόνης</td>
                <td width="25%"><input name="allowAssembly" type =
"checkbox" class="whiteCheck" value="on"/>&nbsp;&nbsp;&nbsp;Επιτρεπτή
αλλαγή αρχείου</td>
                <td width="25%"><input name="setPasswds" type =
"checkbox" class="whiteCheck" value="on"/>&nbsp;&nbsp;&nbsp;Προσθήκη
κωδικών</td>
            </tr>
            <tr>
                <td colspan="4">&nbsp;</td>
            </tr>
            <tr>
                <td width="25%">Κωδικός χρήστη</td>
                <td width="25%"><input type="password" name =
"userpwd" class="whiteButton"/></td>
                <td width="25%">Κωδικός κατόχου</td>
                <td width="25%"><input type="password" name =
"ownerpwd" class="whiteButton"/></td>
            </tr>
            <tr>
                <td colspan="4">&nbsp;</td>
            </tr>
            <tr>
                <td colspan="4">
                    <table width="100%" cellpadding="0" border="0" class="normalText">
                        <tr>

```

```

        colspan="6"><b>Περίληψη</b></td>
    </tr>
    <tr>
        <td>Τίτλος</td>
        <td><input name="sumTitle" type =
"text" class="whiteButton" maxlength="45"/></td>
        <td>Θέμα</td>
        <td><input name="sumSubj" type =
"text" class="whiteButton" maxlength="45"/></td>
        <td>Συγγραφέας</td>
        <td><input name="sumAuth" type =
"text" class="whiteButton" maxlength="45"/></td>
    </tr>
    <tr>
        <td>Δημιουργός</td>
        <td><input type = "text"
name="sumCreat" maxlength="45" class="whiteButton"/></td>
        <td>Λέξεις κλειδιά</td>
        <td colspan="3"><input type =
"text" name="sumKey" maxlength="45" class="whiteButton"/></td>
    </tr>
</table>
</td>
</tr>

</table>
</td>
</tr>

<tr>
    <td colspan="4">&nbsp;</td>
</tr>
<tr>
    <td colspan="2">Δώστε όνομα αρχείου PDF για δημιουργία
:</td>
    <td colspan="2"><input type = "text" name="pdffilename"
maxlength="45" class="whiteButton"/></td>
</tr>
<tr>
    <td colspan="4">&nbsp;</td>
</tr>
<tr>
    <td colspan="3"><center><input name="next" type="button"
value="Συνέχεια" class = "whiteButton"
onClick="formValidate() "/></center></td>
    <td><center><input name="next" type="button"
value="Βοήθεια" class = "whiteButton"
onClick="helpMe() "/></center></td>
</tr>

</table>
</form>
</body>
</html>

```


П.20 extractPdfAction.jsp

```
<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import="java.io.File" %>
<%@ page import="java.io.FileInputStream" %>
<%@ page import="java.io.FileOutputStream" %>
<%@ page import="java.util.Enumeration" %>
<%@ page import="gr.ntua.ece.util.NavigateXmlUtils"%>
<%@ page import = "java.util.List"%>
<%@ page import = "java.util.Iterator" %>
<%@ page import="gr.ntua.ece.util.Concat"%>
<%
        String                selectedTree                =
request.getParameter("selectedTree");
        String                HeaderComments                =
request.getParameter("HeaderComments");
        boolean                HeaderPageCount                =
!(request.getParameter("HeaderPageCount")==null);
        String                FooterComments
=request.getParameter("FooterComments");
        boolean                FooterPageCount                =
!(request.getParameter("FooterPageCount")==null);
        String                contentLang                =
request.getParameter("contentLang");
        boolean                createContents                =
!(request.getParameter("createContents")==null);
        String                pageLayout = request.getParameter("pageLayout");
        String                pageMode = request.getParameter("pageMode");
        boolean                hideToolbar                =
!(request.getParameter("hideToolbar")==null);
        boolean                hideMenuBar                =
!(request.getParameter("hideMenuBar")==null);
        boolean                allowCopy                =
!(request.getParameter("allowCopy")==null);
        boolean                allowModContents                =
!(request.getParameter("allowModContents")==null);
        boolean                allowPrint                =
!(request.getParameter("allowPrint")==null);
        boolean                allowModAnnots                =
!(request.getParameter("allowModAnnots")==null);
        boolean                allowFillIn                =
!(request.getParameter("allowFillIn")==null);
        boolean                allowScreenReader                =
!(request.getParameter("allowScreenReader")==null);
        boolean                allowAssembly                =
!(request.getParameter("allowAssembly")==null);
        boolean                setPasswds                =
!(request.getParameter("setPasswds")==null);
        String                userpwd = request.getParameter("userpwd");
        String                ownerpwd = request.getParameter("ownerpwd");
        String                sumTitle = request.getParameter("sumTitle");
        String                sumSubj = request.getParameter("sumSubj");
        String                sumAuth = request.getParameter("sumAuth");
        String                sumCreat = request.getParameter("sumCreat");
        String                sumKey = request.getParameter("sumKey");
        String                pdffilename = request.getParameter("pdffilename");

```

```

        if (!pdffilename.substring(pdffilename.length() -
4).equals(".pdf")) {
            pdffilename = pdffilename + ".pdf";
        }
        String coverFilePath="";
        if (!(request.getParameter("cover2Upload").equals("")) ) {

            coverFilePath
pageContext.getServletContext().getRealPath("/pdfManager/pdfs")
File.separator + request.getParameter("cover2Upload");
        }
        String webApPath
pageContext.getServletContext().getRealPath("/pdfManager/xmltrees");
        String xmlTreePath = webApPath + File.separator +
selectedTree;
        NavigateXmlUtils navigateXmlUtils = new
NavigateXmlUtils(xmlTreePath,pageContext.getServletContext().getRealP
ath("/"));
        List dataNeeded = navigateXmlUtils.getDataNeeded();

        String args[] = (String[])dataNeeded.get(0);
        String title[] = (String[])dataNeeded.get(1);
        int pagenum[] = (int[])dataNeeded.get(2);
        boolean untitled[] = (boolean[])dataNeeded.get(3);
        int level[] = (int[])dataNeeded.get(4);
        args[0]
pageContext.getServletContext().getRealPath("/pdfManager/odhgoi")
File.separator + pdffilename;
        //Construct a Concat class
        Concat con=new
Concat(args,pagenum,title,untitled,level,pageContext.getServletContext
().getRealPath("/"));
        //Send parametres and information to Concat class
        con.setHeader(HeaderComments,HeaderPageCount);
        con.setFooter(FooterComments,FooterPageCount);
        con.setSummary(sumTitle,sumSubj,sumAuth,sumCreat,sumKey);
        con.setAllowPrinting(allowPrint);
        con.setAllowCopying(allowCopy);
        con.setAllowAssembly(allowAssembly);
        con.setAllowFillIn(allowFillIn);
        con.setAllowModifyAnnotations(allowModAnnots);
        con.setAllowModifyContents(allowModContents);
        con.setAllowScreenReaders(allowScreenReader);
        con.setHideToolbar(hideToolbar);
        con.setHideMenuebar(hideMenuBar);
        con.setPageMode(pageMode);
        con.setPageLayout(pageLayout);
        con.setCreateContents(createContents);
        if (contentLang.equals("gr")) con.setLanguage(true);
        else if (contentLang.equals("en")) con.setLanguage(false);
        if (setPasswds)
        {
            con.setOwnerPassword(ownerpwd,true);
            con.setUserPassword(userpwd,true);
        }
        else
        {
            con.setOwnerPassword(null,false);
            con.setUserPassword(null,false);
        }
        con.setCover(coverFilePath);

```

```

        con.Create();
for (int i = 0; i < args.length; i++) {
    System.out.println("args[" + i + "] : " + args[i]);
}
for (int i = 0; i < title.length; i++) {
    System.out.println("title[" + i + "] : " + title[i]);
}

    for (int i = 0; i < pagenum.length; i++) {
        System.out.println("page[" + i + "] : " + pagenum[i]);
    }
    for (int i = 0; i < level.length; i++) {
        System.out.println("level[" + i + "] : " + level[i]);
    }
for (int i = 0; i < untitled.length; i++) {
    System.out.println("untitled[" + i + "] : " + untitled[i]);
}
%>
<app:checkLogon/>
<html:html locale="true">
<head>
<title></title>
<script language="javascript" type="text/JavaScript">

function refreshTree(){
    opener.parent.frames[0].location      =      'tree.jsp?fname='+
opener.parent.frames[0].tempForm.floaded.value
}
</script>
<html>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-7">
</head>
<link      href="stylesheets/stylesheet.css"      rel="stylesheet"
type="text/css">
<body bgcolor="white" onLoad = "refreshTree()" class="normalText">
<br>
To PDF αρχείο δημιουργήθηκε με επιτυχία. Κάντε κλίκ <a
href="<%= "odhgoi/" + pdffilename%>" target="_New">εδώ</a> για να το
ανοίξετε.<br><br>
<a href="javascript:self.close()">Επιστροφή</a>
</body>
</html>

```

Π.21 loadNodetree.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html; charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "java.io.File" %>
<%
fileUtils fUtils = new fileUtils();

```

```

String                               dirName                               =
pageContext.getServletContext().getRealPath("/pdfManager/xmltrees") +
File.separator;
//dirName = dirName + "pdfManager\\xmltrees\\";
String[] fnames = fUtils.getXmlDirContents(dirName);

%>
<html>
<script language="javascript" type="text/JavaScript">
function ok2continue(what2load){
if (confirm("Είστε σίγουροι ότι θέλετε να επεξεργαστείτε αυτό το
αρχείο?")){
    //top.frames(1).location = what2load
    what2load = 'tree.jsp?fname=' + what2load
    opener.parent.frames[0].location = what2load
    self.close()

    }
}
</script>
<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
<body class="normalText">
Τα διαθέσιμα αρχεία φαίνονται παρακάτω. Παρακαλώ επιλέξτε ένα για
επεξεργασία.<br>
<%
    for (int i = 0; i < fnames.length; i++) {
        String fname = fnames[i];
%>
        <a href="#"
onClick="javascript:ok2continue('<%=fname%>');return
target="leftFrame"><%=fname%></a><br>
<%
    }
%>

</body>
</html>

```

Π.22 mainMenu.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html; charset=iso-8859-7" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ taglib uri="/WEB-INF/app.tld" prefix="app" %>
<app:checkLogon/>
<html:html locale="true">
<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
<head>
<title>Διαχείριση Οδηγού Σπουδών</title>
<html:base/>

```

```

<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-7">
</head>
<body bgcolor="white">
<ul class="normalText">
<li><html:link page="/pdfManager/extractDbData.jsp">Διαχείριση
Δυναμικού περιεχομένου</html:link></li>
<li><html:link page="/pdfManager/fileManager.jsp"
target="_NewWin2">Διαχείριση στατικού περιεχομένου</html:link></li>
<li><html:link page="/pdfManager/guideEditor.jsp"
target="_NewWin">Σύνθεση τελικού Οδηγού Σπουδών</html:link></li>
<li><html:link page="/pdfManager/showGuides.jsp"
target="_NewWin">Προβολή διαθέσιμων οδηγών Σπουδών</html:link></li>
</ul>
</body>
</html:html>

```

Π.23 newFile.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html; charset=iso-8859-7" %>
<link href="stylesheets/stylesheets.css" rel="stylesheet"
type="text/css">
<body bgcolor="white" class="normalText">
<form action="newFileAction.jsp" method = "POST"
enctype="multipart/form-data" name = "form1" >
<table width = "100%" cellpadding="0" cellspacing="0" border="0">
<tr>
<td colspan = "2" class="normalText"><b>Παρακαλώ επιλέξτε το
όνομα του αρχείου που υπάρχει τοπικά στον υπολογιστή σας και πατήστε
συνέχεια</b></td>
</tr>
<tr>
<td colspan = "2" class="normalText">&nbsp;</td>
</tr>
<tr>
<td width="30%" class="normalText">Όνομα αρχείου :</td>
<td width="70%"><input type = "file" name = "editedFile"
class="whiteButton"/></td>
</tr>
<tr>
<td colspan = "2">&nbsp;</td>
</tr>
<tr>
<td colspan=2><center><input type="submit" name="submit"
value="Συνέχεια" class="whiteButton"></center></td>
</tr>
</table>
</form>
</body>
</html>

```

П.24 newFileAction.jsp

```
<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "java.io.File" %>
<%@ page import="java.io.FileInputStream" %>
<%@ page import="java.io.FileOutputStream" %>
<%@ page import="java.util.Enumeration" %>
<%@ page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page import="com.oreilly.servlet.multipart.Part"%>
<%@ page import="com.oreilly.servlet.ParameterParser"%>
<%
String fileType="";
String dirName = "";
String filePath="";
String newName = "";
String fileName = "";
//File size is 10MB
MultipartRequest multi =
    new MultipartRequest(request,
request.getRealPath("tmp"), 10000 * 1024,
    "iso-8859-7", new
com.oreilly.servlet.multipart.DefaultFileRenamePolicy());
Enumeration files = multi.getFileNames();
while (files.hasMoreElements()) {
    String fname = (String)files.nextElement();
    String strGotFile =
multi.getFilesystemName(fname);
    fileName = multi.getOriginalFileName(fname);
    fileType =
fileName.substring(fileName.length()-3);
    if (fileType.equals("pdf")){
dirName =
pageContext.getServletContext().getRealPath("/pdfManager/pdfs");
    }else if (fileType.equals("doc")){
dirName =
pageContext.getServletContext().getRealPath("/pdfManager/msdocs");
    };
    filePath = dirName + File.separator +
fileName;
    File fInput = multi.getFile(fname);
    if (fInput != null) {
        File fTarget = new File(filePath);
        FileInputStream fis = new
FileInputStream(fInput);
        FileOutputStream fos = new
FileOutputStream(fTarget);
        byte[] buf = new
byte[8192];
        while (true) {
            int count =
fis.read(buf, 0, 8192);
            if (count == -
1) break;
            fos.write(buf,
0, count);
        } //while
    }
}
```

```

        fis.close();
        fos.close();
        fInput.delete();
    };
};

%>
<script language="javascript" type="text/JavaScript">
function refreshMe() {
    opener.parent.frames[0].location = 'pdfList.jsp'
    opener.parent.frames[1].location = 'docList.jsp'
    opener.parent.frames[2].location = 'pdfDocMenu.jsp'
    self.close()
}
</script>
<link href="stylesheets/stylesheets.css" rel="stylesheet"
type="text/css">
<head>
<title>Ενημέρωση Κεφαλαίου</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-
7">
</head>
<body bgcolor="white" onLoad = "refreshMe()">
</body>
</html>

```

Π.25 newNodeTree.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page errorPage="errorhandler.jsp"%>
<%@ page contentType = "text/html; charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "java.io.File" %>
<%
fileUtils fUtils = new fileUtils();
String dirName =
pageContext.getServletContext().getRealPath("/pdfManager/xmltrees") +
File.separator;
String filePath;
%>
<html>
<script language="javascript" type="text/JavaScript">

function ok2continue(what2load) {
if (confirm("Είστε σίγουροι ότι θέλετε να επεξεργαστείτε αυτό το
αρχείο?")){
    what2load = 'tree.jsp?fname=' + what2load
    alert(what2load);
    opener.parent.frames[0].location = what2load
    self.close()
}
}

</script>

```

```

<link href="stylesheets/stylesheet.css" rel="stylesheet"
type="text/css">
<%
if (request.getMethod().equals( "POST" )){
String fileName = request.getParameter("filename") ;
if (!fileName.substring(fileName.length()-4).equals(".xml")){
    fileName = fileName + ".xml";
}
filePath = dirName + fileName;
try {
    fUtils.createEmptyXmlFile(filePath);
} catch (Exception e3) {
    // TODO Auto-generated catch block
    e3.printStackTrace();
}
%>
<body class="normalText">
To αρχείο δημιουργήθηκε με επιτυχία. Πατήστε<a
href="tree.jsp?fname=<%=fileName%>" target = "leftFrame" onClick =
"javascript:self.close()"> εδώ </a>για να το επεξεργαστείτε.
<%
} //do your post stuff here

else {
%>
<body class="normalText">
<form name="form1" method="POST" action = "newNodetree.jsp">
Παρακαλώ δώστε το όνομα του οδηγού που θέλετε να δημιουργήσετε και
πατήστε συνέχεια. <br><br>
<input name = "filename" type = "textbox" class="whiteButton" /><br>
<input name = "Συνέχεια" type = "Submit" value = "Συνέχεια"
class="whiteButton">
</form>
<%
}
%>
</body>
</html>

```

Π.26 pdfDocMenu.jsp

```

<html>
<head>
<title>Folder Navigation</title>
<script language="javascript" type="text/JavaScript">
function winopen(fname,foptions){
    newWindow = window.open(fname,'dialogWindow',foptions)
    newWindow.focus()
}
function goBack(){
    if (confirm('Είστε σίγουροι πως θέλετε να επιστρέψετε στο
αρχικό menu?')){
        parent.close()
    }
}
}

```



```

function delFile(){
    file2del = form1.selFileName.value
    if (file2del==''){
        alert('Παρακαλώ επιλέξτε αρχείο για διαγραφή')
        return false
    }
    if (confirm('Είστε σίγουροι πως θέλετε να διαγράψετε το αρχείο
' + file2del + '?')){
        newWindow = window.open('deleteFile.jsp?name=' +
file2del,'dialogWindow','width=100,height=100')
    }
}
function editFile(){
    file2open = form1.selFileName.value
    if (file2open==''){
        alert('Παρακαλώ επιλέξτε αρχείο')
        return false
    }
    fileType = file2open.substr(file2open.length-3)
    filePath=''
    if (fileType=='pdf'){
        filePath = 'pdfs/' + file2open
    }else if (fileType=='doc'){
        filePath = 'msdocs/' + file2open
    }
    newWindow = window.open(filePath,'previewWin')
}

function renFile(){
    file2rep = form1.selFileName.value
    if (file2rep==''){
        alert('Παρακαλώ επιλέξτε αρχείο')
        return false
    }
    newWindow = window.open('renameFile.jsp?name=' +
file2rep,'dialogWindow','width=400,height=300')
}

function repFile(){
    file2rep = form1.selFileName.value
    if (file2rep==''){
        alert('Παρακαλώ επιλέξτε αρχείο')
        return false
    }
    newWindow = window.open('replaceFile.jsp?name=' +
file2rep,'dialogWindow','width=400,height=300')
}

function newFile(){
    newWindow =
window.open('newFile.jsp','dialogWindow','width=400,height=300')
}

function helpMe(){
}
</script>

<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
</head>

```

```

<body      topmargin="15"      leftmargin="15"      marginwidth="15"
marginheight="11" bgcolor="white" alink="black" link="black">
<form name="form1" method="POST">
<input type="hidden" name="selFileName">
<input type="hidden" name="selhrefname" value="dummy">
<table width="100%" cellspacing="0" cellpadding="0"
border="0">
      <tr>
            <td align="center"><input name="butNewFile"
value="Προσθήκη" type="button" class="whiteButton" onClick =
"newFile()" /></td>
            <td align="center"><input name="butRepFile"
value="Αντικατάσταση" type="button" class="whiteButton" onClick =
"repFile()" /></td>
            <td align="center"><input name="butEditFile"
value="Επεξεργασία" type="button" class="whiteButton" onClick =
"editFile()" /></td>
            <td align="center"><input name="butRenFile"
value="Μετονομασία" type="button" class="whiteButton" onClick =
"renFile()" /></td>
            <td align="center"><input name="butDelfile"
value="Διαγραφή" type="button" class="whiteButton" onClick =
"delFile()" /></td>
            <td align="center"><input name="Back"
value="Επιστροφή" type="button" class="whiteButton" onClick =
"goBack()" /></td>
            <td align="center"><input name="helpBut"
value="Βοήθεια" type="button" class="whiteButton" onClick =
"helpMe()" /></td>
      </tr>
    </table>
</form>
</body>
</html>

```

Π.27 pdfList.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page errorPage="errorhandler.jsp"%>
<%@ page import = "java.io.File" %>
<%
fileUtils fUtils = new fileUtils();
String      dirName
pageContext.getServletContext().getRealPath("/pdfManager/pdfs");
dirName = dirName + File.separator;
String filePath;
%>
<html>
<style type="text/css">
<!--
a {

```

```

        text-decoration: none;
        color: blue;
    }
-->
</style>
<script language="javascript" type="text/JavaScript">
function setSelected(item, fileName) {
eval (parent.frames[2].form1.selhrefname.value + ".style.color =
'blue'")
item.style.color = 'red'
parent.frames[2].form1.selhrefname.value = 'parent.frames[0].' +
item.name
parent.frames[2].form1.selFileName.value = fileName
}

function previewMe(fileName) {
alert(fileName)
}

</script>
<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
<body class="normalText">
<a href="#" name="dummy"></a>
<table width="100%" cellspacing="0" cellpadding="0" border="0">
<tr>
<td class="normalText">Τα διαθέσιμα αρχεία PDF φαίνονται παρακάτω.
Επιλέξτε το αρχείο που θέλετε, και κατόπιν επιλέξτε από το menu την
αντίστοιχη ενέργεια που θέλετε να κάνετε</td>
</tr>
<%
String[] pdfFnames = fUtils.getPdfDirContents(dirName);
for (int i = 0; i < pdfFnames.length; i++) {
String fname = pdfFnames[i];
%>
<tr>
<td></td>
</tr>
<tr>
<td class="normalText"><a href="#" name="<%= "pdf" + i %>"
onClick = "setSelected(this, '<%=fname%>')" "><%=fname%></a></TD>
</tr>
<%
}
%>
</body>
</html>

```

Π.28 renameFile.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html; charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "java.io.File" %>
<%@ page errorHandler="errorhandler.jsp"%>

```

```

<%
String fileName = request.getParameter("name");
String fileType="";
String dirName = "";
String filePath="";
String newName = "";
fileUtils fUtils = new fileUtils();
fileType = fileName.substring(fileName.length()-3);
if (fileType.equals("pdf")){
dirName
pageContext.getServletContext().getRealPath("/pdfManager/pdfs");
}else if (fileType.equals("doc")){
dirName
pageContext.getServletContext().getRealPath("/pdfManager/msdocs");
};
filePath = dirName + File.separator + fileName;
%>
<script language="javascript" type="text/JavaScript">
function refreshMe(){
    opener.parent.frames[0].location = 'pdfList.jsp'
    opener.parent.frames[1].location = 'docList.jsp'
    opener.parent.frames[2].location = 'pdfDocMenu.jsp'
    self.close()
}
</script>
<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
<head>
<title>Ενημέρωση Κεφαλαίου</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-7">
</head>
<%
if (!(request.getParameter("newName")==null)){
fUtils.renameFile(filePath,dirName + File.separator+
request.getParameter("newName"));
%>
<body bgcolor="white" onLoad = "refreshMe()">
<%
}else {
fileUtils futils = new fileUtils();
%>
<body bgcolor="white" class="normalText">
<form action="renameFile.jsp" method = "POST" name = "form1">
<input type= "hidden" name = "name" value="<%=fileName%>"/>
<table width = "100%" cellpadding="0" cellspacing="0" border="0">
<tr>
<td colspan = "2" class="normalText"><b>Παρακαλώ αλλάξτε το
όνομα του αρχείου και πατήστε συνέχεια</b></td>
</tr>
<tr>
<td colspan = "2" class="normalText">&nbsp;</td>
</tr>
<tr>
<td width="30%" class="normalText">Όνομα αρχείου :</td>
<td width="70%"><input type = "text" name = "newName" value
="<%=fileName%>" size="45" class="whiteButton"/></td>
</tr>
<tr>
<td colspan = "2">&nbsp;</td>
</tr>

```

```

<tr>
<td colspan=2><center><input type="submit" name="submit"
value="Συνέχεια" class="whiteButton"></center></td>
</tr>
</table>
</form>
<%
} //end else
%>
</body>
</html>

```

Π.29 renameFileOdhgos.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "java.io.File" %>
<%@ page errorPage="errorhandler.jsp"%>
<%
String fileName = request.getParameter("name");
String fileType="";
String dirName = "";
String filePath="";
String newName = "";
fileUtils fUtils = new fileUtils();
dirName =
pageContext.getServletContext().getRealPath("/pdfManager/odhgoi");
filePath = dirName + File.separator + fileName;
%>
<script language="javascript" type="text/JavaScript">
function refreshMe() {
    opener.parent.frames[0].location = 'showGuidesMain.jsp'
    opener.parent.frames[1].location = 'showGuidesBottom.jsp'
    self.close()
}
</script>
<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
<head>
<title>Ενημέρωση Κεφαλαίου</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-
7">
</head>
<%
if (!(request.getParameter("newName")==null)) {
fUtils.renameFile(filePath, dirName + File.separator+
request.getParameter("newName"));
%>
<body bgcolor="white" onLoad = "refreshMe()">
<%
} else {
fileUtils futils = new fileUtils();

```

```

%>
<body bgcolor="white" class="normalText">
<form action="renameFileOdhgos.jsp" method = "POST" name = "form1">
<input type= "hidden" name = "name" value="<%=fileName%>"/>
<table width = "100%" cellpadding="0" cellspacing="0" border="0">
<tr>
<td colspan = "2" class="normalText"><b>Παρακαλώ αλλάξτε το
όνομα του αρχείου και πατήστε συνέχεια</b></td>
</tr>
<tr>
<td colspan = "2" class="normalText">&nbsp;</td>
</tr>
<tr>
<td width="30%" class="normalText">Όνομα αρχείου :</td>
<td width="70%"><input type = "text" name = "newName" value
="<%=fileName%>" size="45" class="whiteButton"/></td>
</tr>
<tr>
<td colspan = "2">&nbsp;</td>
</tr>
<tr>
<td colspan=2><center><input type="submit" name="submit"
value="Συνέχεια" class="whiteButton"></center></td>
</tr>
</table>
</form>
<%
} //end else
%>
</body>
</html>

```

Π.30 replaceFile.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>

<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%
String fileName = request.getParameter("name");
%>
<link href="stylesheets/stylessheet.css" rel="stylesheet"
type="text/css">
<body bgcolor="white" class="normalText">
<form action="replaceFileAction.jsp" method = "POST"
enctype="multipart/form-data" name = "form1" >
<input type= "hidden" name = "name" value="<%=fileName%>"/>
<table width = "100%" cellpadding="0" cellspacing="0" border="0">
<tr>

```

```

        <td colspan = "2" class="normalText"><b>Παρακαλώ επιλέξτε το
        όνομα του αρχείου που υπάρχει τοπικά στον υπολογιστή σας και πατήστε
        συνέχεια</b></td>
    </tr>
    <tr>
        <td colspan = "2" class="normalText">&nbsp;</td>
    </tr>
    <tr>
        <td width="30%" class="normalText">Όνομα αρχείου :</td>
        <td width="70%"><input type = "file" name = "editedFile"
        class="whiteButton"/></td>
    </tr>
    <tr>
        <td colspan = "2">&nbsp;</td>
    </tr>
    <tr>
        <td colspan=2><center><input type="submit" name="submit"
        value="Συνέχεια" class="whiteButton"></center></td>
    </tr>
</table>
</form>
</body>
</html>

```

Π.31 *replaceFileAction.jsp*

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page import = "java.io.File" %>
<%@ page import="java.io.FileInputStream" %>
<%@ page import="java.io.FileOutputStream" %>
<%@ page import="java.util.Enumeration" %>
<%@ page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page import="com.oreilly.servlet.multipart.Part"%>
<%@ page import="com.oreilly.servlet.ParameterParser"%>
<%
String fileType="";
String dirName = "";
String filePath="";
String newName = "";
String fileName = "";

//File size is 10MB
MultipartRequest multi =
    new MultipartRequest(request,
request.getRealPath("tmp"), 10000 * 1024,
    "iso-8859-7", new
com.oreilly.servlet.multipart.DefaultFileRenamePolicy());
Enumeration files = multi.getFileNames();
while (files.hasMoreElements()) {
    fileName = multi.getParameter("name");
    fileType = fileName.substring(fileName.length()-3);

```

```

        if (fileType.equals("pdf")){
            dirName
pageContext.getServletContext().getRealPath("/pdfManager/pdfs");
        }else if (fileType.equals("doc")){
            dirName
pageContext.getServletContext().getRealPath("/pdfManager/msdocs");
        };
        filePath = dirName + File.separator + fileName;
            String fname = (String)files.nextElement();
            String strGotFile
multi.getFileSystemName(fname);
            String strGotFilename
multi.getOriginalFileName(fname);
            File fInput = multi.getFile(fname);
            if (fInput != null) {
                File fTarget = new File(filePath);
                FileInputStream fis = new
FileInputStream(fInput);
                FileOutputStream fos = new
FileOutputStream(fTarget);
                byte[] buf = new
byte[8192];
                while (true) {
                    int count =
fis.read(buf, 0, 8192);
                    if (count == -
1) break;
                    fos.write(buf,
0, count);
                } //while
                fis.close();
                fos.close();
                fInput.delete();
            };
        };
%>
<script language="javascript" type="text/JavaScript">
function refreshMe(){
    opener.parent.frames[0].location = 'pdfList.jsp'
    opener.parent.frames[1].location = 'docList.jsp'
    opener.parent.frames[2].location = 'pdfDocMenu.jsp'
    self.close()
}
</script>
<link href="stylesheets/stylesheets.css" rel="stylesheet"
type="text/css">
<head>
<title>Ενημέρωση Κεφαλαίου</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-
7">
</head>
<body bgcolor="white" onLoad = "refreshMe()">
</body>
</html>

```


Π.32 showGuidesBottom.jsp

```
<html>
<head>
<title>Folder Navigation</title>
<script language="javascript" type="text/JavaScript">
function winopen(fname,foptions){
newWindow = window.open(fname,'dialogWindow',foptions)
//,'width=400,height=300'
newWindow.focus()
}
function goBack(){
if (confirm('Είστε σίγουροι πως θέλετε να επιστρέψετε στο αρχικό
menu?')){
parent.close()
}
}
function delFile(){
file2del = form1.selFileName.value
//alert(file2del)
if (file2del==''){
alert('Παρακαλώ επιλέξτε αρχείο για διαγραφή')
return false
}
if (confirm('Είστε σίγουροι πως θέλετε να διαγράψετε το αρχείο
' + file2del + '?')){
newWindow
window.open('deleteFileOdhgos.jsp?name='
file2del,'dialogWindow','width=100,height=100')
//parent.close()
}
}
function prevFile(){
file2open = form1.selFileName.value
if (file2open==''){
alert('Παρακαλώ επιλέξτε αρχείο για προβολή')
return false
}

filePath = 'odhgoi/' + file2open
newWindow = window.open(filePath,'previewWin')
}
function renFile(){
file2rep = form1.selFileName.value
if (file2rep==''){
alert('Παρακαλώ επιλέξτε αρχείο για μετονομασία')
return false
}
newWindow = window.open('renameFileOdhgos.jsp?name='
file2rep,'dialogWindow','width=400,height=300')
}
function helpMe(){

}

</script>

<link href="stylesheets/stylessheet.css" rel="stylesheet"
type="text/css">
```

```

</head>

<body      topmargin="15"      leftmargin="15"      marginwidth="15"
marginheight="11" bgcolor="white" alink="black" link="black">
<form name="form1" method="POST">
<input type="hidden" name="selFileName">
<input type="hidden" name="selhrefname" value="dummy">
<table width="100%" cellspacing="0" cellpadding="0"
border="0">
      <tr>
            <td align="center"><input name="butEditFile"
value="Προβολή" type="button" class="whiteButton" onClick =
"prevFile()" /></td>
            <td align="center"><input name="butRenFile"
value="Μετονομασία" type="button" class="whiteButton" onClick =
"renFile()" /></td>
            <td align="center"><input name="butDelfile"
value="Διαγραφή" type="button" class="whiteButton" onClick =
"delFile()" /></td>
            <td align="center"><input name="Back"
value="Επιστροφή" type="button" class="whiteButton" onClick =
"goBack()" /></td>
            <td align="center"><input name="helpBut"
value="Βοήθεια" type="button" class="whiteButton" onClick =
"helpMe()" /></td>
      </tr>
    </table>
</form>
</body>
</html>

```

Π.33 showGuidesMain.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
%>
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import = "gr.ntua.ece.util.fileUtils" %>
<%@ page errorPage="errorhandler.jsp"%>
<%@ page import = "java.io.File" %>
<%
fileUtils fUtils = new fileUtils();
String      dirName
pageContext.getServletContext().getRealPath("/pdfManager/odhgoi");
dirName = dirName + File.separator;
String filePath;
%>
<html>
<style type="text/css">
<!--
a {
    text-decoration: none;
    color: blue;
}
-->

```

```

</style>
<script language="javascript" type="text/JavaScript">

function setSelected(item, fileName) {
eval (parent.frames[1].form1.selhrefname.value + ".style.color =
'blue'")
item.style.color = 'red'
parent.frames[1].form1.selhrefname.value = 'parent.frames[0].' +
item.name
parent.frames[1].form1.selFileName.value = fileName
}

function previewMe(fileName){
alert(fileName)
}

</script>
<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
<body class="normalText">
<a href="#" name="dummy"></a>
<table width="100%" cellspacing="0" cellpadding="0" border="0">
<tr>
<td class="normalText">Οι διαθέσιμοι οδηγοί Σπουδών φαίνονται
παρακάτω. Επιλέξτε το αρχείο που θέλετε, και κατόπιν επιλέξτε από το
menu την αντίστοιχη ενέργεια που θέλετε να κάνετε</td>
</tr>
<%
String[] pdfnames = fUtils.getPdfDirContents(dirName);
for (int i = 0; i < pdfnames.length; i++) {
String fname = pdfnames[i];
%>
<tr>
<td></td>
</tr>
<tr>
<td class="normalText"><a href="#" name="<%= "pdf" + i%>"
onClick = "setSelected(this, '<%=fname%>')" "><%=fname%></a></TD>
</tr>
<%
}
%>
</body>
</html>

```

Π.34 tree.jsp

```

<%
request.setCharacterEncoding("iso-8859-7");
String fName = request.getParameter("fname");
%>
<%@ page contentType = "text/html;charset=iso-8859-7" %>
<%@ page import = "org.dom4j.Document" %>

```

```

<%@ page import = "org.dom4j.Element" %>
<%@ page import = "org.dom4j.Node" %>
<%@ page import = "org.dom4j.io.SAXReader" %>
<%@ page import = "java.io.File" %>
<%@ page import = "gr.ntua.ece.util.NavigateXmlUtils" %>
<%@ page import = "java.util.List"%>
<%@ page import = "java.util.Iterator" %>
<%
String separatorChar = File.separator;
String fileName =
pageContext.getServletContext().getRealPath("/pdfManager/xmltrees");
fileName = fileName + separatorChar + fName;
//out.println(fileName);
SAXReader reader = new SAXReader();
Document xmlDoc=null;
try {
    xmlDoc = reader.read(new File(fileName));
} catch (Exception e) {
    e.printStackTrace();
    response.sendRedirect("errorLoafFile.html");
}
%>
<html>
<head>
<title>Folder Navigation</title>
<META NAME="CODED BY" CONTENT="nkir">
<style type="text/css">
<!--
a {
    text-decoration: none;
}
-->
</style>
<link href="stylesheets/styleSheet.css" rel="stylesheet"
type="text/css">
<script language="javascript" src="foldernav.js"></script>
<script language="javascript" type="text/JavaScript">
</script>
</head>
<body topmargin="15" leftmargin="15" marginwidth="15"
marginheight="11" bgcolor="white" alink="black" vlink="black"
link="black">
<form name="tempForm" method="POST"><input type="hidden" name =
"floaded" value = "<%=fName%>"/>
<input type="hidden" name = "selectednode" value="ma"/>
</form>
<div ID="mParent" class="parent">
<a name="ma" href="#" onClick="FolderExpand('m','top',this)">
<font
color="black" size="4" class="item"
face="Arial">Περίεχόμενα</font></a>
</div>
<div ID="mChild" ><%
NavigateXmlUtils xmlNav = new NavigateXmlUtils(xmlDoc);
List xpathList = xmlNav.getXpathList();
String str2write;
boolean[] depthArray = new boolean[30];
int divsOwned = 0;
int currDepth=0;
for (Iterator iter = xpathList.iterator(); iter.hasNext();) {

```

```

String xpath = (String) iter.next();
Node chapterNode = xmlDoc.selectSingleNode(xpath);
Element chapterElement = (Element)chapterNode;
xpath = chapterElement.getUniquePath();
int findDepth = xmlNav.findDepth(xpath);
boolean isLeaf =
(chapterElement.elements("chapter").size()==0);
String smallXPath = xpath.replace('/', '_');
char[] tempArr = new char[smallXPath.length()];
int j=0;
for (int i = 0; i < smallXPath.length(); i++) {
    if ((smallXPath.charAt(i) != '[') &&
(smallXPath.charAt(i) != '['))
        tempArr[j++]=smallXPath.charAt(i);
}
smallXPath = smallXPath.valueOf(tempArr);
smallXPath = smallXPath.trim();
smallXPath = smallXPath.substring(6, smallXPath.length());
//elegxo ean to chapter exei alla apo kato i oxi
boolean isLastChild = false;
String fatherXPath =
chapterElement.getParent().getUniquePath();
Node lastChildNode =
chapterElement.getParent().selectSingleNode("chapter[position()=last(
)]");
String lastChildNodeXPath =
lastChildNode.getUniquePath();
if (xpath.equals(lastChildNodeXPath)) {
    isLastChild=true;
    depthArray[findDepth]=true;
} else {
    depthArray[findDepth]=false;
};
//edo brika oti to sygkekrimeno node einai teleytaio
paidi toy patera toy
// o depthArray einai array me booleans poy me voithaei
na sxhmatizo to dentro me javascript
//ean ginei true, apo kato apo ayto to node se ayto to
epipedo tha typono kena anti gia grammes I
if ((isLastChild)&&(!(isLeaf))&&(findDepth!=1)){
    divsOwned++;
    currDepth = findDepth;
}
String titlos =
chapterElement.element("chaptertitle").getText();
if (isLeaf){
    //typose childNode
    str2write =
xmlNav.printChild(smallXPath, findDepth, isLastChild, titlos, depthArray,
xpath);
    out.println(str2write);
    if (isLastChild && (findDepth==currDepth+1)){
        for (int i = 0; i < divsOwned; i++) {
            out.println("</div>");
        }
        //System.out.println("findDepth: " +
findDepth);
        divsOwned=0;
    }
} else {
    //typose fatherNode

```

```
                str2write                                =
xmlNav.printFather (smallXPath, findDepth, isLastChild, titlos, depthArray
, xPath);
                out.println (str2write);
        }
}
%>

</body>
</html>
```