



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Επικοινωνιών, Ηλεκτρονικής & Συστημάτων Πληροφορικής

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Επεξεργασία και Μορφοποίηση σε εκτυπώσιμη μορφή πληροφορίας απο
βάσεις δεδομένων με χρήση τεχνολογίας XML/XSL**

του

Βλοτομά Νεόφυτου

Επιβλέπων : Νικόλαος Μήτρου

Αθήνα, Ιούνιος 2003

Περιεχόμενα

Συνομογραφίες	3
Πρόλογος	4
1. Εισαγωγή	5
1.1 Αντικείμενο της διπλωματικής	6
1.2 Οργάνωση του τόμου	6
2. Περιγραφή Διαθέσιμων Τεχνολογιών	8
2.1 Java	8
2.2 Java Servlets	10
2.3 Αντικειμενοστραφής σχεδιασμός με UML	10
2.4 XML	11
2.5 XSL	13
2.6 Adobe Acrobat Pdf	14
3. Υλοποίηση Συστήματος	16
3.1. Εισαγωγικά	16
3.2 PdfManager	16
3.3 updatePdfM	25
3.4 Δημιουργία XSL-FO	29
4. Λειτουργικό Σενάριο	35
4.1 Download των XML αρχείων απο την δικτυακή βάση του Ε.Μ.Π.	35
4.2 Λειτουργία PdfManager	36
4.3 Πρακτική εφαρμογή - Δημιουργία Οδηγού Σπουδών Ηλεκτρολόγων Μηχανικών Ε.Μ.Π.	47
5. Κώδικας	50
5.1 Πακέτο PdfManager	50
5.1.1 PdfManager.java	50
5.1.2 PdfCreator.java	65
5.1.3 Info.java	69
5.1.4 EditSourceFrame.java	71
5.1.5 DynamicTree.java	92
5.1.6 DisplayUrl.java	94
5.1.7 CreatePdfFrame.java	97
5.1.8 Concat.java	111
5.1.9 Acceptedfiles.java	122
5.1.10 SplitPdf.java	123
5.2 Πακέτο UpdatePdfM	129
5.2.1 ConvertDBtoXML.java	129
6. ΣΥΜΠΕΡΑΣΜΑΤΑ	138

Συντομογραφίες

HTML	HyperText Markup Language
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
PDF	Portable Document Format
PdfM	Η Εφαρμογή PdfManager
Uml	Unified Modeling Language
FOP	Formatting Objects Processor
API	Application Programming Interfaces
E.M.Π.	Εθνικό Μετσόβειο Πολυτεχνείο
Web Browser	Ξεφυλλιστής ιστού
XSL-FO	XSL-Formating Objects

Πρόλογος

Η παρούσα διπλωματική εργασία εκπονήθηκε κατά τη διάρκεια του εαρινού εξαμήνου του ακαδημαϊκού έτους 2002-2003 από τον Βλοτομά Νεόφυτο .Η υπόδειξη του θέματος έγινε από τον καθηγητή του Εθνικού Μετσόβειου Πολυτεχνείου κ. Μήτρου Νικόλαο με αντικειμενικό σκοπό την δημιουργία του οδηγού σπουδών του τμήματος Ηλεκτρολόγων Μηχανικών . Η εφαρμογή αυτή μπορεί να χρησιμοποιηθεί και για την δημιουργία άλλων αρχείων με επέκταση PDF .

Η μετατροπή του τόμου σε εκτυπώσιμη μορφή έγινε με την βοήθεια της συγκεκριμένης εφαρμογής .

Βλοτομάς Νεόφυτος
Ιούνιος 2003

1.ΕΙΣΑΓΩΓΗ

Στα πλαίσια των νέων δεδομένων της σημερινής εποχής, όπου επικρατούν ο συνεχόμενος καταμερισμός της εργασίας, η επιτάχυνση των διαδικασιών της οικονομικής ολοκλήρωσης, η υπερβολική ανάπτυξη των εταιρειών και ιδίως των πολυεθνικών επέφεραν αλλαγές στον τρόπο που οι άνθρωποι ζουν αλλά κυρίως στον τρόπο εργασίας τους. Αυτή η αλλαγή συνέπεσε και ίσως έχει και βαθύτερη σχέση με την ανάπτυξη της τεχνολογίας. Ο Παγκόσμιος Ιστός (World Wide Web) και το Διαδίκτυο (Internet) συνέβαλαν τα μέγιστα σε αυτή την διαφοροποίηση. Οι εταιρείες ξέφυγαν από τα στενά τους δεδομένα και άρχισαν να αναπτύσσουν διαφορετικές και περισσότερες επιχειρηματικές δραστηριότητες. Όμως δημιουργήθηκαν και κάποια προβλήματα. Οι εταιρείες έπρεπε να εισαγάγουν τις νέες τεχνολογίες και στην παραγωγή αλλά και στη διοίκηση. Τα δεδομένα πλέον έγιναν τα πιο χρήσιμα εργαλεία και τα πιο προχωρημένα 'όπλα' στην χάραξη στρατηγικής για της εταιρείες αλλά και στις καθημερινές τους συναλλαγές. Η επεξεργασία των δεδομένων παράγει την πληροφορία και η πληροφορία με τη σειρά της την γνώση.

Το πρόβλημα που δημιούργησε η νέα αυτή κατάσταση ήταν ο τεράστιος όγκος αυτών των δεδομένων και η ταχύτητα με την οποία αυτά μεταβάλλονταν τα οποία έκαναν πολύ δύσκολη την παρουσίαση των δεδομένων που ενδιέφεραν έναν χρήστη. Σε αυτό το πρόβλημα οι υπάρχουσες τεχνολογίες των Βάσεων Δεδομένων (Data Bases) δεν μπορούσαν να δώσουν απάντηση.

Τα τελευταία χρόνια τα εργαλεία και τα συστήματα παρουσίασης της πληροφορίας γίνονται όλο και πιο απαραίτητα σε μία ποικιλία περιβαλλόντων στο Internet. Οι εταιρείες έχουν στρέψει το ερευνητικό τους ενδιαφέρον στην όσο το δυνατόν πιο γρήγορη και αποδοτική εξόρυξη της χρήσιμης πληροφορίας από τις βάσεις δεδομένων και την παρουσίαση των αποτελεσμάτων στον χρήστη.

Ο ψηφιακός τρόπος παρουσίασης εξαρτάται κατά κύριο λόγο από την μορφή της πληροφορίας και κατά δεύτερο από τον χρήστη στον οποίο απευθύνεται. Η HTML είναι η τυπική γλώσσα για τη δημιουργία και την εμφάνιση ιστοσελίδων. Ωστόσο υπάρχουν αρκετά λογισμικά διαθέσιμα στο Internet για κάθε τύπο πληροφορίας όπου μπορούν να ενσωματωθούν σε μια ιστοσελίδα και να έχουμε το επιθυμητό αποτέλεσμα.

Εδώ και πολλά χρόνια το λογισμικό Adobe Acrobat έχει δημιουργήσει ένα προσιτό και υπαρκτό πρότυπο για την διανομή εγγράφων, το Adobe Portable Document Format (PDF). Το συγκεκριμένο λογισμικό είναι η ιδανική λύση για την κατασκευή και διανομή ηλεκτρονικών εγγράφων. Τα έγγραφα που

παράγονται μπορούν εύκολα να διανεμηθούν στο διαδίκτυο και να γίνουν προσβάσιμα από άλλους χρήστες .

1.1 Αντικείμενο της διπλωματικής

Το θέμα της παρούσης διπλωματικής εργασίας αφορά την Επεξεργασία και Μορφοποίηση σε εκτυπώσιμη μορφή πληροφορίας από βάσεις δεδομένων με χρήση τεχνολογίας XML/XSL.

Στα πλαίσια της διπλωματικής θα παρουσιασθεί πώς παίρνουμε πληροφορίες από την δικτυακή βάση του Εθνικού Μετσόβειου Πολυτεχνείου σε μορφή XML και με την βοήθεια XSL αρχείων γίνεται η παρουσίαση των δεδομένων σε εκτυπώσιμη μορφή , πιο συγκεκριμένα PDF ,κειμένου .Τα νέα αρχεία που δημιουργούνται είναι κεφάλαια του οδηγού σπουδών του τμήματος Ηλεκτρολόγων Μηχανικών του Ε.Μ.Π. Ακολούθως με την προσθήκη αρχείων που ήδη υπάρχουν (με στατικό περιεχόμενο) έχουμε σαν αποτέλεσμα τον ολοκληρωμένο οδηγό σπουδών του τμήματος .Λόγω της ευρείας χρησιμοποίησης αρχείων σε PDF μορφή η εφαρμογή αυτή έχει επεκταθεί για να μπορεί να χρησιμοποιηθεί και για την δημιουργία άλλων κειμένων με στατικό ή δυναμικό περιεχόμενο. Το interface της εφαρμογής είναι γραμμένο σε Java και απευθύνεται σε όλους τους χρήστες .

Στο νέο αρχείο που δημιουργείται υπάρχει η δυνατότητα εκμετάλλευσης όσο το δυνατόν καλύτερα των προηγμένων τεχνολογιών που μπορούν να προσφέρουν τα PDF αρχεία.Οι τεχνολογίες που έχουν αναφερθεί παρουσιάζονται αναλυτικά σε επόμενο κεφάλαιο.

1.2 Οργάνωση του τόμου .

Ο παρόν τόμος αποτελείται από πέντε κεφάλαια .

- Στο πρώτο κεφάλαιο εκτός από την παρούσα περιγραφή γίνεται μια εισαγωγή για το αντικείμενο της διπλωματικής με την οποία ο αναγνώστης μπορεί να αποκτήσει μια συνολική ιδέα της διπλωματικής εργασίας.
- Στο δεύτερο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο της εργασίας. Γίνεται εισαγωγή σε θέματα με τα οποία ο αναγνώστης πρέπει να είναι εξοικειωμένος προκειμένου να παρακολουθήσει τη ροή της εργασίας. Πιο συγκεκριμένα παρουσιάζονται συνοπτικά τεχνολογίες που αφορούν την γλώσσα προγραμματισμού Java ,τα Java Servlets ,UML ,XML,XSL ,AdobeAcrobat PDF , και XSL-FO . Ο αναγνώστης που

κατέχει τα θέματα αυτά μπορεί να παραλείψει την ανάγνωσή τους και να προσφύγει σε αυτά ως αναφορά όποτε αυτό κριθεί απαραίτητο .

- Στο τρίτο κεφάλαιο γίνεται περιγραφή της υλοποίησης του συστήματος με την βοήθεια UML διαγραμμάτων .Επίσης περιγράφονται κρίσιμες λειτουργίες της εφαρμογής.
- Στο τέταρτο κεφάλαιο γίνεται αρχικά αναφορά στις δυνατότητες του συστήματος και ακολουθεί επίδειξη της λειτουργίας του με την δημιουργία του οδηγού σπουδών του τμήματος Ηλεκτρολόγων Μηχανικών του Ε.Μ.Π.
- Στο πέμπτο κεφάλαιο παρουσιάζεται ο κώδικας της εφαρμογής .
- Στο έκτο κεφάλαιο διατυπώνονται τα τελικά συμπεράσματα σχετικά με την μελλοντική χρησιμοποίηση της εφαρμογής και επεκτάσεις που μπορούν να γίνουν σε μελλοντικό στάδιο .

2.ΠΕΡΙΓΡΑΦΗ ΔΙΑΘΕΣΙΜΩΝ ΤΕΧΝΟΛΟΓΙΩΝ

2.1 Java

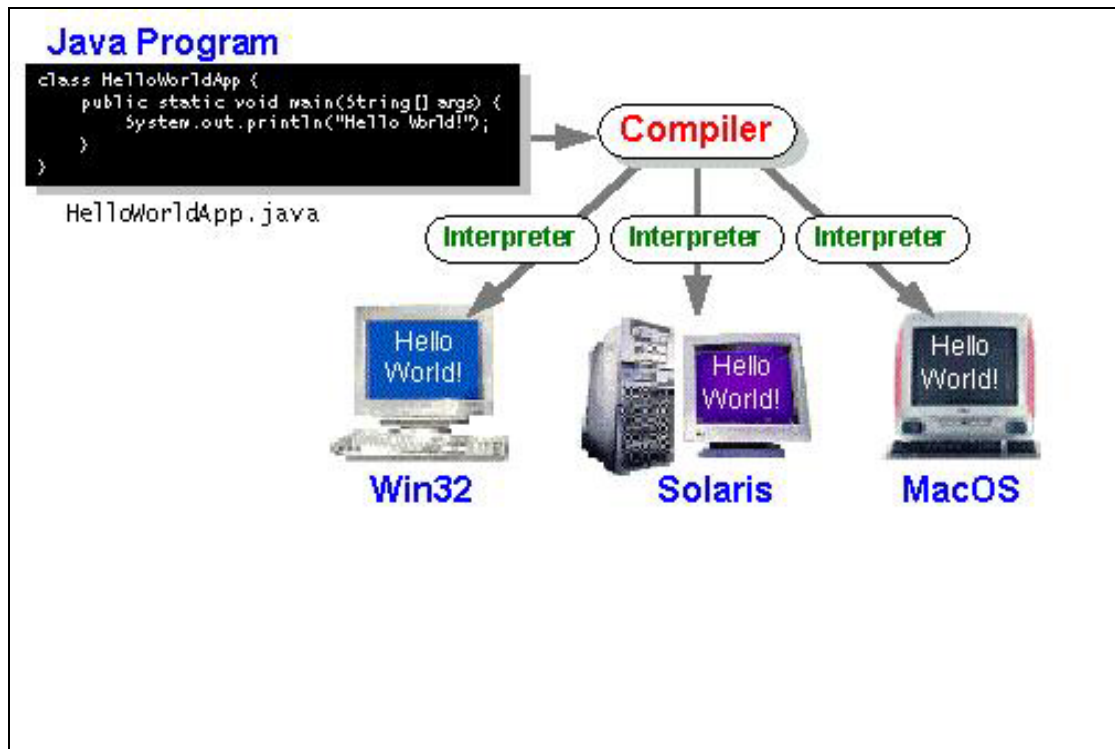
Η γλώσσα προγραμματισμού Java προκάλεσε ίσως το μεγαλύτερο ενδιαφέρον σε σύγκριση με οποιαδήποτε άλλη εξέλιξη στον κόσμο του Internet. Είναι η πρώτη που κατάφερε να συμπεριλάβει ήχο και κίνηση σε μια ιστοσελίδα. Επιπλέον επιτρέπει στους χρήστες να αλληλεπιδρούν (interact) με την ιστοσελίδα. Εκτός από το να διαβάζει απλά και ίσως να συμπληρώνει μία φόρμα, ο χρήστης μπορεί τώρα να παίζει παιχνίδια, να συνομιλήσει, να λαμβάνει συνεχώς τις πιο πρόσφατες πληροφορίες και πολλά άλλα .

Είναι μία γλώσσα προγραμματισμού πολύ υψηλού επιπέδου . Αναπτύχθηκε από την εταιρεία Sun microsystems και γρήγορα έλαβε μεγάλη δημοσιότητα λόγω της ανεπανάληπτης χρησιμότητάς της στην ανάπτυξη (κυρίως Web) εφαρμογών. Χαρακτηρίζεται από τα εξής : απλή (simple), αντικειμενοστραφής (object-oriented), κατανεμημένη (distributed), εύρωστη (robust), μεταφέρισιμη (portable), υψηλής απόδοσης (high performance), πολυνηματώδης (multithreaded), ασφαλής (secure) .

Με την Java γράφονται δύο ειδών προγράμματα:

- Ανεξάρτητες εφαρμογές (applications) που τρέχουν σε οποιαδήποτε πλατφόρμα που περιέχει την JVM.
- Εφαρμογές που τρέχουν στους Web browsers και ονομάζονται applets. Καθώς ένας browser μεταφράζει μία HTML σελίδα, μόλις συναντήσει το tag της HTML <applet> ανακτά από τον server ένα πρόγραμμα Java (ήδη μεταφρασμένο) και μέσω του interpreter του browser το μετατρέπει σε bytewords και κατόπιν το εκτελεί. Έτσι οι HTML σελίδες παύουν να είναι πλέον στατικές και αποκτούν δυναμικότητα στον υπολογιστή του browser, εκτελώντας σ' αυτόν πλήθος χρήσιμων λειτουργιών που θα ήταν αδύνατο να επιτευχθούν διαφορετικά.

Η ανεξαρτησία απο το υπολογιστικό σύστημα είναι ο σημαντικότερος λόγος για την επιλογή της γλώσσας προγραμματισμού Java για την δημιουργία του interface της εφαρμογής μας .Η δυνατότητα δηλαδή ενός προγράμματος να μεταφέρεται εύκολα απο ένα είδος υπολογιστή σ' ένα άλλο . Τα προγράμματα μπορούν να τρέχουν σε οποιοδήποτε σύστημα για το οποίο έχει εγκατασταθεί μια «εικονική μηχανή» Java.Η εικόνα 2.1 παρουσιάζει τα αποτελέσματα αυτής της διαδικασίας : πολλαπλά εκτελέσιμα προγράμματα για διαφορετικά συστήματα.

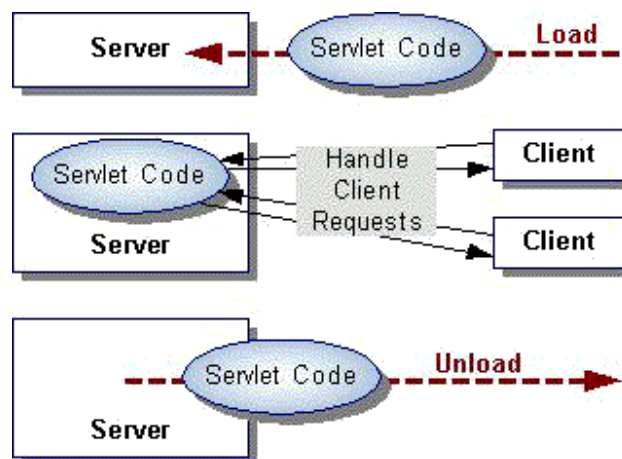


εικόνα 2.1

Επιπλέον η Java περιέχει τα λεγόμενα Java APIs (Application Programming Interfaces). Ένα Java API είναι μία συλλογή από έτοιμα προγράμματα. Το κάθε API παρέχει μία ολόκληρη βιβλιοθήκη με κλάσεις που καλύπτουν τις προγραμματιστικές ανάγκες για ένα συγκεκριμένο πεδίο εφαρμογών. Για παράδειγμα υπάρχουν τα παρακάτω APIs : AWT API (Abstract Window Toolkit, για graphical User Interfaces), JDBC API (Java DataBase Connectivity) για σύνδεση με βάσεις δεδομένων), Commerce API, Security API, Management API, RMI API (Remote Method Invocation), JavaBeans API, Servlet API (για κατασκευή εφαρμογών που τρέχουν αντί στον browser ,όπως κάνουν τα applets, στον Web Server) και πολλά άλλα που μπορεί να χρησιμοποιηθούν εύκολα .

2.2 Java Servlets

Τα servlets στην Java παρέχουν ένα πρωτόκολλο και μια μέθοδο (ανεξαρτήτως πλατφόρμας υλοποίησης) για δημιουργία web-based εφαρμογών. Ένα servlet τρέχει σε ένα web-server και αλληλεπιδρά με το servlet engine μέσω requests και responses. Ένα πρόγραμμα client, γραμμένο σε οποιαδήποτε γλώσσα, έχει πρόσβαση στον server και υποβάλλει ένα request. Μετά την επεξεργασία του στο servlet engine αποστέλλεται response πίσω στον client όπως βλέπουμε και στην εικόνα 2.2 .



εικόνα 2.2

Πιο αναλυτικά όταν ένας client κάνει μια αίτηση στον server τότε το servlet που είναι αποθηκευμένο στον server αποφασίζει αν είναι GET ή POST εφαρμογή .Ανάλογα καλεί την HttpServlet.doGet ή την HttpServlet.doPost κλάση όπου παίρνουν σαν παραμέτρους τις (HttpServletRequest) και (HttpServletResponse) .

Χρήσεις HTTP Servlets

- Επεξεργασία, αποθήκευση δεδομένων που υποβάλλονται από HTML form
- Παροχή δυναμικού περιεχομένου (επιστροφή ενός data query στον χρήστη)

2.3 Αντικειμενοστραφής σχεδιασμός με UML

Η Unified Modeling Language (UML) είναι μία γλώσσα που χρησιμοποιείται για προδιαγραφές, αναπαράσταση με οπτικό τρόπο (visualizing), δημιουργία και τεκμηρίωση των τμημάτων των συστημάτων λογισμικού, καθώς και για

μοντελοποίηση εταιρικών και άλλων συστημάτων που δεν αφορούν λογισμικό. Η UML αποτελεί ένα συνδυασμό των καλύτερων πρακτικών, οι οποίες ήδη έχουν αποδείξει πόσο επιτυχημένες ήταν στη μοντελοποίηση μεγάλων και σύνθετων συστημάτων.

Έχει αναπτυχθεί από τη Rational Software και τους εταίρους της. Επίσης πολλές εταιρίες έχουν ενσωματώσει τη UML ως πρότυπο στη διαδικασία ανάπτυξής τους και στα προϊόντα τους, τα οποία καλύπτουν περιοχές όπως η business modeling, η διαχείριση απαιτήσεων, η ανάλυση και ο σχεδιασμός, ο προγραμματισμός και ο έλεγχος.

Η ανάπτυξη ενός μοντέλου για ένα σύστημα λογισμικού που θα μπορέσει να χρησιμοποιηθεί στη βιομηχανία, προτού δομηθεί ή επανασχεδιαστεί το σύστημα, είναι τόσο απαραίτητο όσο το να υπάρχουν τα αρχιτεκτονικά σχέδια για ένα μεγάλο κτίσμα. Τα καλά μοντέλα είναι απαραίτητα για την επικοινωνία ανάμεσα στις ομάδες ενός έργου και επίσης για να υπάρχει μία ορθή αρχιτεκτονική του συστήματος. Δημιουργούμε μοντέλα των σύνθετων συστημάτων γιατί δεν μπορούμε να τα κατανοήσουμε στην ολότητά τους. Όσο αυξάνει η πολυπλοκότητα των συστημάτων, αυξάνει και η σημασία της ύπαρξης καλών τεχνικών μοντελοποίησης.

Σε ό,τι αφορά τις όψεις ενός μοντέλου, η UML ορίζει τα ακόλουθα γραφικά διαγράμματα:

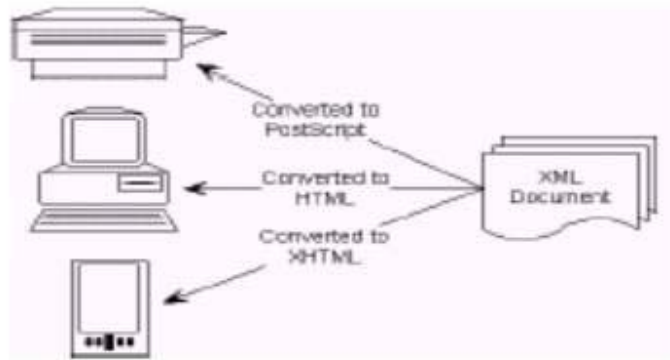
- Διάγραμμα περιπτώσεων χρήσης (use case)
- Διάγραμμα κλάσεων (class)
- Διαγράμματα συμπεριφοράς
- Διαγράμματα υλοποίησης

Τα διαγράμματα αυτά παρέχουν διαφορετικές απόψεις του συστήματος κατά τη φάση της ανάλυσης και της ανάπτυξης. Το υποκείμενο μοντέλο ολοκληρώνει τις απόψεις αυτές, ούτως ώστε να μπορεί να αναλυθεί και να δομηθεί ένα πλήρες, συνεπές με τον εαυτό του σύστημα. Τα διαγράμματα αυτά, καθώς και η σχετική τεκμηρίωση είναι τα βασικά στοιχεία που βλέπει αυτός που μοντελοποιεί, αν και η UML και τα εργαλεία που την υποστηρίζουν θα παρέχουν και ένα πλήθος παραγόμενων όψεων.

2.4 XML

Η μεταφορά πληροφοριών μέσω του Internet και μεταξύ εφαρμογών λογισμικού ήταν πάντοτε δύσκολη, λόγω των διαφορών σε μορφές δεδομένων και αποκλειστικών δομών. Την λύση στο πρόβλημα αυτό έδωσε η δημιουργία της XML (Extensible Markup Language).

Η XML είναι ένα πρωτόκολλο που βασίζεται σε πρότυπα και ελέγχεται από το κονσόρτσιουμ World Wide Web Consortium (W3C). Το κύριο πλεονέκτημα της είναι η επικέντρωση της στη δομή του εγγράφου, γεγονός που την κάνει ανεξάρτητη από το μέσο παράδοσης του εγγράφου .



εικόνα 2.3

Για να κατανοήσουμε την εφαρμογή αυτή της XML αρκεί να δώσουμε το εξής παράδειγμα (εικόνα 2.3) . Πολλές φορές κατά την εμφάνιση ενός site σε διαφορετικούς browsers συναντώνται ασυμβατότητες . Αυτό οδηγεί στην ανάγκη ανάπτυξης δύο ή περισσότερων εκδόσεων του ίδιου site. Το κόστος αυτού είναι πολλές φορές απαγορευτικό . Στο σημείο αυτό έρχεται η XML η οποία επιτρέπει να τηρείται μια έκδοση του site σε ανεξάρτητου μέσου μορφή, η οποία μπορεί να μετατραπεί οποιαδήποτε στιγμή σε HTML, Postscript, PDF, RTF και άλλες .

Η XML συνοδεύεται από μια σειρά προτύπων (standards). Παρακάτω αναφέρονται μερικά από αυτά .

- Φύλλα Στυλ (Style Sheets): Υπάρχουν δύο φύλλα στυλ: Το XSL(XML Stylesheet Language) και το CSS(Cascading Style Sheet). Τα φύλλα στυλ είναι πιθανότατα τα περισσότερο διαδεδομένα πρότυπα. Καθορίζουν πως πρέπει να απεικονίζονται τα έγγραφα στην οθόνη, σε χαρτί ή σε ένα επεξεργαστή κειμένου .
- XML Namespaces (Χώροι Ονοματοδοσίας): Το πρότυπο αυτό επιτρέπει στον προγραμματιστή της XML να προσθέτει σε υπάρχοντα στοιχεία ετικέτες χωρίς να δημιουργούνται αντιθέσεις στα ονόματα των στοιχείων, καθώς χρησιμοποιούνται συγκεκριμένες τεχνικές ονοματοδοσίας. Αυτές ορίζονται από το πρότυπο.
- DOM και SAX: Τα DOM (Document Object Model) και SAX (Simple API for XML) είναι προγραμματιστικά περιβάλλοντα για να έχουν πρόσβαση σε XML έγγραφα. Αυτά επιτρέπουν στις εφαρμογές να

διαβάζουν XML έγγραφα χωρίς να εμπλέκονται θέματα σύνταξης. Αυτά είναι συμπληρωματικά μεταξύ τους: Το DOM είναι καταλληλότερο για φόρμες και επεξεργαστές κειμένου, ενώ το SAX ενδείκνυται για ανταλλαγή δεδομένων μεταξύ εφαρμογών.

- Xlink και Xpointer: Είναι δύο μέρη ενός προτύπου που χρησιμοποιείται για την εδραίωση δεσμών μεταξύ των εγγράφων

Ένα XML έγγραφο αποτελείται από δεδομένα χαρακτήρων (character data) και σήμανση (markup). Και τα δύο αναπαρίστανται με κείμενο . Η πληροφορία βρίσκεται στα δεδομένα χαρακτήρων ενώ η σήμανση είναι σημαντική επειδή καταγράφει τη δομή του εγγράφου. Η ερμηνεία των δεδομένων αφήνεται εξ ολοκλήρου στην εφαρμογή που τα διαβάζει .

Κάθε δεδομένο της XML βρίσκεται ανάμεσα σε δύο ετικέτες, αρχής και τέλους . Επιπλέον, είναι πιθανή η προσθήκη επιπρόσθετων πληροφοριών στα στοιχεία με την μορφή των χαρακτηριστικών (attributes) .

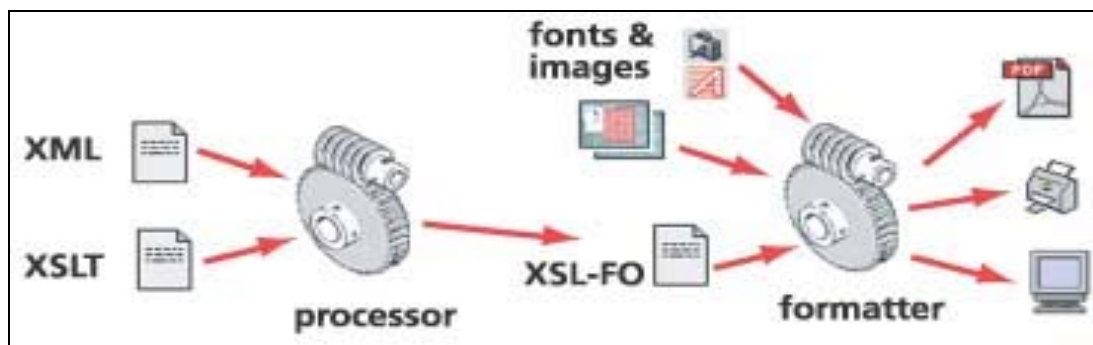
2.5 XSL

Η XSL ή αλλιώς Extensible Stylesheet Language δεν είναι μια γλώσσα όπως αρχικά τουλάχιστον φαίνεται αλλά δύο . Η XSL αποτελείται από μια γλώσσα που χρησιμοποιείται για μετασχηματισμούς και μια άλλη που χρησιμοποιείται για μορφοποιήσεις . Η γλώσσα που χρησιμοποιείται για μετασχηματισμούς ονομάζεται XSLT ή αλλιώς Extensible Stylesheet Language Transformations . Αυτή χρησιμοποιείται για το μετασχηματισμό XML εγγράφων σε άλλα XML έγγραφα . Η δεύτερη γλώσσα που περιέχεται στην XSL είναι η XSL-FO ή αλλιώς Extensible Stylesheet Language Formatting Objects . Αυτή η γλώσσα χρησιμοποιείται για να περιγράψει πως ένα έγγραφο XML θα μορφοποιηθεί για να παρουσιαστεί στον αναγνώστη .

Οι δύο γλώσσες που αποτελούν την XSL μπορούν να λειτουργούν αυτόνομα και ανεξάρτητα η μία από την άλλη , για παράδειγμα χρησιμοποιώντας την XSLT μπορούμε να μετασχηματίσουμε ένα XML έγγραφο σε καλά μορφοποιημένο έγγραφο HTML χωρίς να χρησιμοποιήσουμε την XSL-FO . Ένα έγγραφο XSLT ή ένα έγγραφο XSL-FO είναι ουσιαστικά και τα δύο , καλά μορφοποιημένα (well formed) έγγραφα XML που ακολουθούν τους κανόνες σύνταξης των XML εγγράφων . Αυτό σημαίνει ότι έχουν στην αρχή μια δήλωση XML , ένα στοιχείο ρίζα (root element) , στοιχεία παιδιά (child elements) και γενικά υπακούουν σε όλους εκείνους τους κανόνες που συνηγορούν σε ένα καλά μορφοποιημένο έγγραφο XML . Κατά συνθήκη , ένα αρχείο το οποίο περιέχει κανόνες μετασχηματισμού πρέπει να έχει κατάληξη .xsl ενώ ένα αρχείο που περιέχει κανόνες μορφοποίησης κατάληξη .fo . Αντί

για τις προηγούμενες καταλήξεις μπορεί να μπει και η κατάληξη .xml αφού όπως ειπώθηκε παραπάνω και τα δύο είναι καλά μορφοποιημένα έγγραφα XML .

Στην παρούσα εφαρμογή χρησιμοποιήθηκαν και οι δύο γλώσσες για την παραγωγή PDF αρχείων απο τα XML που παίρνουμε απο την δικτυακή βάση. Για την καλύτερη κατανόηση για το πως ακριβώς λειτουργεί βλέπουμε στην εικόνα 2.3 την διαδικασία που ακολουθείται για να παραχθεί το επιθυμητό αποτέλεσμα.



εικόνα 2.4

Βλέπουμε ότι με την βοήθεια ενός ειδικά διαμορφωμένου αρχείου XSLT τα δεδομένα μας (σε μορφή XML) μετασχηματίζονται σε ένα αρχείο XSL-FO . Στην συνέχεια με την προσθήκη των κατάλληλων fonts και εικόνων που μπορεί να θέλουμε να προσθέσουμε στο αρχείο μας είμαστε έτοιμοι να παράξουμε το PDF αρχείο .Για να γίνει αυτό χρησιμοποιούμε ένα Java API , το FOP (Formatting Objects Processor) .Η εφαρμογή αυτή διαβάζει το XSL-FO αρχείο και μετά παράγει το PDF .Το FOP API είναι μέρος του Apache's XML project και υποστηρίζει και άλλα formats όπως PCL, PS, SVG, XML , Print, AWT, MIF , TXT.Χρησιμοποιήσαμε το For 0.20.4 το οποίο υποστηρίζει XSL-FO Version 1.0 W3C.

2.6 Adobe Acrobat Pdf

Η Adobe Systems, από τους κορυφαίους δημιουργούς τεχνολογιών για προγράμματα έκδοσης και επεξεργασίας εικόνας, κατασκευάζει προϊόντα που απευθύνονται σε επαγγελματίες εκδότες, γραφίστες, σχεδιαστές web, επιχειρήσεις και ιδιώτες. Τα προϊόντα της Adobe επιτρέπουν την δημιουργία και έκδοση πλούσιων εικόνων και εγγράφων για όλα τα μέσα, έντυπα και ηλεκτρονικά.

Ένα απο αυτά είναι και το Adobe Acrobat το οποίο παρέχει τον πιο αξιόπιστο και αποτελεσματικό τρόπο ανταλλαγής πληροφορίας ηλεκτρονικά. Επιτρέπει την μετατροπή οποιουδήποτε εγγράφου σε Adobe Portable Document Format (PDF) χωρίς να αλλάζει την αρχική του εμφάνιση. Τα έγγραφα αυτά στην συνέχεια μπορούν να ανοιχτούν και να τυπωθούν από οποιοδήποτε σύστημα

Με το Acrobat έχουμε πολλές δυνατότητες εμπλουτισμού έτοιμων αρχείων PDF όπως της διόρθωσης ορθογραφικών λαθών, προσθήκης και αφαίρεσης σελίδων, ορισμού διαδρομών ανάγνωσης, ηλεκτρονικής υπογραφής, απόκρυψης. Επιπλέον προσφέρει πλήρη εργαλεία σχολιασμού αρχείων PDF τα οποία μπορούν αν αυξήσουν κατακόρυφα την παραγωγικότητα μιας επιχείρησης αφού μπορούν πλέον να σχολιάζουν και να εγκρίνουν έγγραφα με ηλεκτρονικό τρόπο, ακόμα και απευθείας μέσα από την οθόνη του web browser. Τέλος προσφέρει μοναδικές δυνατότητες κατασκευής ηλεκτρονικών φορμών και συλλογών εγγράφων, μετατροπής ολόκληρων ιστοσελίδων σε έγγραφα PDF, και οπτική αναγνώριση ξενόγλωσσου κειμένου.

Ένα Pdf αρχείο χρησιμοποιεί αντικείμενα (objects) για να περιγράψει το περιεχόμενο του όπως γραμματοσειρές, κείμενα, εικόνες ακόμα και οι σελίδες του αρχείου είναι αντικείμενα. Αυτά τοποθετούνται σε μια ιεραρχική δομή η οποία μοιάζει πολύ με την δομή του XML. Η χρησιμοποίηση των αντικειμένων γίνεται με μια σειρά απο εντολές που έχει το Acrobat Reader. Για να μειώσει το μέγεθος του κειμένου, κατά συνέπεια και το μέγεθος του αρχείου, το Acrobat χρησιμοποιεί αλγόριθμους συμπίεσης κειμένου.

Οι αλγόριθμοι συμπίεσης καθώς και η χρησιμοποίηση εντολών για την παρουσίαση καθιστούν την επεξεργασία ενός Pdf αρχείου αρκετά επίπονη διαδικασία. Λύση στο συγκεκριμένο πρόβλημα δόθηκε με την βοήθεια ενός Java API του iText. Αν και δεν καλύπτει όλες τις δυνατότητες που μας παρέχουν τα Pdf αρχεία (όπως η δημιουργία φορμών) εντούτοις μπορούμε εύκολα να γράψουμε, να συνενώσουμε αρχεία και να θέσουμε απόκρυψη με την βοήθεια του συγκεκριμένου API.

3. ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ

3.1. Εισαγωγικά

Στα πλαίσια αυτής της διπλωματικής επιχειρήθηκε να δημιουργηθεί μια εφαρμογή που να έχει σαν αποτέλεσμα ένα κείμενο σε PDF μορφή με τα εξής χαρακτηριστικά :

- να μπορεί να εκμεταλλευτεί όσο το δυνατόν καλύτερα τις δυνατότητες που παρέχει ένα PDF αρχείο
- να τρέχει σε ένα λειτουργικό περιβάλλον φιλικό προς τον χρήστη
- να μπορεί να εκμεταλλευτεί δυναμικά και στατικά δεδομένα
- να μπορεί εύκολα και γρήγορα να αλλάζει μορφή και περιεχόμενο
- να μπορεί να επεκταθεί εύκολα στο μέλλον

Ξεκινώντας ονόμασα το πακέτο που βρίσκεται το Interface της εφαρμογής μου PdfM (pdfmanager) το οποίο είναι ένα Gui γραμμένο σε γλώσσα προγραμματισμού Java .Για την μετατροπή των δυναμικών δεδομένων σε PDF μορφή δημιούργησα τα κατάλληλα XSL.Για να πάρουμε τα δεδομένα σε κατάλληλη XML μορφή δημιούργησα ένα Java Servlet το οποίο παίρνει πληροφορίες απο την δικτυακή βάση δεδομένων του Ε.Μ.Π. και βρίσκεται σε ένα ξεχωριστό πακέτο , το updatePDFM ,για να τοποθετηθεί στον server της σχολής .

Στο κεφάλαιο αυτό θα γίνει μια περιγραφή των κύριων στοιχείων που σχεδιάστηκαν.

3.2 PDFManager

Το PdfM είναι ένα πρόγραμμα γραμμένο σε Java το οποίο είναι μια ανεξάρτητη εφαρμογή που διαχειρίζεται PDF αρχεία .Σκοπός μου ήταν η δημιουργία κάποιου interface που θα είναι εύκολο απο τον χρήστη να δημιουργήσει PDF αρχεία βάση των δικών του απαιτήσεων και εύκολα τροποποιήσιμο .

Για να γίνει πιο εύκολη η εγκατάσταση του όλες οι απαραίτητες βιβλιοθήκες μαζί με την Java (jdk1.3.1) έχουν εγκατασταθεί στο directory του PdfM . Έπιπλέον έχω δημιουργήσει ένα MS-DOS Batch file ,το Start.bat , που θέτει στην διαδρομή καταλόγων του συστήματος (classpath) της βιβλιοθήκες που χρειάζεται και καλεί την κλάση που βρίσκεται η main της εφαρμογής .Έτσι για να τρέξει η εφαρμογή χρειάζεται μόνο η αντιγραφή του directory του PdfM που βρίσκεται στο CD-ROM (μπορεί να τρέξει και απ' αυτό) σε directory της επιλογής μας και ακολούθως η επιλογή του Start.bat για να ξεκινήσει .

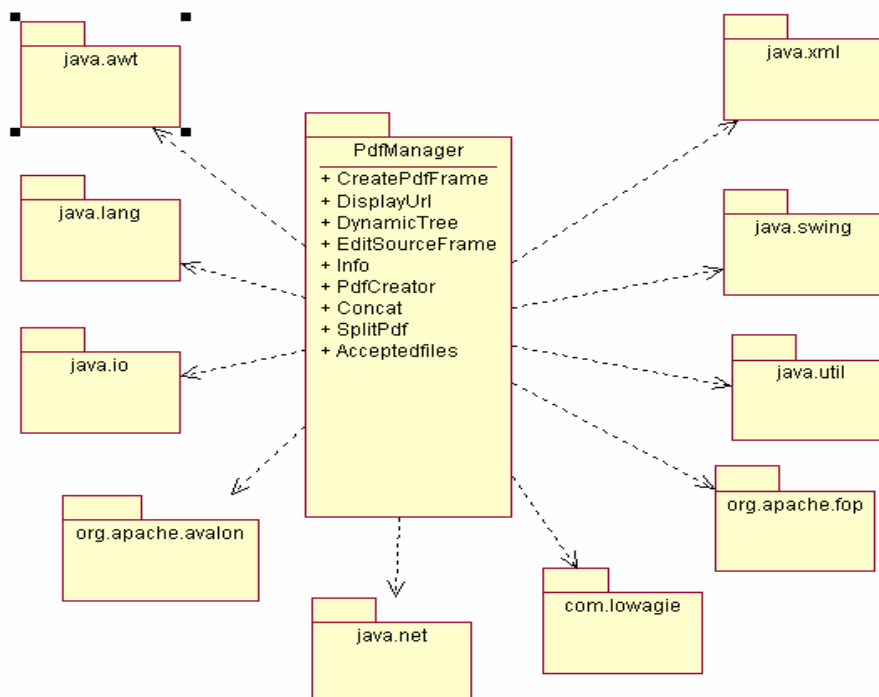
Για την κατασκευή της εφαρμογής χρησιμοποιήθηκαν τα παρακάτω API όπως παρουσιάζεται και στο UML διάγραμμα.

Κλάσεις που είναι διαθέσιμες σε οποιαδήποτε υλοποίηση της Java

- java.lang.*
- java.awt.*
- java.io.*
- java.net.*
- java.util.*

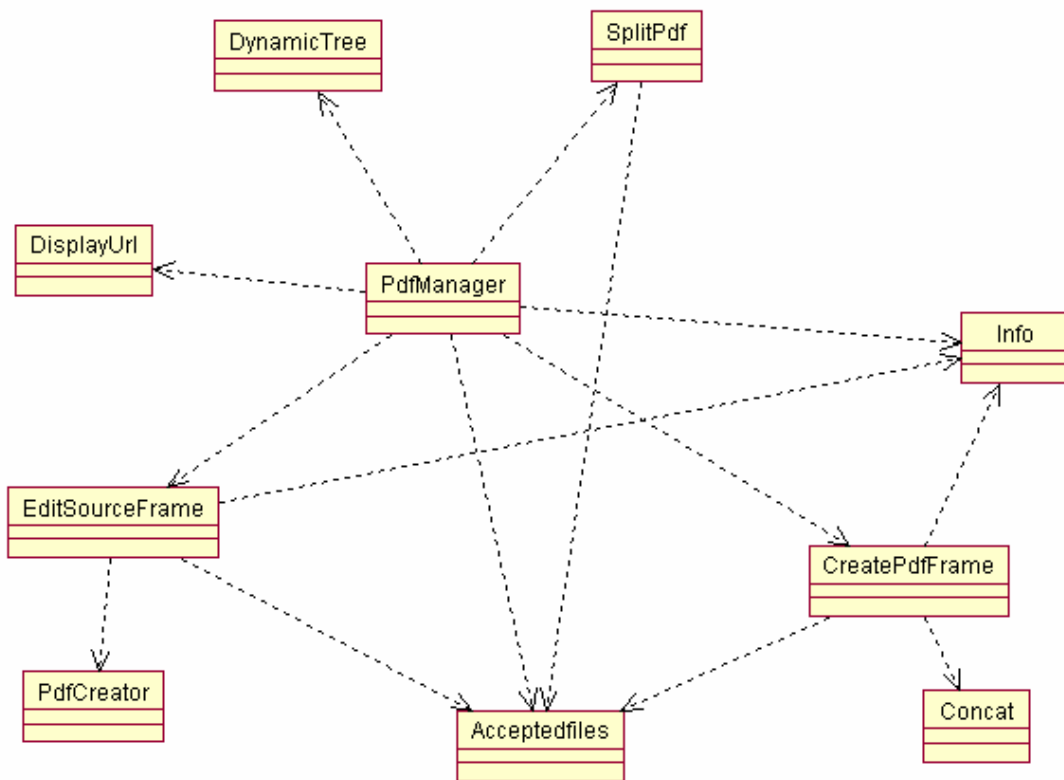
Επιπρόσθετες κλάσεις

- javax.swing.* Το Swing package είναι μέρος της JFC (Java Foundation Classes) .Χρησιμοποιείται για την κατασκευή όλων των αντικειμένων που βοηθούν στην επικοινωνία της εφαρμογής με τον χρήστη .
- javax.xml.* που παρέχει τις κλάσεις για την επεξεργασία των XML και XSL αρχείων
- org.apache.fop.* που είναι μέρος του Apache XML project και χρησιμοποιείται για την παραγωγή των νέων PDF απο αρχεία XSL-FO
- org.apache.avalon.* που είναι μέρος του Apache Avalon project και χρησιμοποιείται για την αρχικοποίηση του FOP .
- com.lowagie.* που είναι μέρος του iText Project και χρησιμοποιείται για την διαχείριση PDF αρχείων .



Η Java με τη χρήση της κληρονομικότητας (inheritance) και την κατασκευή πακέτων (packages) από κλάσεις, δίνει την δυνατότητα σωστής και ολοκληρωμένης διαχείρισης του κώδικα. Με την κληρονομικότητα μπορεί μια κλάση να κληρονομήσει όλα τα members (variables και methods) μιας άλλης κλάσης (της superclass) . Κατ' αυτό τον τρόπο γίνεται αντικειμενοστραφής οργάνωση του κώδικα και επαναχρησιμοποίησή του. Με την οργάνωση του κώδικα σε packages οι κλάσεις είναι ευκολότερο να βρεθούν, αποφεύγονται οι συγκρούσεις ονομάτων και γίνεται καλύτερος έλεγχος στην πρόσβαση μεταξύ των κλάσεων.

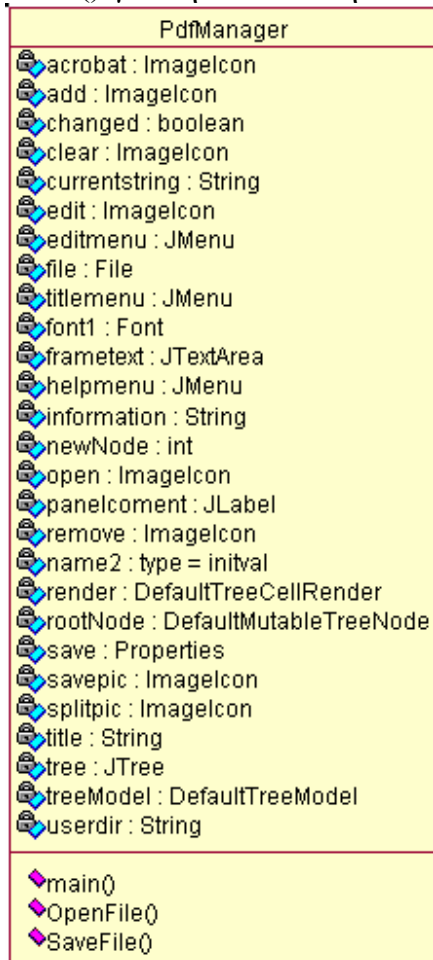
Ο κώδικας του PdfM είναι οργανωμένος σε ένα πακέτο το οποίο περιέχει κλάσεις που σχετίζονται μεταξύ τους . Στην εικόνα 3.1 φαίνεται το διάγραμμα όλων των τάξεων χωρίς να περιλαμβάνονται οι ιδιότητες (attributes)και οι λειτουργίες(operations) της κάθε τάξης λόγω έλλειψης χώρου .Αυτό γίνεται για να φανούν καλύτερα οι σχέσεις (relationships) που έχουν οι κλάσεις μεταξύ τους μέσα στο πακέτο .Οι υπόλοιπες ιδιότητες της κάθε κλάσης ξεχωριστά παρουσιάζεται πιο κάτω.



εικόνα 3.1

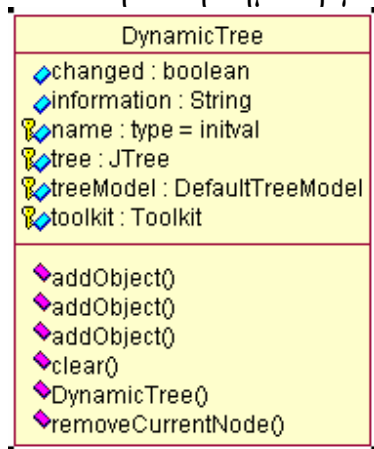
PdfManager

Η βασική κλάση για την επικοινωνία του προγράμματος με το χρήστη. Διαχειρίζεται όλες τις υπόλοιπες κλάσεις του package και σ' αυτήν βρίσκεται η `main()` για την εκτέλεση του προγράμματος.



DynamicTree

Η κλάση αυτή δημιουργεί και διαχειρίζεται το δέντρο των περιεχομένων .



EditSourceFrame

Ένα frame στο οποίο υπάρχουν τρεις επιλογές ανάλογα με το που βρισκόμαστε στο δέντρο των περιεχομένων.

- Εισαγωγή αρχείου που ήδη υπάρχει
- Δημιουργία αρχείου μέσω XML και XSL
- Εισαγωγή κάποιου υποκεφάλαιου

The screenshot displays the class hierarchy for `EditSourceFrame`. The left pane shows the class hierarchy, and the right pane shows the fields and methods.

Class Hierarchy (Left Pane):

- `EditSourceFrame`
 - `valid` : boolean
 - `back` : ImageIcon
 - `backbutton1` : JButton
 - `backbutton2` : JButton
 - `backbutton3` : JButton
 - `backbutton4` : JButton
 - `backbutton5` : JButton
 - `buttonpanel` : JPanel
 - `buttonpanel2` : JPanel
 - `buttonpanel3` : JPanel
 - `buttonpanel4` : JPanel
 - `buttonpanel5` : JPanel
 - `buttonpaneltitle` : JPanel
 - `chaptertitle` : JLabel
 - `commentcreate` : JTextField
 - `comentpanel` : JPanel
 - `comenttext` : JTextField
 - `contentpane` : Container
 - `createpanel` : JPanel
 - `createpdf` : JRadioButton
 - `f1` : Font
 - `fc` : JFileChooser
 - `file` : File
 - `findbutton` : JButton
 - `findlabel` : JLabel
 - `findpanel` : JPanel
 - `findpdf` : JRadioButton
 - `findsource` JButton
 - `findtext` : JLabel
 - `findtitle` : JLabel
 - `findxml` : JButton
 - `findxsl` : JButton
 - `finishbutton` : JButton
 - `finishbutton2` : JButton
 - `frame` : JFrame

Fields and Methods (Right Pane):

- `group` : ButtonGroup
- `helptext` : JTextField
- `labelname` : JLabel
- `labelsource` : JLabel
- `labelxml` : JLabel
- `labelxsl` : JLabel
- `list` : JComboBox
- `mainpanel` : JPanel
- `maintext` : JTextPane
- `name` : JLabel
- `nametext` : JTextField
- `next` : ImageIcon
- `nextbutton1` : JButton
- `nextbutton2` : JButton
- `nextbutton3` : JButton
- `nextbutton4` : JButton
- `node` : DefaultMutableTreeNode
- `okbutton` : JButton
- `open` : ImageIcon
- `page` : int
- `pagelabel` : JLabel
- `pagetext` : JTextField
- `pdffile` : File
- `ResPanel` : JPanel
- `ResPanel2` : JPanel
- `Results` : JTextField
- `Results2` : JTextField
- `source` : String
- `sourcepdffile` : String
- `textname` : JTextField
- `title` : String
- `titlebuttonback` : JButton
- `titlebuttonnext` : JButton
- `titlename` : JTextField
- `titlepanel` : JPanel
- `titlepdf` : JRadioButton
- `titletext` : JTextField
- `titletitle` : JTextField
- `userdir` : String
- `xmlfile` : File
- `xslfile` : File

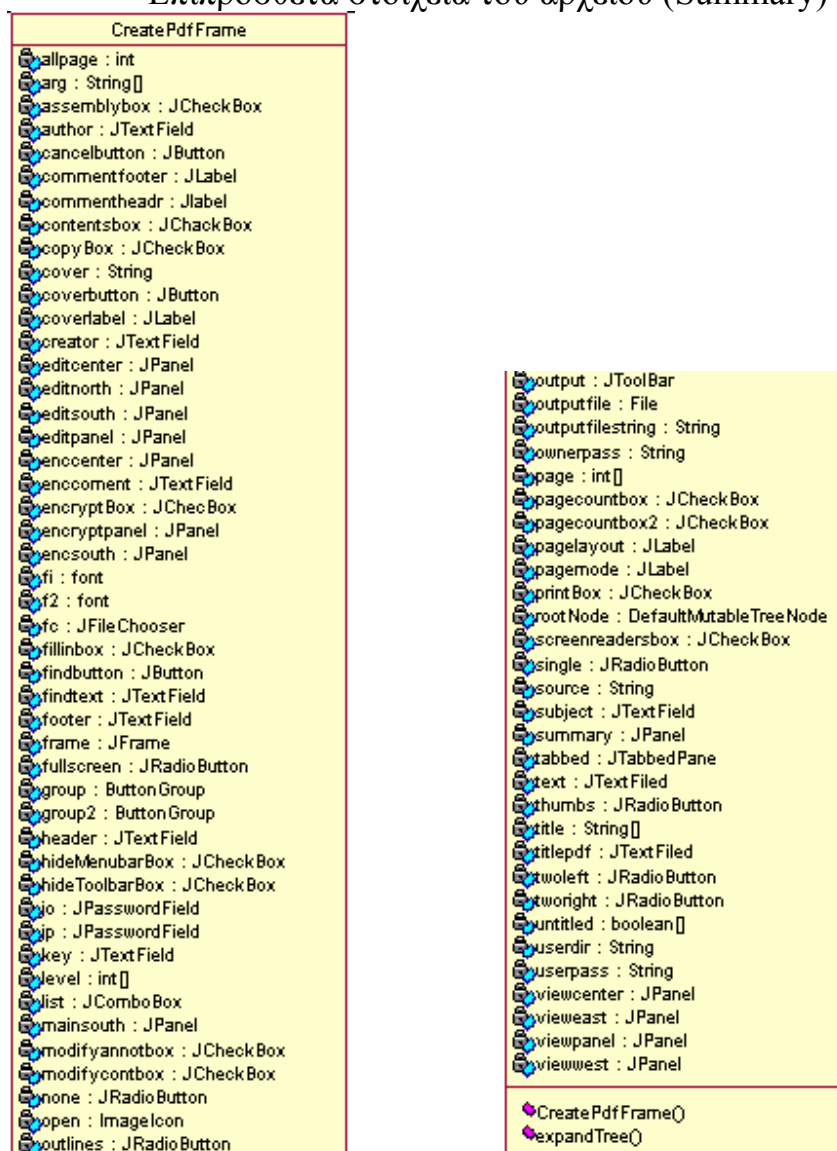
Methods:

- `EditSourceFrame()`
- `initStylesForTextPane()`

CreatePdfFrame

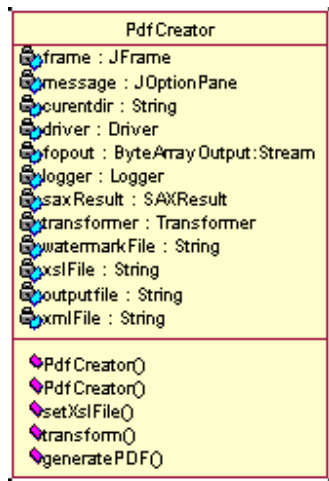
Ένα frame που έχει σαν σκοπό να εισάγει δεδομένα στο τελικό αρχείο που θέλουμε να δημιουργήσουμε :

- Που θα δημιουργηθεί (Source)
- Επιλογή εξώφυλλου
- Δημιουργία περιεχομένων
- Εισαγωγή Header- Footer
- Πως θα παρουσιάζεται (View Preferences)
- Απόκρυψη του αρχείου (Encryption)
- Επιπρόσθετα στοιχεία του αρχείου (Summary)



PdfCreator

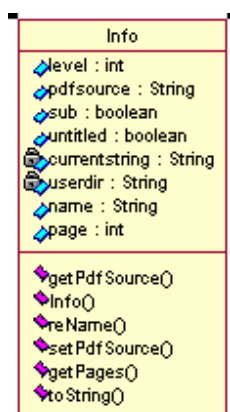
Σ' αυτή την κλάση γίνεται ο μετασχηματισμός των κατάλληλων XML/XSL και δημιουργούνται τα νέα αρχεία PDF.



Info

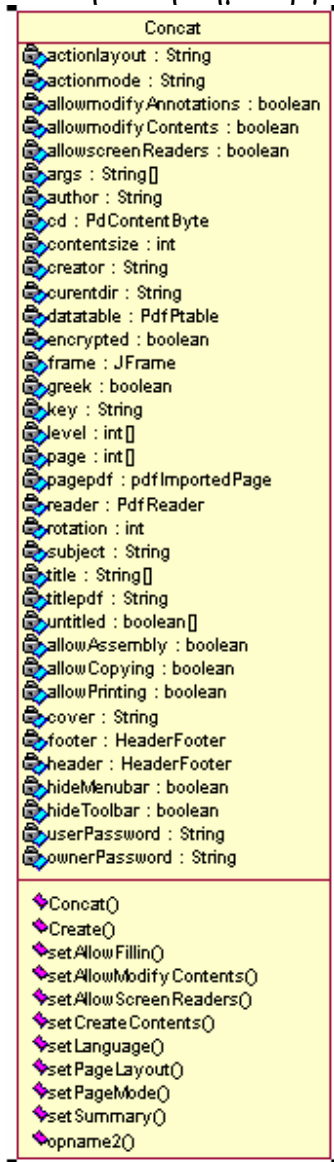
Η κλάση αυτή χρησιμοποιείται για να κρατάει πληροφορίες για τον κάθε κόμβο του δέντρου ξεχωριστά:

- Τίτλο (name)
- Πηγή του αρχείου (pdfsource)
- Αριθμό Σελίδων (page)
- Αν είναι υποκεφάλαιο (sub)
- Αν έχει επεξεργαστεί (untitled)
- Σε ποιο βάθος βρίσκεται στο δέντρο (level)



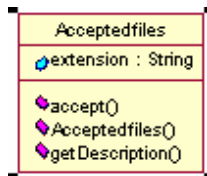
Concat

Είναι η κλάση που δημιουργεί το νέο αρχείο συνενώνοντας τα PDF που βρίσκονται στο δέντρο παίρνοντας δεδομένα από τις υπόλοιπες κλάσεις. Στην κλάση αυτή δημιουργούνται και τα περιεχόμενα του κειμένου.



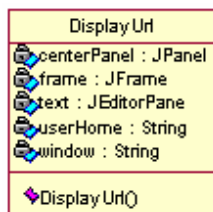
Acceptedfiles

Χρησιμοποιείται για την επιλογή files με μια συγκεκριμένη επέκταση .



DisplayUrl

Είναι ένα frame που βοηθά στην προβολή κάποιου URL



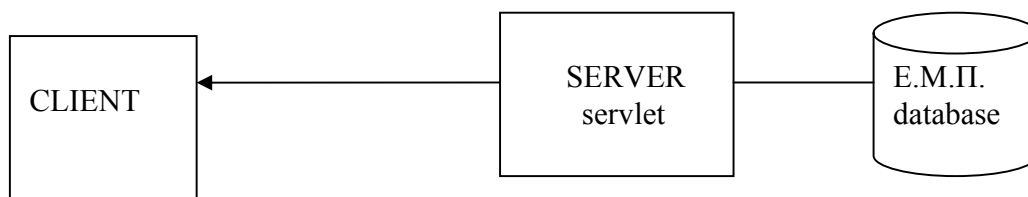
SplitPdf

Είναι ένα frame το οποίο παίρνει σαν είσοδο ένα αρχείο PDF και το διασπά σε όλες τις σελίδες του .Αυτό έγινε για να μπορούμε να διαχειριζόμαστε μεγάλα αρχεία PDF πιο εύκολα .Δεν συμμετέχει άμεσα στο τελικό αποτέλεσμα αλλά βοηθά στην καλύτερη διαχείριση των αρχείων PDF που έχουμε .



3.3 updatePDFM

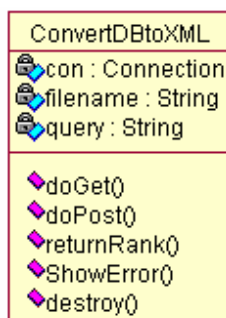
Όπως ανάφερα και στην εισαγωγή το updatePDFM είναι ένα πακέτο το οποίο περιέχει ένα Servlet ,το ConvertDBtoXML.java , για την ανταλλαγή πληροφοριών μεταξύ του χρήστη (client) και της βάσης δεδομένων του Ε.Μ.Π. μέσω XML αρχείων (εικόνα 3.1). Το Interface με τον χρήστη αποτελείται από μια HTML σελίδα που στέλνει τις κατάλληλες πληροφορίες στο Servlet για να γίνει η σύνδεση με την δικτυακή βάση .Οι διαδικασίες που μεσολαβούν παρουσιάζονται αναλυτικά παρακάτω .



εικόνα 3.1

Στην συγκεκριμένη εφαρμογή η doGet (χρησιμοποιείται για να πάρουμε την XML μορφή της βάσης) παίρνει σαν παραμέτρους απο την HTML σελίδα :

- Driver της βάσης που θέλουμε να συνδεθούμε.(Επειδή η βάση του Ε.Μ.Π. είναι η mySql έχουμε σαν default τιμή τον Driver της mySql com.mysql.jdbc.Driver)
- Το Url της βάσης που θέλουμε να συνδεθούμε (default τιμή δόθηκε το Url της βάσης του Ε.Μ.Π. jdbc:mysql://webdev.ece.ntua.gr/ece_dev)
- Username
- Password
- Το αρχείο που θέλουμε να κατεβάσουμε



Οι πληροφορίες που παίρνουμε για τα αρχεία δημιουργούνται μέσω queries στην βάση δεδομένων μας .

Πρέπει να αναφερθεί ότι είναι αναγκαία η προσθήκη επιπλέον πληροφορίας που δεν ήταν δυνατόν να πάρουμε απο την βάση του Ε.Μ.Π. γιατί απλά δεν υπάρχει .Είναι απαραίτητη όμως για την δημιουργία των κεφαλαίων του οδηγού σπουδών του τμήματος γι` αυτό γίνεται αυτή την παρέμβαση . Περισσότερες πληροφορίες υπάρχουν μέσα στο PdfM directory . Ακολουθεί μια σύντομη περιγραφή της δομής του κάθε αρχείου XML (χωρίς την επιπλέον πληροφορία) που χρησιμοποιείται για την δημιουργία των κεφαλαίων του οδηγού σπουδών.

✓ *Chap5.xml*

Χρησιμοποιείται μαζί με το Chap5.xsl για την δημιουργία του κεφαλαίου 5 που παρουσιάζει το προσωπικό της σχολής .Στόχος μας είναι να πάρουμε πληροφορίες για τους καθηγητές της σχολής.Το αρχείο κατατάσσει τους καθηγητές ανά τομέα (Division) και ακολούθως κατά βαθμό(Professor,Assistant,Associate,Lecturer,Emeritus).Η πληροφορία για τον κάθε καθηγητή , (όνομα , επίθετο)βρίσκεται στο <row> .

```
<database >
  <database_data>
    <Division1>Τομέας
      <Professor>Βαθμός καθηγητών
        <row>
          <NameGR><![CDATA[Ιωάννης]]></NameGR>Όνομα
          <SurnameGR><![CDATA[Βομβορίδης]]></SurnameGR>Επίθετο
        </row>
        .....
      </Professor>
      <Assistant>
        <row>
          .....
        </row>
      </Assistant>
      <Associate>
        <row>
          .....
        </row>
      </Associate>
      <Lecturer>
        .....
```

```

        </Lecturer>
        <Emeritus>
        .....
        </Emeritus>
    </Division1>
    <Division2>
    .....
    </Division7>

```

```

</database_data>
</database>

```

✓ Chap7-9.xml

Χρησιμοποιείται για την δημιουργία του κεφαλαίου 7 (Πρόγραμμα Κορμού) και του κεφαλαίου 9 (Περιγραφή Μαθημάτων - Πρόγραμμα Κορμού) με την βοήθεια των Chap7.xsl και Chap9.xsl αντίστοιχα. Έχουμε σαν στόχο να πάρουμε πληροφορίες για τα μαθήματα της σχολής απο το 1^ο μέχρι το 5^ο εξάμηνο .Το αρχείο κατατάσσει τα μαθήματα ανά εξάμηνο και η πληροφορία για το κάθε μάθημα βρίσκεται στο <row> .

```

<database>
  <database_data>
    <semester_1>Εξάμηνο
    <row>
      <CourseID_1><![CDATA[27]]></CourseID_1>Κλειδί
      <Code_2><![CDATA[9.2.04.1]]></Code_2>Κωδικός
      <TitleGR_3><![CDATA[Γραμμική Άλγεβρα]]></TitleGR_3>Τίτλος
      <LectureHours_5><![CDATA[5]]></LectureHours_5>Διδασκ. Ώρες
      <LabHours_6><![CDATA[0]]></LabHours_6>Εργαστ. Ώρες
      <TitleGR_7><![CDATA[Υποχρεωτικό]]></TitleGR_7>Κατηγορία
      <DescriptionGR_8><![CDATA[Διανυσματικός]]></DescriptionGR_8>Περιγραφή
    </row>
    .....
  </semester_1>
  <semester_2>
  .....
  </semester_5>

```

Για να βρούμε τους καθηγητές που διδάσκουν το κάθε μάθημα χρησιμοποιούμε ξεχωριστή δομή μέσα στο αρχείο.

```

<Professors>

```

Βρίσκουμε το μάθημα που διδάσκει ο κάθε καθηγητής με το CourseID attribute του row και η πληροφορία για τον κάθε καθηγητή ξεχωριστά βρίσκεται μέσα στο <professor>.

```
<row CourseID="8">
  <professor>
    <NameGR><![CDATA[Νικόλαος]]></NameGR>Όνομα
    <SurnameGR><![CDATA[Μήτρου]]></SurnameGR>Επίθετο
  </professor>
</row>
<row CourseID="10">
  <professor>
    <NameGR><![CDATA[Εμμανουήλ]]></NameGR>
    <SurnameGR><![CDATA[Πρωτονοτάριος]]></SurnameGR>
  </professor>
.....
</Professors>
</database_data>
</database>
```

✓ Chap8-10.xml

Χρησιμοποιείται για την δημιουργία του κεφαλαίου 8 (Πρόγραμμα Ροών) και του κεφαλαίου 9 (Περιγραφή Μαθημάτων - Πρόγραμμα Ροών) με την βοήθεια των Chap8.xsl και Chap10.xsl αντίστοιχα.

Η δομή του αρχείου είναι παρόμοια με την δομή του Chap7-9.xml με την διαφορά ότι τα μαθήματα είναι απο το 6^ο μέχρι το 9^ο εξάμηνο και τα κατατάσσει σύμφωνα με την ροή τους .

```
<database>
<database_data>
<flow_1>Ροή
<row>
  <CourseID_1><![CDATA[66]]></CourseID_1>Κλειδί
  <Code_2><![CDATA[3.4.12.6]]></Code_2>Κωδικός
  <TitleGR_3><![CDATA[Αρχιτεκτονική Υπολογιστών]]></TitleGR_3>Τίτλος
  <Semester_4><![CDATA[6]]></Semester_4>Εξάμηνο
  <LectureHours_5><![CDATA[4]]></LectureHours_5>Διδάσκ. Ωρες
  <LabHours_6><![CDATA[0]]></LabHours_6>Εργαστ. Ωρες
  <TitleGR_7><![CDATA[Υποχρεωτικό]]></TitleGR_7>Κατηγορία
  <DescriptionGR_8><![CDATA[Αλγόριθμοι.]]></DescriptionGR_8>Περιγραφή
</row>
.....
</row>
</Professors>
```

Και για τους καθηγητές έχουμε την ίδια δομή με το Chap7-9.xml.

```
<row CourseID="8">
  <professor>
    <NameGR><![CDATA[Νικόλαος]]></NameGR>Όνομα
    <SurnameGR><![CDATA[Μήτρου]]></SurnameGR>Επίθετο
  </professor>
</row>
<row CourseID="10">
  <professor>
    <NameGR><![CDATA[Εμμανουήλ]]></NameGR>
    <SurnameGR><![CDATA[Πρωτονοτάριος]]></SurnameGR>
  </professor>
.....
</Professors>
</database_data>
</database>
```

3.4 Δημιουργία XSL-FO

Σαν πρακτική εφαρμογή έχω δημιουργήσει XSL αρχεία που με τον συνδυασμό των κατάλληλων XML που παίρνουμε από την δικτυακή βάση του Ε.Μ.Π. έχουν σαν αποτέλεσμα XSL-FO αρχεία .Ακολουθώντας με την βοήθεια του FOP API έχουμε σαν αποτέλεσμα PDF αρχεία .Για να γίνει πιο κατανοητός ο τρόπος που γίνεται ο μετασχηματισμός θα παρουσιάσω μερικές κρίσιμες λειτουργίες για το FOP και για τα XSL αρχεία .

✓ XSL αρχεία

Ένα XSLT φύλλο στυλ περιέχει ένα σύνολο από κανόνες για το μετασχηματισμό ενός πηγαίου δέντρου , XML στην συγκεκριμένη περίπτωση , σε ένα άλλο . Ο μετασχηματισμός επιτυγχάνεται μέσα από μια ομάδα από κανόνες προτύπου (templates rules) . Ένας κανόνας προτύπου περιέχει δυο συστατικά το pattern και το content constructor . Το pattern επιλέγει κόμβους από το πηγαίο δέντρο και στη συνέχεια καλεί τον content constructor ο οποίος εφαρμόζεται στους κόμβους που έχουν επιλεγεί και παράγει μέρος του τελικού δέντρου. Ο επεξεργαστής XSLT διατρέχει όλο το δέντρο του XML εγγράφου διαβάζοντας όλους τους κόμβους με τη σειρά . Κάθε φορά που διαβάζεται ένας κόμβος , ο επεξεργαστής τον συγκρίνει με το pattern από όλους τους κανόνες

προτύπου (template rules) στο φύλλο στυλ. Όταν ο επεξεργαστής βρει έναν κόμβο που ταιριάζει με το template ενός template rule τότε ο content constructor του template rule εφαρμόζεται στον κόμβο αυτόν. Ανάλογα με το πώς θέλουμε να παρουσιάζονται τα δεδομένα στο PDF που πρόκειται να δημιουργηθεί τροποποιούμε και τον content constructor του XSL-FO αρχείου.

Για να γίνει πιο κατανοητό ακολουθεί ένα μέρος του Chap5.xsl αρχείου με τις πιο σημαντικές λειτουργίες του. Το XSL αυτό με τον συνδυασμό του Chap5.xml που ορίσαμε πιο πάνω δημιουργείτε το κεφάλαιο 5 του οδηγού σπουδών.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" id="IDA5J4JH" >
```

Ο χώρος ονοματοδοσίας XSLT που χρησιμοποιείται για να αναγνωρίζει στοιχεία, ιδιότητες και άλλα ονόματα τα οποία έχουν κάποια ιδιαίτερη σημασία που καθορίζεται μέσα από τις προδιαγραφές του συγκεκριμένου χώρου ονοματοδοσίας.

```
<xsl:template match="/database">
```

Ο επεξεργαστής ψάχνει τον κόμβο database (η ρίζα του Chap5.xml) με το xsl:template για να εφαρμόσει τον content constructor του XSL-FO πιο κάτω.

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" font-family="TimesNewRoman" font-style="normal" font-weight="normal" font-size="12pt">
```

Ορίζεται η ρίζα του (fo:root) XSL-FO αρχείου που πρόκειται να δημιουργηθεί με διάφορες άλλες δηλώσεις για το δέντρο του όπως ονοματοδοσία και γραμματοσειρά.

```
    <fo:layout-master-set>
      <fo:simple-page-master master-name="simple" page-
height="29.7cm" page-width="21cm" margin-top="1.5cm" margin-bottom="2.5cm" margin-
left="2.5cm" margin-right="2.5cm">
        <fo:region-body margin-top="1.5cm"/>
        <fo:region-before extent="1.5cm"/>
        <fo:region-after extent="1.5cm"/>
      </fo:simple-page-master>
    </fo:layout-master-set>
```

Το στοιχείο fo:layout-master-set περιέχει δηλώσεις για την παρουσίαση των σελίδων όπως τα περιθώρια, το μέγεθος και η περιοχή για την εισαγωγή header και footer.

```
<fo:page-sequence master-reference="simple">
  <fo:flow flow-name="xsl-region-body">
```

Στο στοιχείο αυτό ορίζεται το περιεχόμενο της κάθε σελίδας ξεχωριστά

```
<fo:block font-family="ArialBold" font-size="14pt">
  5. ΠΡΟΣΩΠΙΚΟ ΤΗΣ ΣΧΟΛΗΣ
</fo:block>
```

Με το fo:block «γράφουμε» στην συγκεκριμένη σελίδα με τις κατάλληλες γραμματοσειρές

```
<fo:block line-height="1.5">
  <xsl:value-of select="addinfo/president"/>
</fo:block>
```

Το στοιχείο xsl:value-of υπολογίζει το περιεχόμενο από κάποιο κόμβο και την αντιγράφει στο παραγόμενο έγγραφο . Η ιδιότητα select του στοιχείου xsl:value-of προσδιορίζει ακριβώς τον κόμβο του οποίου η τιμή υπολογίζεται .Στο συγκεκριμένο παράδειγμα επιλέγεται ο κόμβος που βρίσκεται στην «διεύθυνση» database/addinfo/president και γράφουμε το περιεχόμενο του με την βοήθεια του fo:block .

```
<fo:table text-align="left">
  <fo:table-column column-width="4in" padding="6pt"/>
  <fo:table-column column-width="4in" padding="6pt"/>
  <fo:table-body>
```

Για να εισάγουμε δεδομένα με κάποια συγκεκριμένη μορφή χρειάζεται η δημιουργία ενός πίνακα .Αυτό επιτυγχάνεται με το fo:table .Ακολουθώς δημιουργείται ένας πίνακας με δύο στήλες (fo:table-column) και με τις απαραίτητες δηλώσεις .

```
<xsl:apply-templates select="//Division1"/>
```

Για να προχωρήσουμε παραπέρα από τον κόμβο ρίζα (database) πρέπει να πούμε στον επεξεργαστή XSLT να προχωρήσει και στα παιδιά του κόμβου ρίζα . Για να το πετύχουμε αυτό θα πρέπει αναδρομικά να επισκεφτούμε όλους τους

κόμβους του αρχικού δέντρου (δηλαδή του εισερχόμενου εγγράφου) , αυτό μπορούμε να το πετύχουμε με το στοιχείο `xsl:apply-templates`.

Συνεχίζοντας επισκεπτόμαστε όλους τους κόμβους του XML αρχείου που θέλουμε να εξάγουμε πληροφορία και καθορίζουμε πως θέλουμε να γίνεται η παρουσίαση με κατάλληλες εντολές του XSL-FO .

Με αυτό τον τρόπο έγινε η δημιουργία όλων των XSL αρχείων που παράγουν τα XSL-FO αρχεία .

- ✓ Αρχικοποίηση του Fop Driver μέσα στην κλάση PdfCreator για PDF αρχεία

Δημιουργούμε ένα νέο constructor της κλάσης Driver() του FOP και την αρχικοποιούμε για να έχει έξοδο PDF αρχεία.

```
Driver driver = new Driver();
ByteArrayOutputStream fopout = new ByteArrayOutputStream();
driver.setOutputStream(fopout);
//Set Pdf output
driver.setRenderer(Driver.RENDER_PDF);
org.apache.avalon.framework.logger.ConsoleLogger logger =
    new org.apache.avalon.framework.logger.ConsoleLogger(
        org.apache.avalon.framework.logger.ConsoleLogger.LEVEL_FATAL);
org.apache.fop.messaging.MessageHandler.setScreenLogger(logger);
driver.setLogger(logger);
```

- ✓ Μετασχηματισμός των XML /XSL αρχείων και δημιουργία του PDF μέσα στην κλάση PdfCreator

Για να γίνει ο μετασχηματισμός πρώτα δημιουργούμε ένα μετασχηματιστή για το XSL αρχείο .

```
Transformer transformer = TransformerFactory.newInstance().
newTransformer( new StreamSource(new File(xslFile)));
```

Στην συνέχεια μετασχηματίζουμε το XML αρχείο και σαν αποτέλεσμα (result) με την βοήθεια του `org.apache.fop.apps.Driver` παίρνουμε το PDF.

```
transformer.transform(new StreamSource(new File(xmlFile)),
new SAXResult(driver.getContentHandler()));
```

- ✓ Χρησιμοποίηση και του iText API στην κλάση PdfCreator

Για να εμφανιστεί το κείμενο που δημιουργείται με τον μετασχηματισμό χρησιμοποιήσαμε το iText API .Με αυτό τον τρόπο μπορούμε να εκμεταλλευτούμε τις δυνατότητες που μας παρέχει ένα PDF αρχείο. Προς το παρόν μας εμφανίζει μόνο τον αριθμό των σελίδων του κειμένου γιατί έχουμε την κλάση CreatePdfFrame για να εισάγει τα meta-data(δεδομένα που σχετίζονται με τον τρόπο παρουσίασης ,απόκρυψη κ.τ.λ.). Μπορεί όμως να επεκταθεί σε μια άλλη έκδοση του PdfM ή με κατάλληλη τροποποίηση να τρέχει και σαν μια ανεξάρτητη εφαρμογή .Αυτός ήταν και ο βασικός λόγος που χρησιμοποιήθηκε το συγκεκριμένο API , η μελλοντική αναβάθμιση του κώδικα.

✓ Προσθήκη νέων Γραμματοσειρών (fonts)

Το FOP υποστηρίζει τις βασικές γραμματοσειρές του Adobe PDF (Helvetica, Times, Courier, Symbol, ZapfDingbats) . Αυτές όμως δεν υποστηρίζουν Ελληνικούς χαρακτήρες και η προσθήκη νέων ήταν αναγκαία.

Για να μπορέσει το FOP να υποστηρίξει τις νέες γραμματοσειρές που πρόκειται να προσθέσουμε χρησιμοποιεί ένα αρχείο XML όπου περιέχει μετρικά στοιχεία (font-metrics) για την κάθε γραμματοσειρά ξεχωριστά .Για την δημιουργία του XML αρχείου το FOP API περιέχει μια κλάση ,την org.apache.fop.fonts.apps.TTFReader που μπορεί να το παράξει με την εντολή:

```
java -cp %LOCALCLASSPATH% org.apache.fop.fonts.apps.TTFReader -fn fontpath.ttf fontpath.ttf XMLname.xml
```

Για να γίνει πιο εύκολο δημιουργήθηκε ένα MS-DOS Batch αρχείο (fontrender.bat)που είναι στο directory του PdfM το οποίο τοποθετεί τις κατάλληλες βιβλιοθήκες στο LOCALCLASSPATH και παράγει τα XML αρχεία .Απλά κάθε φορά χρειάζεται να αλλάζει το όνομα της γραμματοσειράς (fontpath)που θέλουμε να προσθέσουμε και του XML (XMLname).

Στην συνέχεια για να δηλωθούν όλες οι νέες γραμματοσειρές μέσα στο FOP είναι απαραίτητη η δημιουργία ενός νέου XML ,με την ονομασία userconfig.xml το οποίο θα «κρατά» πληροφορίες για τις γραμματοσειρές και θα έχει την παρακάτω δομή:

```
<configuration>  
<fonts>
```

```
.....
```

```
<font metrics-file="XMLPath" kerning="yes" embed-file="FontPath">
```

```
<font-triplet name="FontName" style="normal" weight="normal" />
```

```

<font-triplet name="FontName" style="italic" weight="bold" />
<font-triplet name="FontName" style="normal" weight="bold" />
<font-triplet name="FontName" style="italic" weight="normal" />
</font>
. ....
</fonts>
</configuration>

```

Έπειτα για να δηλωθεί το userconfig.xml αρχείο στην κλάση PdfCreator χρησιμοποιούμε την εντολή :

```

try
{
org.apache.fop.apps.Options options = new Options(new
File(curentdir+System.getProperty("file.separator")+
"userconfig.xml"));
}
catch (Exception ex){System.err.print(ex.getMessage());}

```

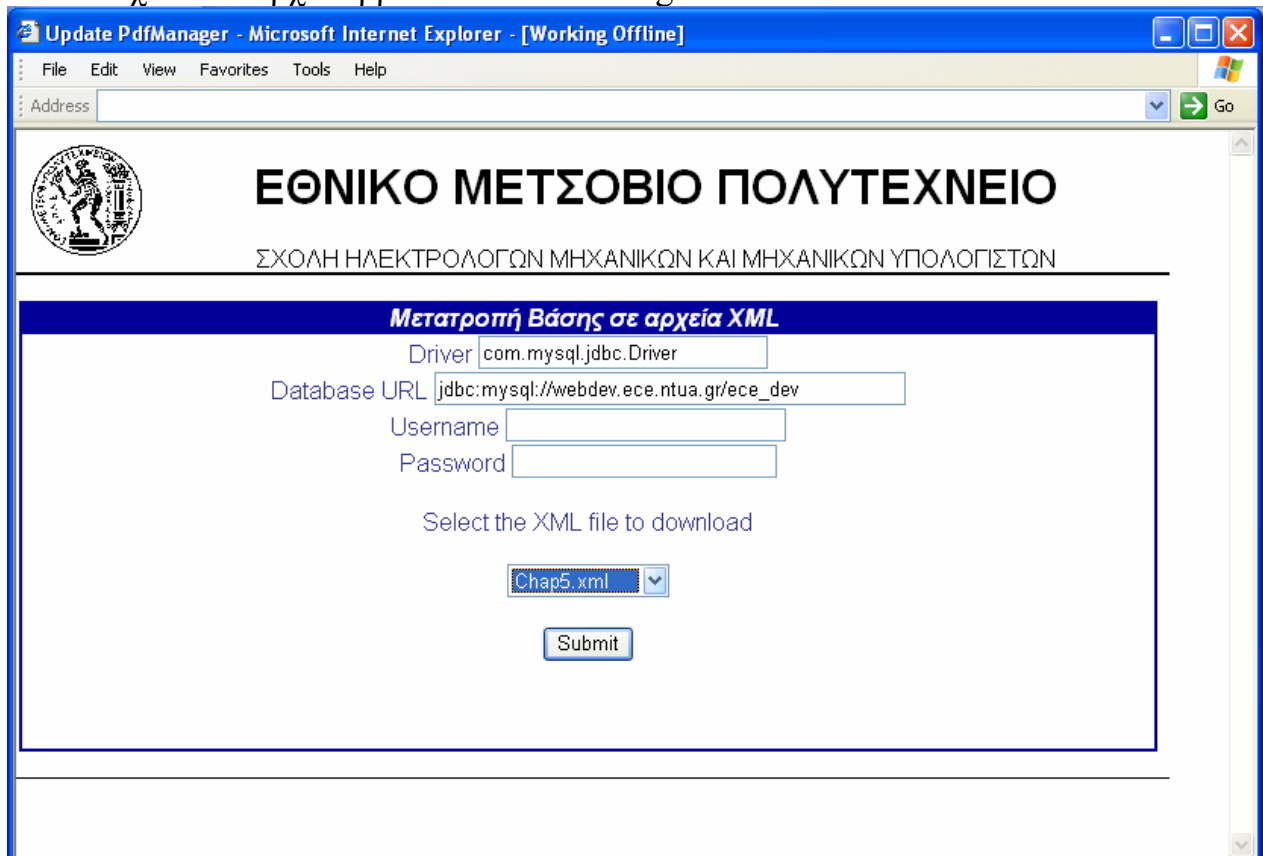
Όλα τα XML αρχεία μαζί με τις γραμματοσειρές βρίσκονται στο directory \conf της εφαρμογής.

4. ΛΕΙΤΟΥΡΓΙΚΟ ΣΕΝΑΡΙΟ

Για να φανούν στην πράξη τόσο οι δυνατότητες του συστήματος όσο και ο τρόπος λειτουργίας του ακολουθεί μία σύντομη επίδειξη του .Ακολουθώντας σαν πρακτική εφαρμογή δημιουργήθηκε ο οδηγός σπουδών του τμήματος Ηλεκτρολόγων Μηχανικών του Ε.Μ.Π. Μερικά απο τα κεφάλαια του οδηγού θα δημιουργηθούν δυναμικά και τα υπόλοιπα θα τα πάρουμε σαν στατικό περιεχόμενο .

4.1 Download των XML αρχείων απο την δικτυακή βάση του Ε.Μ.Π.


Για να δημιουργήσουμε δυναμικά τα PDF αρχεία, που στην ουσία είναι κεφάλαια απο τον οδηγό σπουδών, πρέπει πρώτα να κατεβάσουμε τα κατάλληλα XML μέσω του Java Servlet (updatepdfm.ConvertDBtoXML) που έχουμε στην διεύθυνση <http://localhost:8080/index.html> (εικόνα 4.1) .Τα αντίστοιχα XSL αρχεία βρίσκονται στο \odhgos\xsl .



Update PdfManager - Microsoft Internet Explorer - [Working Offline]

File Edit View Favorites Tools Help

Address Go

 **ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Μετατροπή Βάσης σε αρχεία XML

Driver

Database URL

Username

Password

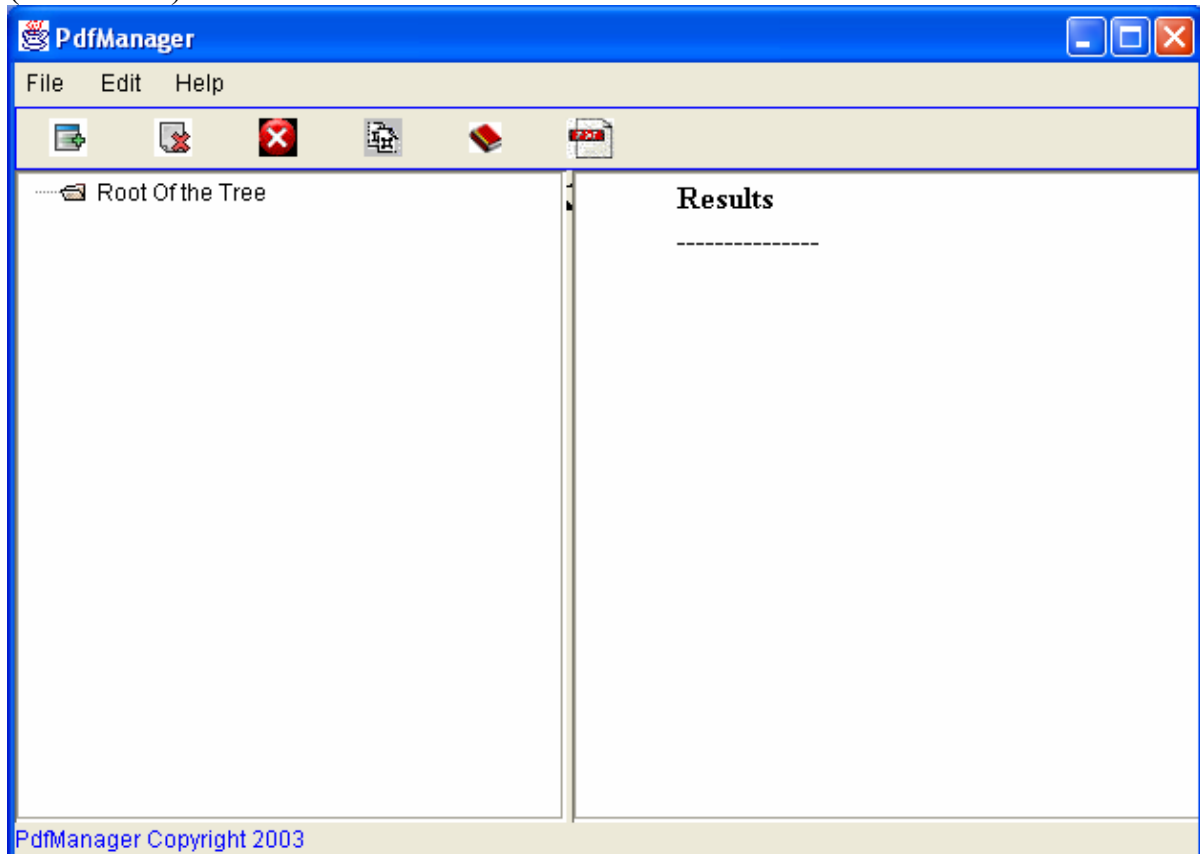
Select the XML file to download

εικόνα 4.1

Αφού εισάγουμε username και password μπορούμε να επιλέξουμε ποιο αρχείο θέλουμε να κατεβάσουμε .

4.2 Λειτουργία PdfManager

Ξεκινώντας την εφαρμογή εμφανίζεται στην οθόνη το κυρίως μέρος της (εικόνα 4.2).



εικόνα 4.2

Δίνεται η δυνατότητα στον χρήστη οργανώνοντας τα περιεχόμενα του αρχείου που θέλουμε να παράξουμε να δημιουργείται και το επιθυμητό αρχείο .Τα περιεχόμενα είναι διατεταγμένα σε μορφή δέντρου και ο κάθε κόμβος αντιστοιχεί σε ένα κεφάλαιο ή υποκεφάλαιο .

- Κάθε **κεφάλαιο** αντιστοιχεί και σε ένα ξεχωριστό αρχείο PDF το οποίο είτε δημιουργούμε δυναμικά είτε προϋπάρχει και το χρησιμοποιούμε .Οι κόμβοι που είναι παιδιά της ρίζας είναι κεφάλαια.
- Με την προσθήκη **υποκεφάλαιων** μπορούμε να οργανώσουμε καλύτερα τις σελίδες των περιεχομένων . Δεν αντιστοιχούν σε κάποιο αρχείο απλά φέρουν κάποιο τίτλο και τον αριθμό της

σελίδας που αντιστοιχεί . Οι κόμβοι που δεν είναι παιδιά της ρίζας είναι υποκεφάλαια .

Το πρόγραμμα παίρνει τα αρχεία που εισάγει ο χρήστης και τα συνενώνει με την σειρά που βρίσκονται στο δέντρο .Σαν αποτέλεσμα έχουμε ένα ξεχωριστό αρχείο που σ` αυτό μπορεί να εισάγουμε διάφορες επιπλέον δυνατότητες που προσφέρουν τα PDF όπως απόκρυψη, ορισμό διαδρομών ανάγνωσης ,ηλεκτρονικής υπογραφής κ.α.

Στα αριστερά της οθόνης (όπως φαίνεται στην εικόνα 4.2) μπορούμε να δούμε την δομή του δέντρου που δημιουργείται και στα δεξιά το περιεχόμενο του κόμβου που επιλέγουμε κάθε φορά .Στο πάνω μέρος της οθόνης βρίσκεται το menu επιλογών και τα εργαλεία της εφαρμογής.

Ακολουθεί μια σύντομη περιγραφή του κάθε εργαλείου με τα σχετικά παραδείγματα .

- ✓  Add - Προσθήκη κεφαλαίου ή υποκεφαλαίου

Επιλέγοντας ένα κόμβο η εντολή Add προσθέτει ένα καινούργιο κόμβο στο δέντρο που είναι παιδί του επιλεγμένου κόμβου . Αρχικά τα κεφάλαια ή υποκεφάλαια που εισάγονται στο δέντρο δεν φέρουν κάποια πληροφορία (untitled) .Έτσι ανάλογα με το πού βρίσκεται ο επιλεγμένος κόμβος έχουμε προσθήκη κεφαλαίου ή υποκεφαλαίου .

Προσθήκη κεφαλαίου : Για την προσθήκη κεφαλαίου πρέπει να επιλέξουμε την ρίζα του δέντρου .Έτσι ο νέος κόμβος θα είναι παιδί της ρίζας και κατά συνέπεια κεφάλαιο .

Προσθήκη υποκεφαλαίου : Για να μπορούμε να εισάγουμε ένα υποκεφάλαιο σε κάποιο κόμβο (κεφάλαιο ή άλλο υποκεφάλαιο) πρέπει ο επιλεγμένος να μην είναι untitled ,να φέρει δηλαδή πληροφορία , αλλιώς εμφανίζεται μήνυμα λάθους .

- ✓  Remove - Διαγραφή κεφαλαίου ή υποκεφαλαίου

Διαγράφει τον κόμβο που επιλέγεται.

- ✓  Delete - Διαγράφει όλο το δέντρο

Γίνεται η διαγραφή όλου του δέντρου που δημιουργήθηκε.

- ✓  Split Διαχωρίζει κάποιο αρχείο

Η εντολή αυτή δεν παίζει κάποιο λειτουργικό ρόλο στην εφαρμογή .Η προσθήκη της όμως ήταν αναγκαία για να γίνει πιο εύκολη η διαχείριση μεγάλων PDF αρχείων .Με την εντολή αυτή επιλέγουμε ένα αρχείο και διαχωρίζεται σε όλες τις σελίδες του .

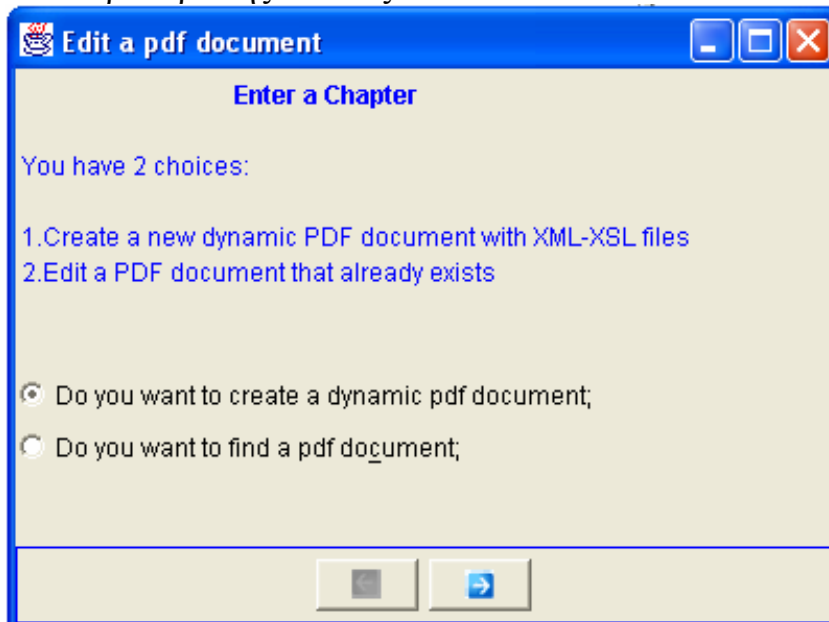
Για παράδειγμα αν επιλέξουμε ένα αρχείο με 60 σελίδες τότε θα έχουμε σαν αποτέλεσμα 60 νέα αρχεία με 1 σελίδα το κάθε ένα που δημιουργείται σε directory της επιλογής μας .Στην συνέχεια με την βοήθεια του PdfM μπορούμε να συνενώσουμε τις σελίδες που θέλουμε και να δημιουργήσουμε ένα πιο μικρό αρχείο που να ταιριάζει στις απαιτήσεις μας.

- ✓  Edit - Προσθήκη πληροφορίας σε κεφάλαιο ή υποκεφάλαιο

Επιλέγοντας ένα κόμβο μπορούμε να εισάγουμε πληροφορίες σχετικά με το κεφάλαιο ή το υποκεφάλαιο που θέλουμε να αντιπροσωπεί .Ανάλογα με το πού βρίσκεται ο κόμβος εμφανίζεται και το αντίστοιχο παράθυρο .

Προσθήκη πληροφορίας σε κεφάλαιο.

Αν βρισκόμαστε σε κόμβο που αντιστοιχεί σε κεφάλαιο (παιδί της ρίζας) εμφανίζεται το παράθυρο της εικόνας 4.3.



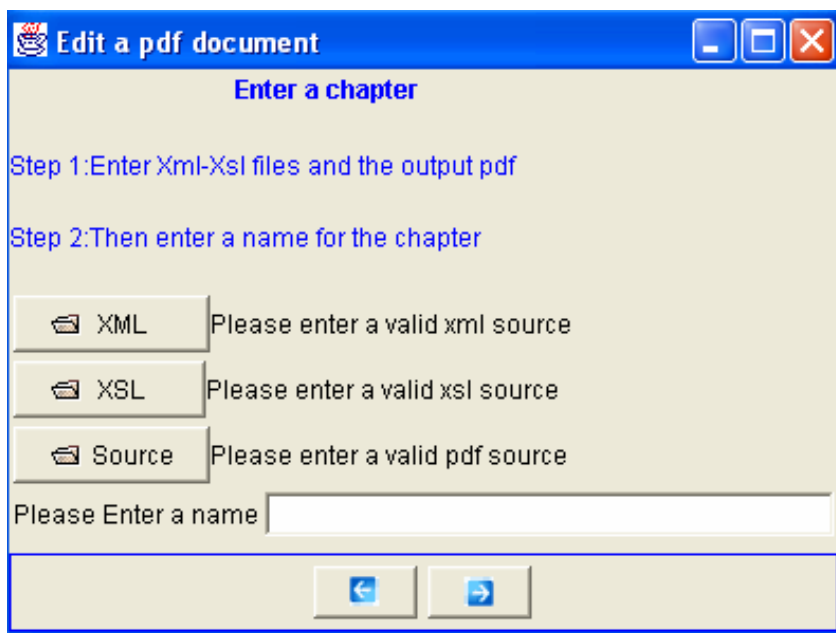
εικόνα 4.3

Υπάρχουν δύο επιλογές :

- Δημιουργία ενός δυναμικού αρχείου με τα κατάλληλα XML/XSL αρχεία.

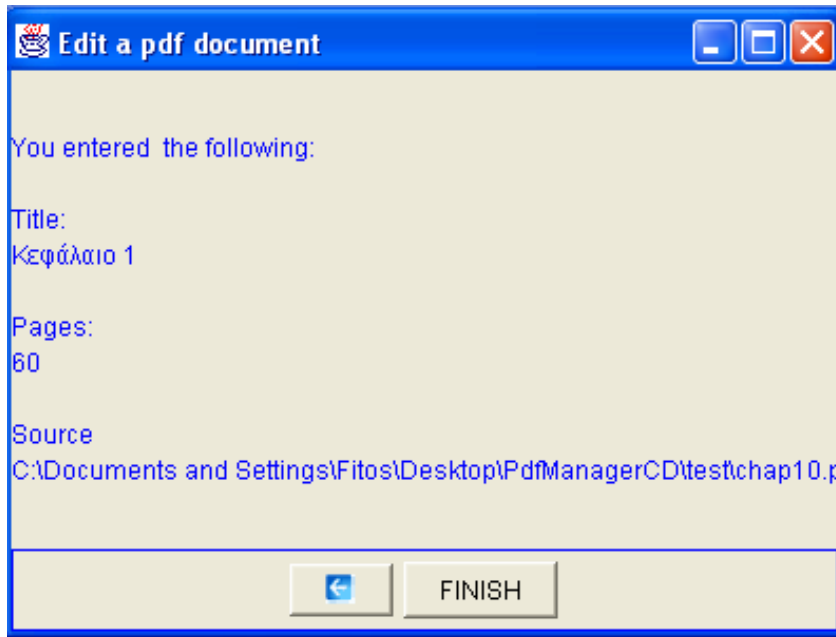
Με αυτή την επιλογή μπορούμε να δημιουργήσουμε ένα νέο Pdf αρχείο με τον μετασχηματισμό κατάλληλων XML/XSL αρχείων .

Συνεχίζοντας με αυτή την επιλογή εμφανίζεται το παράθυρο της εικόνας 4.4.



εικόνα 4.4

Πρώτα εισάγουμε τα κατάλληλα XML/XSL αρχεία και καθορίζουμε που θα δημιουργηθεί το νέο αρχείο(Source) .Έπειτα δηλώνουμε τον τίτλο που θα έχει το κεφάλαιο στον πίνακα τον περιεχομένων .Αν όλοι οι παράμετροι που θέσαμε είναι έγκυροι δημιουργείται το αρχείο και εμφανίζεται το επόμενο παράθυρο της εικόνας 4.5 .



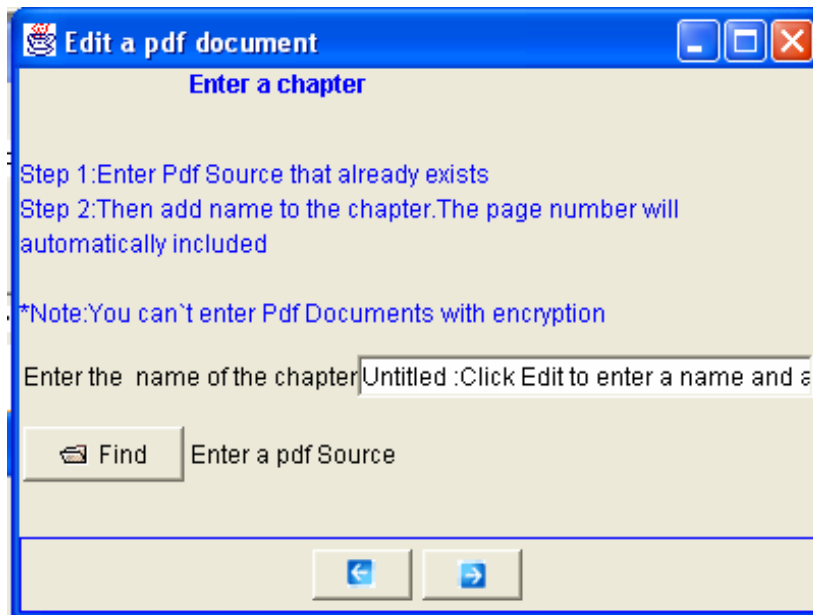
εικόνα 4.5

Εδώ εμφανίζονται οι πληροφορίες που θα έχει το νέο κεφάλαιο .Ο αριθμός των σελίδων εισάγεται αυτόματα απο το πρόγραμμα .Πατώντας finish εισάγονται οι πληροφορίες στο κεφάλαιο και εμφανίζονται στο δέντρο.

- Η χρησιμοποίηση κάποιου αρχείου PDF που ήδη υπάρχει .

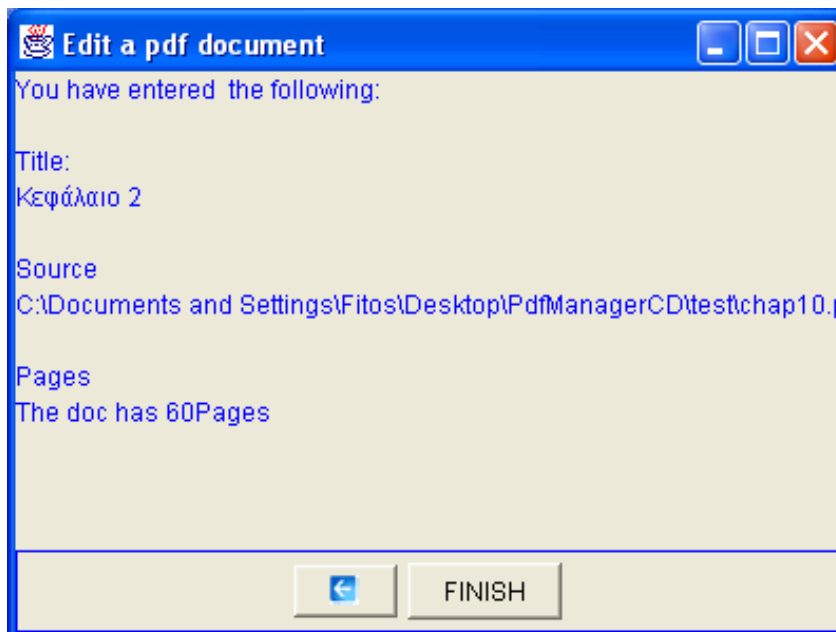
Με αυτή την επιλογή χρησιμοποιείται ένα αρχείο που προϋπάρχει .Αξίζει να επισημάνουμε ότι δεν μπορούμε να χρησιμοποιήσουμε αρχεία PDF που έχουν απόκρυψη (encrypted).

Το παράθυρο που αντιστοιχεί σ` αυτή την επιλογή φαίνεται στην εικόνα 4.6



εικόνα 4.6

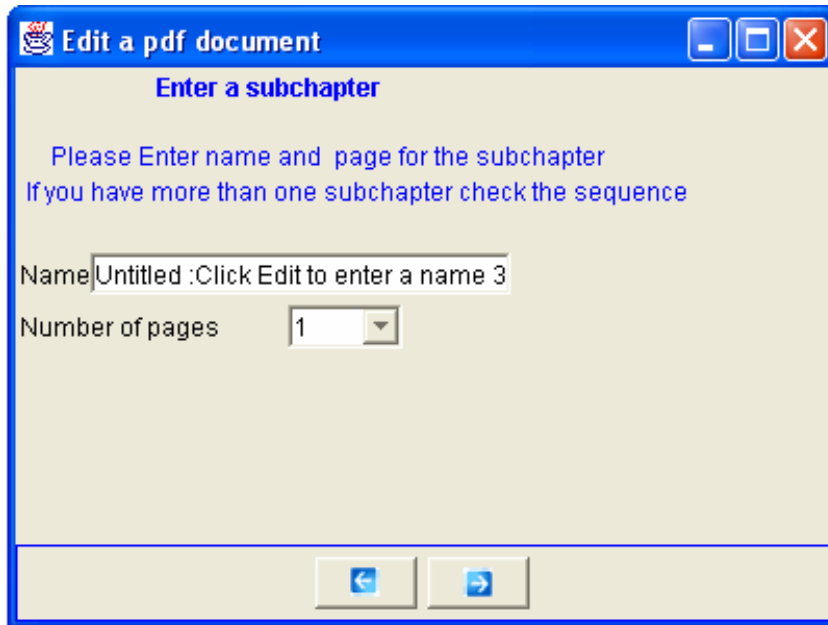
Εισάγουμε τις κατάλληλες πληροφορίες (τίτλο , που βρίσκεται) και στην συνέχεια , αν οι επιλογές μας είναι έγκυρες εμφανίζονται στο επόμενο παράθυρο (εικόνα 4.7) για επιβεβαίωση αλλιώς εμφανίζεται μήνυμα λάθους. Και εδώ ο αριθμός των σελίδων εισάγεται αυτόματα απο το πρόγραμμα . Πατώντας finish οι πληροφορίες εισάγονται στο κεφάλαιο και εμφανίζονται στο δέντρο.



εικόνα 4.7

Προσθήκη πληροφορίας σε υποκεφάλαιο.

Αν βρισκόμαστε σε κόμβο που αντιστοιχεί σε υποκεφάλαιο (δεν είναι παιδί της ρίζας) εμφανίζεται το παράθυρο της εικόνας 4.8.



εικόνα 4.8

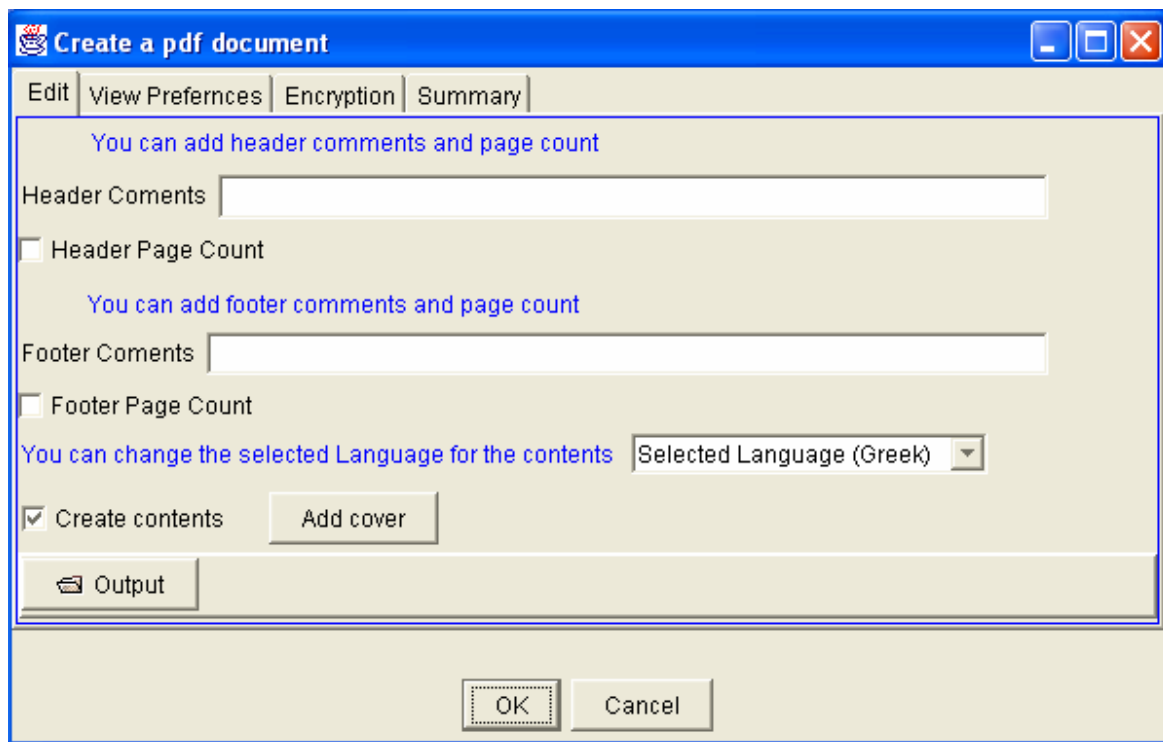
Εδώ είναι απαραίτητη η εισαγωγή της σελίδας που βρίσκεται το υποκεφάλαιο και ο τίτλος του .Για την εισαγωγή της σελίδας ισχύουν οι εξής περιορισμοί:

- Δεν μπορεί να υπερβαίνει τον αριθμό των σελίδων του κεφαλαίου που αντιστοιχεί.Για παράδειγμα αν το κεφάλαιο έχει 60 σελίδες το υποκεφάλαιο λογικά δεν μπορεί να είναι στην 61 σελίδα γιατί μπαίνει στις σελίδες άλλου κεφαλαίου.
- Αν υπάρχουν περισσότερα απο ένα υποκεφάλαια ελέγχεται η σειρά που είναι τοποθετημένα (αύξουσα σύμφωνα με την σελίδα).

Σε περίπτωση που δεν ισχύει κάποιος απο τους περιορισμούς εμφανίζεται μήνυμα λάθους στην οθόνη αλλιώς οι πληροφορίες εισάγονται επιτυχώς στο υποκεφάλαιο .

- ✓  Create Pdf - Δημιουργία του τελικού PDF αρχείου.

Η εφαρμογή παίρνει απο το δέντρο τις πληροφορίες που έχει το κάθε κεφάλαιο ή υποκεφάλαιο και δημιουργεί το τελικό αρχείο με τα περιεχόμενα του .Εμφανίζεται το παράθυρο της (εικόνας 4.9) με δυνατότητα επιλογής διαφορετικών Panel η λειτουργία των οποίων αναλύεται παρακάτω .



εικόνα 4.9

- Edit Panel (εικόνα 4.9) .

Η πιο βασική επιλογή στο Panel αυτό είναι η επιλογή του ονόματος και του directory που θα δημιουργηθεί το νέο αρχείο (***output***) . Χωρίς αυτή την επιλογή το νέο αρχείο δεν μπορεί να δημιουργηθεί .Επιπλέον μπορούμε να εισάγουμε περισσότερες πληροφορίες για το νέο αρχείο όπως;

Add cover: Επιλογή ένα εξώφυλλου σε PDF μορφή .

Create contents : Δημιουργία περιεχομένων που έχουν την δυνατότητα να κάνουν link στην σελίδα που αντιστοιχεί το κάθε κεφάλαιο ή υποκεφάλαιο . Μαζί με τα περιεχόμενα δημιουργείται και σελιδοδείκτης (Bookmark) που εμφανίζεται στα δεξιά του Acrobat Reader .

Selected Language : Επιλογή γλώσσας για τον τίτλο των περιεχομένων .

Header Coments ή ***Footer Coments***: Εισαγωγή σχολίων στην αρχή ή στο τέλος κάθε σελίδας αντίστοιχα του νέου αρχείου

Header Page Count ή ***Footer Page Count*** :Εισαγωγή αρίθμησης στην αρχή ή στο τέλος κάθε σελίδας αντίστοιχα του νέου αρχείου.

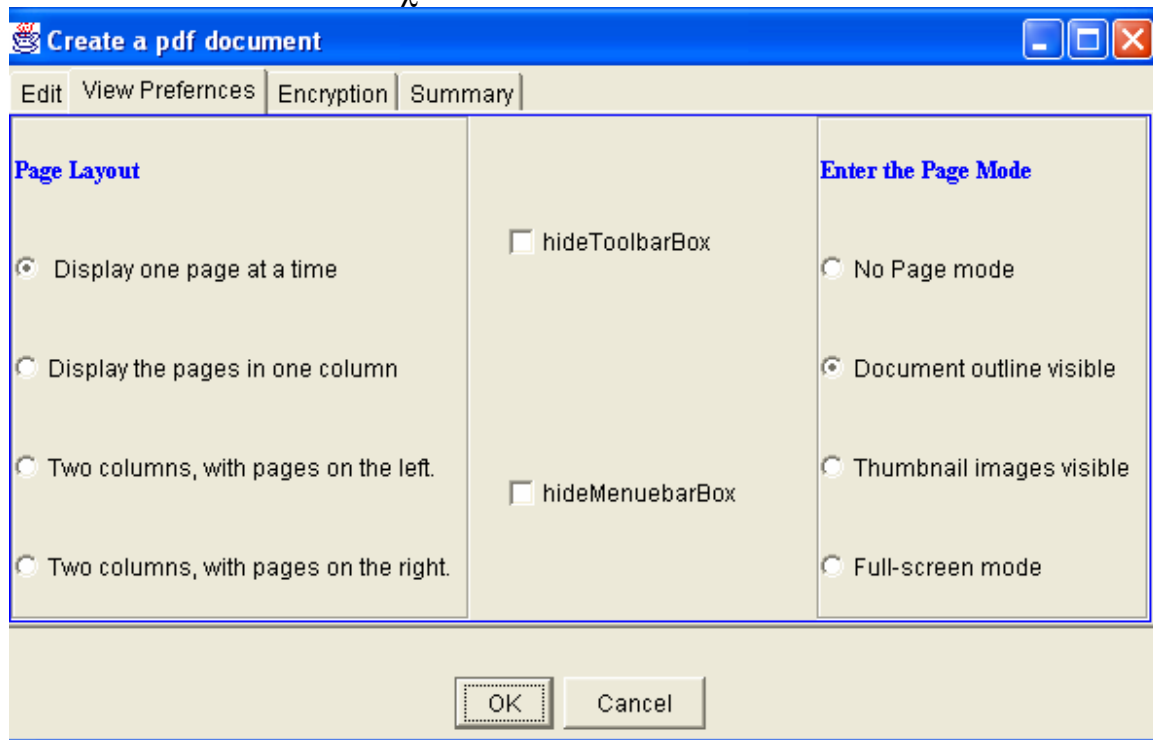
- View Preferences Panel (εικόνα 4.10).

Σ' αυτό το Panel επιλέγουμε πώς θέλουμε να παρουσιάζεται το PDF μόλις το ανοίγουμε .

Η στήλη με τίτλο **Page Layout** καθορίζει πως θα εμφανίζονται οι σελίδες.

Η στήλη με τίτλο **Enter the Page Mode** καθορίζει ποιά απο τα στοιχεία του Acrobat που βρίσκονται στην αριστερή πλευρά του θα εμφανίζονται.

Το **hideToolBarBox** και **hideMenubarBox** αποκρύπτει το Toolbar και το Menubar του Acrobat αντίστοιχα.



εικόνα 4.10

- Encryption Panel (εικόνα 4.12)

Αν θέλουμε να προστατεύσουμε το αρχείο απο άλλους χρήστες μπορούμε να θέσουμε συγκεκριμένους περιορισμούς .Η προστασία ενός εγγράφου με κωδικούς πρόσβασης σας προστατεύει απο μη εξουσιοδοτημένη χρήση .

Allow copy : Επιτρέπει την αντιγραφή του.

Allow Modify Contents : Επιτρέπει να αλλάζει τα δεδομένα του

Allow Print : Επιτρέπει την εκτύπωση του.

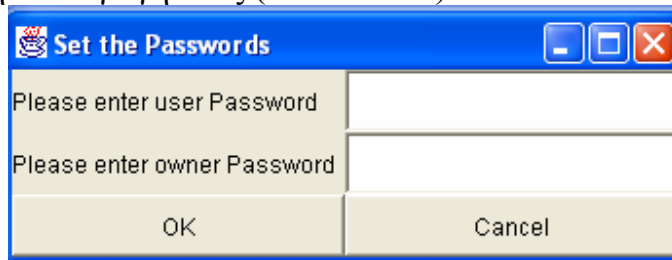
Allow Modify Annotations : Επιτρέπει να αλλάζει τις σημειώσεις του

Allow Fill in : Επιτρέπει να συμπληρώνονται οι φόρμες του

Allow ScreenReader:Επιτρέπει να διαβαστεί απο ένα ScreenReader

Allow Assembly:Επιτρέπει να αλλάξει η δομή ενός αρχείου.

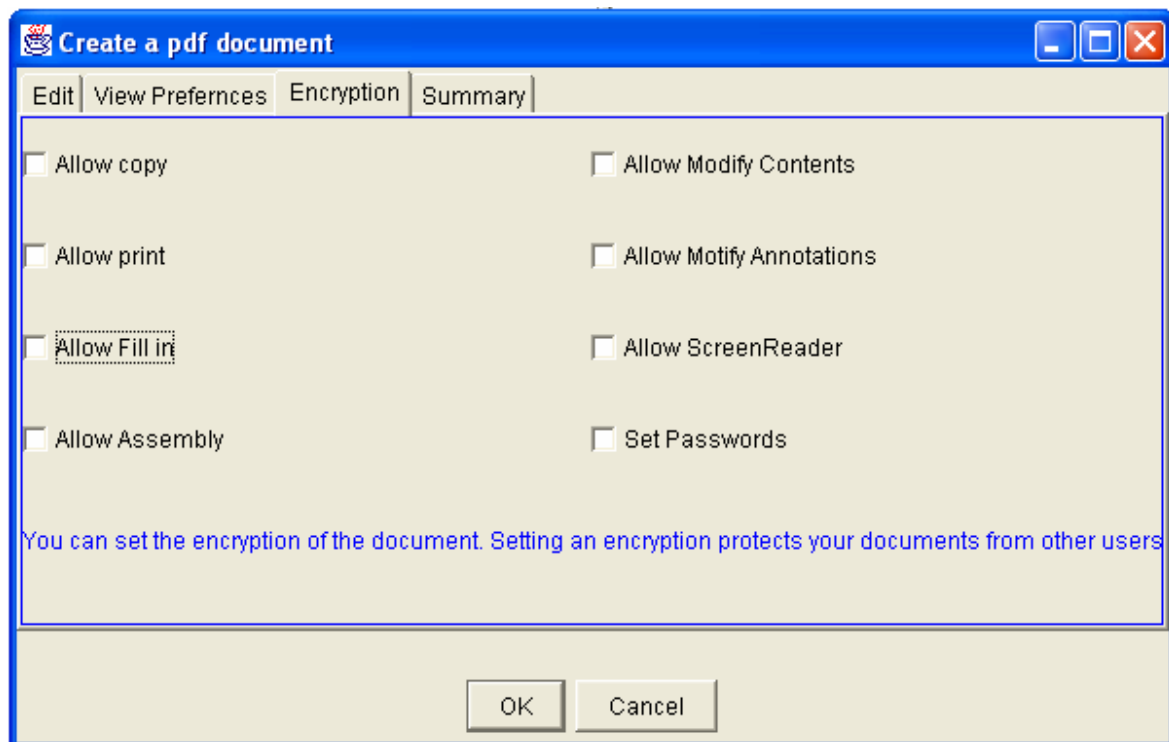
Set Passwords : Υπάρχουν δύο ειδών passwords για αυτό εμφανίζεται ένα νέο παράθυρο για την εισαγωγή τους (εικόνα 4.11) .



εικόνα 4.11

Το **UserPassword** το οποίο ζητείτε από τον χρήστη μόλις ανοίξει το αρχείο για να δει τα περιεχόμενα του .

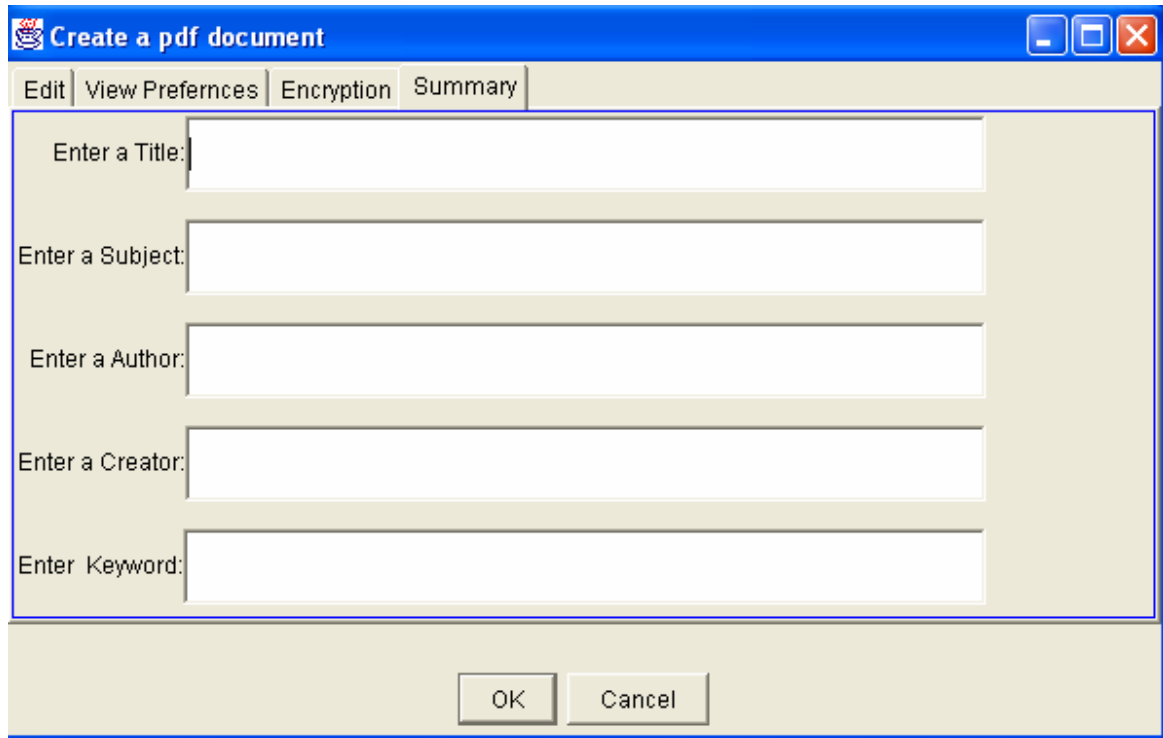
Το **OwnerPassword** το οποίο ζητείτε από τον χρήστη αν θέλει να αλλάξει τους περιορισμούς του αρχείου (encryption).



εικόνα 4.12

- Summary Panel (εικόνα 4.13)

Εδώ απλά εισάγουμε ενημερωτικά στοιχεία για το αρχείο όπως τίτλο ,θέμα ,συγγραφέα ,δημιουργό , λέξεις κλειδιά όπως φαίνεται και στην εικόνα 4.12 χωρίς να παίζουν κάποιο λειτουργικό ρόλο.



εικόνα 4.13

✓ Menu Επιλογών

Στο Menu επιλογών που βρίσκεται στο πάνω μέρος της οθόνης υπάρχουν διάφορες επιπλέον λειτουργίες που βοηθούν στην καλύτερη λειτουργία του προγράμματος .

Open – Save επιλογές .

Δίνεται η δυνατότητα στον χρήστη να σώσει το δέντρο με όλες τις πληροφορίες που δημιούργησε με το save και να το ξαναχρησιμοποιήσει με το open .Οι πληροφορίες του δέντρου σώζονται σε αρχεία με επέκταση PDFM.

Credits

Εμφανίζει πληροφορίες για την εφαρμογή .

4.3 Πρακτική εφαρμογή – Δημιουργία Οδηγού Σπουδών Ηλεκτρολόγων Μηχανικών Ε.Μ.Π.

Όπως έχει αναφερθεί και στην αρχή του κεφαλαίου δημιουργήθηκε ο οδηγός σπουδών του Ε.Μ.Π. σαν πρακτική εφαρμογή. Όλα τα PDF αρχεία που χρειάζονται για την δημιουργία του οδηγού βρίσκονται στο directory \odhgos και τα XSL για τον μετασχηματισμό στο \odhgos\xsl-xml. Τα κεφάλαια που μπορούν να αναβαθμιστούν είναι τα 5,7,8,9,10. Τα υπόλοιπα έχουν στατικό περιεχόμενο και δεν χρειάζονται αναβάθμιση.

Ακολουθώντας τα βήματα του παραδείγματος είναι προφανές ότι η ετήσια αναπαραγωγή του οδηγού θα γίνεται σε ελάχιστο χρόνο και θα υπάρχει η δυνατότητα εμπλουτισμού του αρχείου με νέες τεχνολογίες PDF.

Βήμα 1^ο

Με την βοήθεια του menu File->Open ανοίγουμε το αρχείο odhgos-sproudown.PDFM. Στο αρχείο αυτό έχω σώσει το δέντρο με τα κεφάλαια και υποκεφάλαια του οδηγού προηγούμενου έτους (εικόνα 4.14).

Βήμα 2^ο

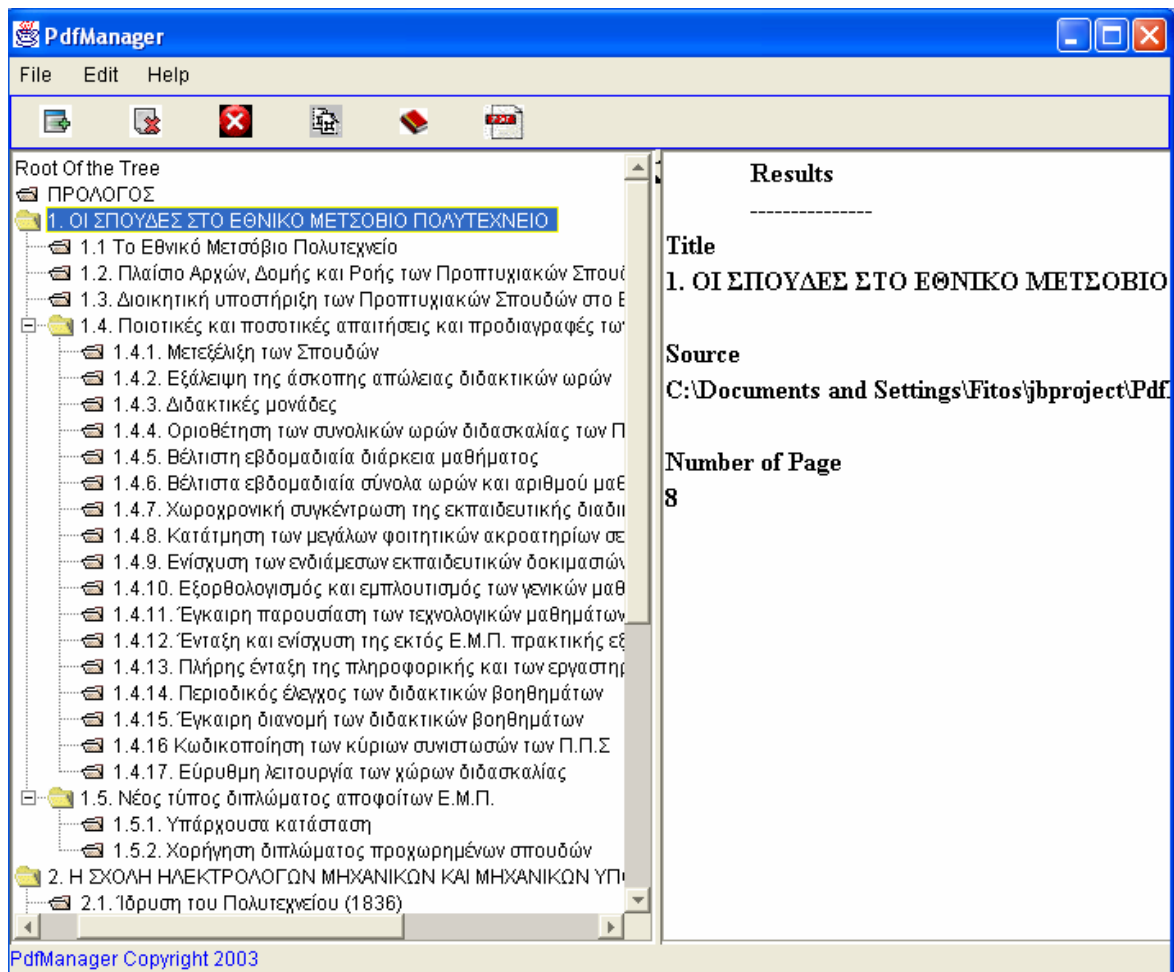
Ανοίγουμε ένα browser και κατεβάζουμε απο την διεύθυνση <http://localhost:8080/index.htm> τα 3 XML αρχεία που θα μας χρειαστούν για την δημιουργία των νέων κεφαλαίων. (Προσθέτουμε επιπρόσθετη πληροφορία στα αρχεία που χρειάζονται).

Βήμα 3^ο

Επιλέγουμε στο δέντρο των περιεχόμενων το κεφάλαιο που θέλουμε να αναβαθμίσουμε και δημιουργούμε το νέο κεφάλαιο του οδηγού σε directory της επιλογής μας με την βοήθεια της εντολής Edit.

Επειδή τα υποκεφάλαια έχουν διαμορφωθεί σύμφωνα με το παλιό κεφάλαιο του οδηγού, μετά την δημιουργία του νέου κεφαλαίου χρειάζεται ένας έλεγχος στις σελίδες τους για να έχουμε το επιθυμητό αποτέλεσμα. Αν χρειάζεται αλλαγή σε κάποιο απ αυτά χρησιμοποιούμε την εντολή Edit.

Αν τυχόν κάποιο απο τα στατικά κεφάλαια χρειάζεται να αλλάξει απλά αντικαθιστούμε το αρχείο του συγκεκριμένου κεφαλαίου και ελέγχουμε τις σελίδες των υποκεφαλαίων του.

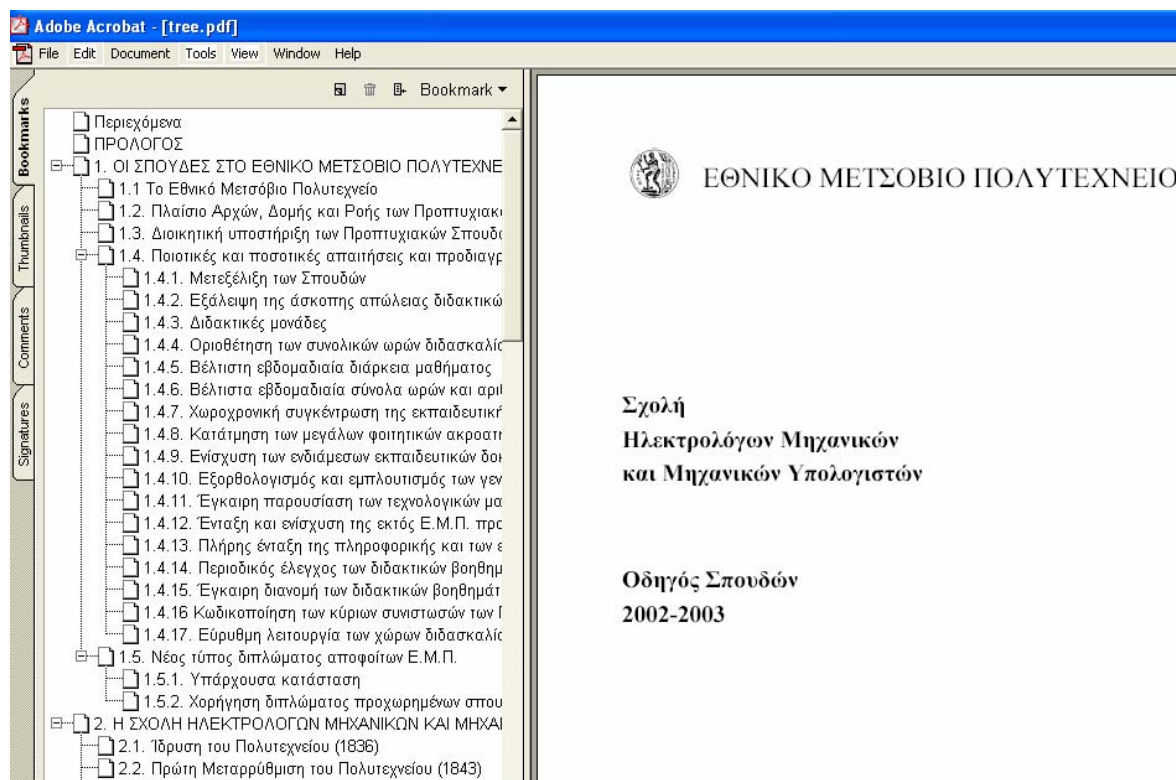


εικόνα 4.14

Βήμα 4^ο

Το δέντρο του νέου οδηγού σπουδών είναι έτοιμο. Με την βοήθεια του Create Pdf εισάγουμε τα meta-data (πως θα παρουσιάζεται ,αν θα έχει απόκρυψη ,ποιο θα είναι το εξώφυλλο) όπως δείξαμε πιο πάνω.

Αν όλα τα στοιχεία που έχουμε εισαγάγει είναι σωστά θα δημιουργηθεί ο νέος οδηγός σπουδών του τμήματος και το μόνο που μένει είναι να ανοίξουμε το αρχείο με την βοήθεια ενός Acrobat Reader(εικόνα 4.15)..



εικόνα 4.15

5. ΚΩΔΙΚΑΣ

Παρουσιάζεται παρακάτω ο κώδικας της εφαρμογής με συνοδευτικά σχόλια που καθιστούν πιο εύκολη την κατανόηση του .

5.1 Πακέτο PdfManager

5.1.1 PdfManager.java

Η βασική κλάση για την επικοινωνία του προγράμματος με το χρήστη. Διαχειρίζεται όλες τις υπόλοιπες κλάσεις του package αυτού και σ' αυτήν βρίσκεται η main() για την εκτέλεση του προγράμματος.

```
package pdfmanager;

/**
 * Title: PdfManager
 * Description: A program tha manage Pdf documents
 * Copyright: Copyright (c) 2003
 * Company: Ece Ntua
 * @Vlotomas Neofytos e197605
 * @version 1.0
 */
import java.awt.*;
import java.awt.event.*;
import java.beans.*;
import java.io.*;
import java.util.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.tree.*;

public class PdfManager extends JPanel {
/*
Declarations for the tree
*/
//the root of the tree
private static DefaultMutableTreeNode rootNode;
private JTree tree;
private DefaultTreeModel treeModel ;
private DefaultTreeCellRenderer renderer ;
/*
Swing declarations
```

```

*/
private JTextArea frametext;
private JLabel panelcoment;
private JMenu filemenu, helpmenu,editmenu;
/*
Some Icons
*/
private ImageIcon savepic = new ImageIcon("images/save.gif");
private ImageIcon add = new ImageIcon("images/add.jpg");
private ImageIcon remove = new ImageIcon("images/remove.jpg");
private ImageIcon edit = new ImageIcon("images/edit.gif");
private ImageIcon acrobat = new ImageIcon("images/pdfdoc.gif");
private ImageIcon clear = new ImageIcon("images/clear.gif");
private ImageIcon open = new ImageIcon("images/open.gif");
private ImageIcon splitpic = new ImageIcon("images/split.gif");
/*
global Declarations
*/
private int newNode=1;
private String title="      Results \n      -----";
private String information="";
private String userdir=System.getProperty("user.dir")+
System.getProperty("file.separator");
private Properties save;
private File file;
private boolean changed=false;
private String curentstring="PdfManager:curentdir"+System.getProperty("file.separator");
private Font font1=new Font("TimesRoman",Font.BOLD,16);

public PdfManager(JFrame frame) {

/*
Construct the Tree
*/
rootNode = new DefaultMutableTreeNode("Root Of the Tree");
treeModel = new DefaultTreeModel(rootNode);
tree = new JTree(treeModel);
tree.putClientProperty("JTree.lineStyle", "Angled");
renderer = new DefaultTreeCellRenderer();
renderer.setLeafIcon(open);
renderer.setForeground(Color.red);
tree.setCellRenderer(renderer);
tree.setEditable(false);
final DynamicTree treePanel = new DynamicTree(rootNode,treeModel,tree);

```

```

tree.getSelectionModel().setSelectionMode(TreeSelectionMode.SINGLE_TREE_SELECTION);
tree.setShowsRootHandles(true);
//Tree Listener
tree.addTreeSelectionListener(new TreeSelectionListener()
{
public void valueChanged(TreeSelectionEvent e) {
DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree.getLastSelectedPathComponent();

if (node == null || node==rootNode) return;else
{
Object nodeInfo = node.getUserObject();
Info per=(Info)nodeInfo;
//Display information of each node of the tree in frametext
if (node.getLevel()==1 )
{//Display information for chapter
information="Title\n"+per.getName()+"\n\n";
information=information+"Source\n"+per.getPdfSource()+
"\n\nNumber of Page\n"+per.page;
information=information+"\n\nDescription\nChapter";
frametext.setText(title+"\n"+information);
information="";
} //Display information for subchapter
else frametext.setText(title+"\n"+per.getName()+"\n"+"Page\n"+
+per.getPages()+"\n\nDescription\nSubChapter");
}
}
});

/*
Make the Buttons
*/
//The Button to add a node
JButton addButton = new JButton("",add);
addButton.setBorderPainted(false);
//Listener for the addbutton
addButton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
final JFrame message=new JFrame();
DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree.getLastSelectedPathComponent();
//Add a chapter

```

```

if (node == null || node==rootNode)
{
    //Add a chapter with default information
    treePanel.addObject(new Info("Untitled :Click Edit to enter a name and a pdf source
"+newNode,null,0,false,true,1));
    newNode++;
    changed=true;
}
else
{//Add a subchapter
    Object nodeInfo = node.getUserObject();
    Info per=(Info)nodeInfo;
    if(!per.untitled && per.sub)
    { //Add a subchapter with default information
        treePanel.addObject(new Info("Untitled :Click Edit to enter a name
"+newNode,null,0,true,true,node.getLevel()+1));
        newNode++;
        changed=true;
    }
    else if(per.getPdfSource()!=null && !per.sub)
    {
        treePanel.addObject(new Info("Untitled :Click Edit to enter a name
"+newNode,null,0,true,true,node.getLevel()+1));
        newNode++;
        changed=true;
    }
    //You cant add a subchapter if the selected node is untitled
    else JOptionPane.showMessageDialog(message,"You have to enter a valid PDF source
,or name first",
    "You cant add a subchapter ", JOptionPane.WARNING_MESSAGE);
}
});
//The Button to remove a node
JButton removeButton = new JButton("",remove);
removeButton.setBorderPainted(false);
//Listener for the removebutton
removeButton.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        //Call DynamicTree class to remove the node
        treePanel.removeCurrentNode();
        changed=treePanel.changed;
    }
});

```

```

//The Button to delete all the tree
JButton clearButton = new JButton("",clear);
clearButton.setBorderPainted(false);
//Listener for the clearbutton
clearButton.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        final JFrame message=new JFrame();
        //Display a message
        final JOptionPane optionPane = new JOptionPane( "Do you want to delete all the
chapters;\n"
        ,JOptionPane.QUESTION_MESSAGE, JOptionPane.YES_NO_OPTION);
        final JDialog dialog = new JDialog(message, "Delete all;",true);
        //Listener for the message
        optionPane.addPropertyChangeListener(
        new PropertyChangeListener()
        {
            public void propertyChange(PropertyChangeEvent e)
            {
                String prop = e.getPropertyName();
                if (dialog.isVisible() && (e.getSource() == optionPane) &&
(prop.equals(JOptionPane.VALUE_PROPERTY)
||prop.equals(JOptionPane.INPUT_VALUE_PROPERTY)))
                {
                    dialog.setVisible(false);
                }
            }
        });
        dialog.setDefaultCloseOperation(
        JDialog.DO_NOTHING_ON_CLOSE);
        dialog.addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent we)
            {
                }
        });
        dialog.setContentPane(optionPane);
        dialog.pack();
        dialog.setLocation(200, 200);
        dialog.setVisible(true);
        int value = ((Integer)optionPane.getValue()).intValue();
        //if we choose NO ,it doesnt do nothing
        if (value == JOptionPane.NO_OPTION)
        {
            message.dispose();
        }
    }
}

```

```

    }
    //if we choose YES the tree will be deleted
    else if(value == JOptionPane.YES_OPTION)
    {
        //Call the DynamicTree class
        treePanel.clear();
        changed=false;
        message.dispose();
    }
}
});
//The button to edit a node
JButton editbutton=new JButton("",edit);
editbutton.setBorderPainted(false);
//Listener for the editbutton
editbutton.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        DefaultMutableTreeNode node =
(DefaultMutableTreeNode)tree.getLastSelectedPathComponent();
        if (node == null) return;
        else
        {
            //Call EditSourceFrame to edit informations for the selected nodes
            EditSourceFrame edit=new EditSourceFrame(node);
            changed=true;
        }
    }
});
//The button to create the final Pdf file
JButton createbutton=new JButton("",acrobat);
createbutton.setBorderPainted(false);
//Listener for the createbutton
createbutton.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        //Call the CreatePdfFrame class to create the file
        CreatePdfFrame create=new CreatePdfFrame(rootNode);
    }
});
//The button to split a pdf file
JButton splitbutton=new JButton("",splitpic);
splitbutton.setBorderPainted(false);
//Listener for the splitbutton

```



```

splitbutton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
//Call the SplitPdf class to split a Pdf file
SplitPdf split=new SplitPdf();
}
});

/*
Set the Menu Bar
*/
JMenuBar menuBar=new JMenuBar();
menuBar.setBorderPainted(false);
frame.setJMenuBar(menuBar);
//Build file menu.
filemenu = new JMenu("File ");
filemenu.setMnemonic(KeyEvent.VK_ALT);
//Build help menu.
helpmenu = new JMenu("Credits ");
//Build edit menu.
editmenu=new JMenu("Edit ");
menuBar.add(filemenu);
menuBar.add(editmenu);
menuBar.add(helpmenu);
//Set FileMenu
//Open File menu
JMenuItem openfile = new JMenuItem("Open Pdfm files",open);
openfile.setMnemonic(KeyEvent.VK_B);
//Openfile Listener
openfile.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
JFrame message=new JFrame();
//If we make changes to a tree without saving a message will be displayed
final JOptionPane optionPane = new JOptionPane( "Do you want to save the changes;\n"
,JOptionPane.QUESTION_MESSAGE, JOptionPane.YES_NO_CANCEL_OPTION);
final JDialog dialog = new JDialog(message, "Save changes;",true);
//Listener for the message
optionPane.addPropertyChangeListener(new PropertyChangeListener()
{
public void propertyChange(PropertyChangeEvent e)
{
String prop = e.getPropertyName();

```

```

    if (dialog.isVisible() && (e.getSource() == optionPane) &&
(prop.equals(JOptionPane.VALUE_PROPERTY)
||prop.equals(JOptionPane.INPUT_VALUE_PROPERTY)))
    {
        dialog.setVisible(false);
    }
    });
dialog.setDefaultCloseOperation(
JDialog.DO_NOTHING_ON_CLOSE);
dialog.addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent we)
    {

    }
});

if (changed)
{
    dialog.setContentPane(optionPane);
    dialog.pack();
    dialog.setLocation(200, 200);
    dialog.setVisible(true);

    int value = ((Integer)optionPane.getValue()).intValue();
    //if we choose no open a file
    if (value == JOptionPane.NO_OPTION) OpenFile(treePanel);
    //if we choose yes save a file
    else if (value == JOptionPane.YES_OPTION) SaveFile();
}
else OpenFile(treePanel);
});
//Save file menu
filemenu.add(openfile);
filemenu.addSeparator();
JMenuItem savefile = new JMenuItem("Save Pdfm files", savepic);
savefile.setMnemonic(KeyEvent.VK_D);
//Save file Listener
savefile.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        SaveFile();
    }
}

```

```

});
filemenu.add(savefile);
//Set HelpMenu
JMenuItem Credits = new JMenuItem("Credits");
//Display Credits
Credits.setMnemonic(KeyEvent.VK_D);
Credits.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
DisplayUrl cred=new DisplayUrl("credits");
}
});
helpmenu.add(Credits);
//Set edit menu(Makes the same job with buttons )
JMenuItem Addmenu = new JMenuItem("Add",add);
editmenu.add(Addmenu);
Addmenu.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
final JFrame message=new JFrame();
DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree.getLastSelectedPathComponent();

if (node == null || node==rootNode)
{
treePanel.addObject(new Info("Untitled :Click Edit to enter a name and a pdf source
"+newNode,null,0,false,true,1));
newNode++;
}
else
{
Object nodeInfo = node.getUserObject();
Info per=(Info)nodeInfo;
if(!per.getName().startsWith("Untitled") && per.sub)
{
treePanel.addObject(new Info("Untitled :Click Edit to enter a name
"+newNode,null,0,true,true,node.getLevel()+1));
newNode++;
}
else if(per.getPdfSource()!=null && !per.sub)
{
treePanel.addObject(new Info("Untitled :Click Edit to enter a name
"+newNode,null,0,true,true,node.getLevel()+1));

```

```

        newNode++;
    }
    else JOptionPane.showMessageDialog(message,"You have to enter a valid PDF source
,or name first",
    "You cant add a subchapter ", JOptionPane.WARNING_MESSAGE);
    }
    }
    });

```

```

JMenuItem removemenu = new JMenuItem("Remove",remove);
editmenu.add(removemenu);
removemenu.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
    treePanel.removeCurrentNode();
}
});

```

```

JMenuItem deletemenu = new JMenuItem("Delete",clear);
editmenu.add(deletemenu);
deletemenu.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
    treePanel.clear();
}
});

```

```

editmenu.addSeparator();
JMenuItem Splitmenu = new JMenuItem("Split",splitpic);
editmenu.add(Splitmenu);
Splitmenu.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
    SplitPdf split=new SplitPdf();
}
});

```

```

JMenuItem editchaptermenu = new JMenuItem("Edit",edit);
editchaptermenu.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
    DefaultMutableTreeNode node = (DefaultMutableTreeNode)

```

```

tree.getLastSelectedPathComponent();
if (node == null) return;
else {Object nodeInfo = node.getUserObject();
Info per=(Info)nodeInfo;
EditSourceFrame edit=new EditSourceFrame(node);
}
}
});
editmenu.add(editchaptermenu);
JMenuItem createmenu = new JMenuItem("CreatePdf",acrobat);
editmenu.add(createmenu);
createmenu.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
CreatePdfFrame create=new CreatePdfFrame(rootNode);
}
});
//Lay everything to frame.
frametext=new JTextArea();
JScrollPane Scrolltext = new JScrollPane(frametext);
//Makes a scrollpane
JSplitPane splitPane = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,
treePanel, Scrolltext);
splitPane.setOneTouchExpandable(true);
splitPane.setDividerLocation(500);
setLayout(new BorderLayout());
treePanel.setPreferredSize(new Dimension(505, 630));
Scrolltext.setPreferredSize(new Dimension(505,630));
add(splitPane, BorderLayout.WEST);
frametext.setText(title);
frametext.setFont(font1);
frametext.setEditable(false);
JToolBar panel = new JToolBar();
JPanel dpanel=new JPanel();
menuBar.setBackground(panel.getBackground());
panel.setBorder(BorderFactory.createLineBorder(Color.blue));
panel.setAlignmentY(TOP_ALIGNMENT);
filemenu.setBackground(panel.getBackground());
editmenu.setBackground(panel.getBackground());
helpmenu.setBackground(panel.getBackground());
panel.setLayout(new BoxLayout(panel,BoxLayout.X_AXIS));
panel.add(addButton);
addButton.setToolTipText("Add a new chapter ");
panel.add(removeButton);
removeButton.setToolTipText("Remove curent chapter");

```

```

panel.add(clearButton);
clearButton.setToolTipText("Remove all chapters");
panel.add(splitbutton);
splitbutton.setToolTipText("Split the selected pdf page by page");
panel.add(editbutton);
editbutton.setToolTipText("Edit curent chapter");
panel.add(createbutton);
panelcoment=new JLabel("PdfManager Copyright 2003");
panelcoment.setForeground(Color.blue);
createbutton.setToolTipText("Create the pdf document");
dpanel.setLayout(new GridLayout(1,0));
dpanel.setAlignmentX(dpanel.LEFT_ALIGNMENT);
dpanel.add(panelcoment);
add(panel, BorderLayout.NORTH);
add(dpanel, BorderLayout.SOUTH);
}

```

```

//Save a file with Pdfm extension
public void SaveFile()
{
    save =new Properties();
    boolean next=true;
    Object nodeInfo;
    int x=1;
    DefaultMutableTreeNode node=rootNode;
    //Begin from the root of the tree
    while(next)
    {
        node= node.getNextNode();
        if (node==null)next=false;
        else
        { //Get the information for the current node
            nodeInfo = node.getUserObject();
            Info per=(Info)nodeInfo;
            //Save Title,Source,page,if its a leaf,if its untitled,level
            save.put("Title"+x,per.getName());
            save.put("Source"+x,per.pdfsource+""");
            save.put("Page"+x, """+per.getPages());
            save.put("Leaf"+x,per.sub+""");
            save.put("Untitled"+x,per.untitled+""");
            save.put("Level"+x, """+per.level);
            x++;
        }
    }
    //Select the File to save the tree
    JFileChooser fc=new JFileChooser(userdir+"save");

```

```

//Show only file with Pdfm extension
fc.addChoosableFileFilter(new Acceptedfiles("pdfm"));
fc.setAcceptAllFileFilterUsed(false);
int returnVal = fc.showOpenDialog(PdfManager.this);
if (returnVal == JFileChooser.APPROVE_OPTION)
{
    file = fc.getSelectedFile();
}
else return;
try
{
    DataOutputStream out;
    //Store tree informations
    if (file.getAbsolutePath().endsWith(".pdfm"))
    {
        out = new DataOutputStream( new BufferedOutputStream( new
FileOutputStream(file.getAbsolutePath() ) ) );
    }
    else
    {
        out = new DataOutputStream( new BufferedOutputStream( new
FileOutputStream(file.getAbsolutePath()+".pdfm" ) ) );
    }
    //Store the date also
    save.store(out, new java.util.Date().toString());
}
catch(Exception ex)
{
    System.out.println("Error:" + ex);
}
changed=false;
}

```

```

//Open a file with Pdfm extension
public void OpenFile(DynamicTree treePanel)
{
    JFrame message=new JFrame();
    save =new Properties();
    boolean isLeaf;
    boolean isUntitled;
    String source="";
    int nodes=1;
    int parentlevel=0;
    int level=0;
    DefaultMutableTreeNode parent=rootNode;
    //Select the file to open

```

```

JFileChooser fc=new JFileChooser(userdir+"save");
fc.addChoosableFileFilter(new Acceptedfiles("pdfm"));
fc.setAcceptAllFileFilterUsed(false);
int returnVal = fc.showOpenDialog(PdfManager.this);
if (returnVal == JFileChooser.APPROVE_OPTION)
{
    file = fc.getSelectedFile();
}
else return;
try
{
    //Get the informations from the selected file
    DataInputStream in = new DataInputStream( new BufferedInputStream(new
FileInputStream(file.getAbsolutePath() ) ));
    save.load(in);
}
catch(IOException ex)
{
    JOptionPane.showMessageDialog(message,ex.getMessage(),
    "Opening Error ", JOptionPane.ERROR_MESSAGE);
}
treePanel.clear();
while(save.getProperty("Title"+nodes)!=null)
{ //Get the informations
    if (save.getProperty("Leaf"+nodes).equals("false"))
        isLeaf=false;else isLeaf=true;
    if (save.getProperty("Untitled"+nodes).equals("false"))
        isUntitled=false;else isUntitled=true;
    if (save.getProperty("Source"+nodes).equals("null"))
        source=null;
    else
        source=save.getProperty("Source"+nodes);
    level= Integer.parseInt(save.getProperty("Level"+nodes));
    //Set the level of the current node
    if(level==1)
    {
        treePanel.addObject(new Info(save.getProperty("Title"+nodes),
        source,(Integer.parseInt(save.getProperty("Page"+nodes))), isLeaf,isUntitled,1));
        nodes++;
        parentlevel=1;
        parent=parent.getNextNode();
    }
    else
    {
        if (level>parentlevel)
        {

```



```

treePanel.addObject(parent,new Info(save.getProperty("Title"+nodes),
source,(Integer.parseInt(save.getProperty("Page"+nodes))), isLeaf,isUntitled,level));
nodes++;
parentlevel=level;
parent=parent.getNextNode();
}
else if (level==parentlevel)
{
//Find the parent
DefaultMutableTreeNode p=parent;
while(p.getLevel()!=level-1)p=p.getPreviousNode();
treePanel.addObject(p,new Info(save.getProperty("Title"+nodes),
source,(Integer.parseInt(save.getProperty("Page"+nodes))), isLeaf,isUntitled,level));
nodes++;
parent=parent.getNextNode();
}
else
{
//Find the parent
DefaultMutableTreeNode p=parent;
while(p.getLevel()!=level-1)p=p.getPreviousNode();
treePanel.addObject(p,new Info(save.getProperty("Title"+nodes),
source,(Integer.parseInt(save.getProperty("Page"+nodes))), isLeaf,isUntitled,level));
nodes++;
parentlevel=level; parent=parent.getNextNode();
}
}
}

changed=false;
}
//main
public static void main(String[] args)
{
JFrame frame = new JFrame("PdfManager");
try
{
UIManager.setLookAndFeel( UIManager.getSystemLookAndFeelClassName() );
}
catch( Exception e )
{
e.printStackTrace();
}
Container contentPane = frame.getContentPane();
contentPane.setLayout(new GridLayout(1,1));
contentPane.add(new PdfManager(frame));
}
}

```

```

frame.addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});

frame.pack();
frame.setVisible(true);
}
}

```

5.2 PdfCreator.java

Σ' αυτή την κλάση γίνεται ο μετασχηματισμός των κατάλληλων XML/XSL και δημιουργούνται τα νέα αρχεία PDF.

```

package pdfmanager;

/**
 * Title: PdfManager
 * Description: A program tha manage Pdf documents
 * Copyright: Copyright (c) 2003
 * Company: Ece Ntua
 * @Vlotomas Neofytos e197605
 * @version 1.0
 */

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import javax.swing.*;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.stream.StreamSource;
import javax.xml.transform.sax.SAXResult;
import com.lowagie.text.*;
import com.lowagie.text.pdf.*;
import org.apache.fop.apps.*;

public class PdfCreator
{
/**

```

```

These variables need to be initialized for the transformation
*/
private String xmlFile, xslFile, outputFile, watermarkFile;
private Transformer transformer = null;
/*
  Needed by FOP
*/
private Driver driver = null;
private SAXResult saxResult;
private ByteArrayOutputStream fopout;
private org.apache.avalon.framework.logger.Logger logger = null;
/*
global Declarations
*/
private      String      curentdir=System.getProperty("user.dir")+
System.getProperty("file.separator")+"conf" ;
private JOptionPane message=new JOptionPane();
private JFrame frame;

public PdfCreator()
{
}
//Set the files for the tranformation (from EditSource Class)
public PdfCreator(String xmlFile, String xsltFile, String outputFile)
{
  this();
  this.setXmlFile(xmlFile);
  this.setXsltFile(xsltFile);
  this.outputFile = outputFile;
}
//Set Xsl file
public void setXsltFile(String xslFile)
{
  if (xslFile == null)
  {
    this.xslFile = null;
    return;
  }
  if (xslFile.equals(this.xslFile)) { return; }
  this.xslFile = xslFile;
}

}

//Set output file (with Pdf extension)
public void setOutputFile(String outputFile)
{

```

```

    this.outputFile = outputFile;
}

//Set the Xml file
public void setXmlFile(String fileName)
{
    this.xmlFile = fileName;
    File f = new File(fileName);
    // we need the baseDir to use graphics
    org.apache.fop.configuration.Configuration.put("baseDir", "file:/// " + f.getParent() +
System.getProperty("file.separator"));
}
//Trasform the XML/XSL files.
public void transform() throws Exception
{
    //Set the output file for the PDF
    FileOutputStream outfile = new FileOutputStream(outputFile);
    //Generade the PDF file
    PdfReader reader = generatePDF();
    //Show the PDF with iText API
    Document document = new Document(reader.getPageSizeWithRotation(1));
    PdfWriter writer = PdfWriter.getInstance(document, outfile );
    document.open();
    PdfContentByte cb = writer.getDirectContent();
    // handling the pages
    int rotation = 0;
    int i = 0;
    //Get the pagenumber of the file
    int n = reader.getNumberOfPages();
    Rectangle rect = reader.getPageSizeWithRotation(1);
    PdfImportedPage page = null;
    //Show all the page of the new pdf file
    while (i < n)
    {
        i++;
        document.setPageSize(reader.getPageSizeWithRotation(i));
        document.newPage();
        try
        {
            page = writer.getImportedPage(reader, i);
        }
        catch (Exception ex)
        {
            System.err.println("-----"+ex.getMessage());
        }
        rotation = reader.getPageRotation(i);
    }
}

```

```

    if (rotation == 90 || rotation == 270)
    {
        cb.addTemplate(page, 0, -1f, 1f, 0, 0,
            reader.getPageSizeWithRotation(i).height());
    }
    else
    {
        cb.addTemplate(page, 1f, 0, 0, 1f, 0, 0);
    }
}
document.close();
}

private PdfReader generatePDF() throws Exception
{
    //Set the fop-output file
    ByteArrayOutputStream fopout = new ByteArrayOutputStream();
    //Initialises the Driver and sets the Logger.
    Driver driver = new Driver();
    driver.setOutputStream(fopout);
    driver.setRenderer(Driver.RENDER_PDF);
    org.apache.avalon.framework.logger.ConsoleLogger logger =
    new org.apache.avalon.framework.logger.ConsoleLogger(
    org.apache.avalon.framework.logger.ConsoleLogger.LEVEL_FATAL);
    org.apache.fop.messaging.MessageHandler.setScreenLogger(logger);
    driver.setLogger(logger);
    try
    {
        //Set the xml file for font configurations
        org.apache.fop.apps.Options options = new Options(new
File(curentdir+System.getProperty("file.separator")+"userconfig.xml"));
    }
    catch (Exception ex){System.err.print(ex.getMessage());}
    // handling the FOP transformation
    //Create a Transformer with TransformerFactory Instance and loads it with an XSLT file
    Transformer transformer = TransformerFactory.newInstance().
    newTransformer( new StreamSource(new File(xslFile)));
    //Transforms the xmlfile and generate a pdf file
    transformer.transform(new StreamSource(new File(xmlFile)),
    new SAXResult(driver.getContentHandler()));
    //FOP done
    return new PdfReader(fopout.toByteArray());
}
}
}

```

5.1.3 Info.java

Η κλάση αυτή χρησιμοποιείται για να κρατάει πληροφορίες για τον κάθε κόμβο του δέντρου ξεχωριστά:

- Τίτλο (name)
- Πηγή του αρχείου (pdfsource)
- Αριθμό Σελίδων (page)
- Αν είναι υποκεφάλαιο (sub)
- Αν έχει επεξεργαστεί (untitled)
- Σε ποιο βάθος βρίσκεται στο δέντρο (level)

```
package pdfmanager;
```

```
/**  
 * Title: PdfManager  
 * Description: A program tha manage Pdf documents  
 * Copyright: Copyright (c) 2003  
 * Company: Ece Ntua  
 * @Vlotomas Neofytos el97605  
 * @version 1.0  
 */
```

```
import com.lowagie.text.pdf.*;
```

```
public class Info {  
/*  
global Declarations  
*/  
public String name;  
public String pdfsource;  
public int page;  
public boolean sub;  
public boolean untitled;  
public int level=0;  
//The files in PdfManager begin with curentstring  
private String curentstring="PdfManager:curentdir";  
//We get the PdfManager current directory  
private String userdir=System.getProperty("user.dir");  
  
public Info(String name,String pdfsource,int page,boolean sub,boolean untitled,int level)  
{ //Set the information for the node of the tree  
this.name = name;  
this.page=page;  
this.sub=sub;
```

```

this.pdfsource=pdfsource;
this.untitled=untitled;
this.level=level;
}
// getter methods
public String toString(){return name;}
public String getName() { return name; }
public void reName(String newname){this.name=newname;}
public String getPdfSource()
{
    if(pdfsource==null)return null;
    else
    { //if pdfsource begin with curentstring means that the file is
      //in PdfManager directory
      if(pdfsource.startsWith(curentstring))
      {
          String result="";
          result=userdir+
System.getProperty("file.separator")+pdfsource.substring(curentstring.length()+1);
          return result;
      }
      else return pdfsource;
    }
}

public void setPage(int pages)
{
    this.page=pages;
}
//get the number off page
public int getPages()
{
    if (pdfsource==null) return page;
    else
    {
        try{// we retrieve the total number of pages
            if(pdfsource.startsWith(curentstring))
            { //if the file is in PdfManager directory
                PdfReader reader = new PdfReader(userdir+
System.getProperty("file.separator")+pdfsource.substring(curentstring.length()+1));
                page = reader.getNumberOfPages();
            }
            else
            { //if the file is somewhere else
                PdfReader reader = new PdfReader(pdfsource);

```

```

        page = reader.getNumberOfPages();
    }
}
catch (Exception ex)
{
    System.err.println(ex.getMessage());
    return -2;
}
}
this.page=page; return page;
}
//Set the Pdfsource(for chapter)
public void setPdfSource(String source)
{
    if(source==null)this.pdfsource=null;
    else if (source.startsWith(userdir))
    { //if the file is in PdfManager directory the pdfsource begin with currentstring
        this.pdfsource=
        System.getProperty("file.separator")+source.substring(userdir.length()+1);
    }
    else this.pdfsource=source;
}
}
}

```

5.1.4 EditSourceFrame.java

Ένα frame στο οποίο υπάρχουν τρεις επιλογές ανάλογα με το που βρισκόμαστε στο δέντρο των περιεχομένων.

- Εισαγωγή αρχείου που ήδη υπάρχει
- Δημιουργία αρχείου μέσω XML και XSL
- Εισαγωγή κάποιου υποκεφάλαιου

```

package pdfmanager;

/**
 * Title: PdfManager
 * Description: A program tha manage Pdf documents
 * Copyright: Copyright (c) 2003
 * Company: Ece Ntua
 * @Vlotomas Neofytos e197605
 * @version 1.0
 */

import java.awt.*;

```



```

import javax.swing.*;
import javax.swing.text.*;
import java.awt.event.*;
import javax.swing.filechooser.*;
import java.io.*;
import javax.swing.tree.*;
import javax.swing.event.*;
import com.lowagie.text.pdf.*;
import java.beans.*;

public class EditSourceFrame extends JFrame {
/*
Swing declarations
*/
//Swings for mainpanel and buttonpanel
private JRadioButton createpdf,findpdf,titlepdf;
private ButtonGroup group;
private JPanel mainpanel,buttonpanel;
private JTextPane maintext;
private JButton nextbutton1,backbutton1;
//Swings for titlepanel and its buttonpanel
private JButton okbutton,titlebuttonback,titlebuttonnext;
private JLabel labelname,pagelabel;
private JTextField pagetext,titletitle,titlename,titletext,helptext;
private JComboBox list;
private JPanel titlepanel,buttonpaneltitle;
//Swings for find pdf panel and its buttonpanel
private JPanel findpanel,buttonpanel2,comentpanel;
private JButton findbutton,nextbutton2,backbutton2;
private JLabel findtext,findlabel,findtitle;
private JTextField comenttext,nametext;
//Swings for Res panel and its buttonpanel
private JPanel ResPanel,buttonpanel3;
private JButton finishbutton,backbutton3;
private JTextField Results;
//Swings for create pdf panel and its buttonpanel
private JPanel createpanel,buttonpanel4;
private JButton findxml,findxsl,findsource,nextbutton4,backbutton4;
private JLabel labelxml,labelxsl,labelsource,name,chaptertitle;
private JTextField comentcreate,textname;
//Swings for Res panel 2 and its buttonpanel
private JPanel ResPanel2,buttonpanel5;
private JButton finishbutton2,backbutton5;
private JTextField Results2;
/*
global declarations

```

```

*/
private String userdir=System.getProperty("user.dir")
+ System.getProperty("file.separator");
private JFileChooser fc;
private File file,xmlfile,xslfile,pdffile;
private String title,source;
private int page=0;
private DefaultMutableTreeNode node;
private boolean valid=true;
//Some icons
private ImageIcon next = new ImageIcon("images/next.gif");
private ImageIcon back = new ImageIcon("images/back.gif");
private ImageIcon open = new ImageIcon("images/open.gif");
private JFrame frame;
private Font f1;
private Container contentPane ;
private String sourcepdffile;

public EditSourceFrame(DefaultMutableTreeNode nodes)
{
//Set the node to edit
this.node =nodes;
//Construct an Info for the current node
Object nodeInfo = node.getUserObject();
final Info Inf =(Info)nodeInfo;
f1=new Font("Arial",Font.BOLD,12);
frame = new JFrame("Edit a pdf document");
contentPane = frame.getContentPane();
contentPane.setLayout(new BorderLayout(20,20));
fc = new JFileChooser(userdir);
/*
Set the main panel (Enter a chapter)
*/
mainpanel=new JPanel();
maintext=new JTextPane();
//Set the RadioButtons to select
createpdf = new JRadioButton("Do you want to create a dynamic pdf document;");
createpdf.setMnemonic(KeyEvent.VK_B);
createpdf.setSelected(true);
findpdf = new JRadioButton("Do you want to find a pdf document;");
findpdf.setMnemonic(KeyEvent.VK_C);
titlepdf = new JRadioButton("Just a Title");
createpdf.setMnemonic(KeyEvent.VK_B);
createpdf.setSelected(true);
//Group the radio buttons.
group = new ButtonGroup();

```



```

findbar.add(findpdf);
findbar.setBorderPainted(false);
try
{
doc.insertString(doc.getLength(),
"          Enter a Chapter \n\n",
maintext.getStyle( "bold"));
doc.insertString(doc.getLength(),
"You have 2 choices:\n\n"+
"1.Create a new dynamic PDF document with XML-XSL files \n"+
"2.Edit a PDF document that already exists \n"+
"\n", maintext.getStyle( "regular"));
}
catch (BadLocationException ble)
{
System.err.println("Couldn't insert initial text.");
}

}
mainpanel.add("North",maintext);
mainpanel.add("Center",findbar);
//Set the Button panel of mainPanel
buttonpanel=new JPanel();
nextbutton1=new JButton(next);
//Listener for nextbutton1
nextbutton1.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
if (findpdf.isSelected())
{ //if we choose to find a Pdf file
mainpanel.setVisible(false);
buttonpanel.setVisible(false);
contentPane.remove(mainpanel);
contentPane.remove(buttonpanel);
contentPane.add("North",findpanel);
contentPane.add("South",buttonpanel2);
buttonpanel2.setVisible(true);
findpanel.setVisible(true);
}
else if (createpdf.isSelected())
{ //if we choose to ceate a pdf
mainpanel.setVisible(false);
buttonpanel.setVisible(false);
contentPane.remove(mainpanel);
contentPane.remove(buttonpanel);
}
}
}

```

```

contentPane.add("North",createpanel);
contentPane.add("South",buttonpanel4);
buttonpanel4.setVisible(true);
createpanel.setVisible(true);
}
else if (titlepdf.isSelected())
{//if we must edit a subchapter
mainpanel.setVisible(false);
buttonpanel.setVisible(false);
contentPane.remove(mainpanel);
contentPane.remove(buttonpanel);
contentPane.add("North",titlepanel);
contentPane.add("South",buttonpaneltitle);
titlepanel.setVisible(true);
buttonpaneltitle.setVisible(true);
}
}
});
backbutton1=new JButton(back);
buttonpanel.add("Center",backbutton1);
buttonpanel.add("West",nextbutton1);
backbutton1.setEnabled(false);
contentPane.add("South",buttonpanel);
buttonpanel.setBorder(BorderFactory.createLineBorder(Color.blue));
/*
Set the Title panel (Enter a subchapter)
*/
titlepanel=new JPanel();
titlepanel.setLayout(new BorderLayout());
JToolBar northbar=new JToolBar();
northbar.setBorderPainted(false);
northbar.setFloatable(false);
northbar.setOrientation(northbar.VERTICAL);
titletitle=new JTextField();
titletitle.setFont(fl);
titletitle.setForeground(Color.blue);
titletitle.setBorder(BorderFactory.createEmptyBorder());
titletitle.setBackground(mainpanel.getBackground());
titletitle.setEditable(false);
titletitle.setText("          Enter a subchapter\n");
northbar.add(titletitle);
titletext=new JTextField();
titletext.setFont(mainpanel.getFont());
titletext.setForeground(Color.blue);
titletext.setBorder(BorderFactory.createEmptyBorder());
titletext.setBackground(mainpanel.getBackground());

```

```

titletext.setEditable(false);
titletext.setText(" Please Enter name and page for the subchapter \n "+
"If you have more than one subchapter check the sequence \n");
northbar.add(titletext);
pagelabel=new JLabel("Number of pages ");
pagetext=new JTextField(3);
okbutton=new JButton("OK");
JToolBar titlebar=new JToolBar();
titlebar.setLayout(new BorderLayout(titlebar,BoxLayout.X_AXIS));
titlename=new JTextField(Inf.getName());
labelname=new JLabel("Name");
JToolBar namebar=new JToolBar();
namebar.setLayout(new BorderLayout(namebar,BoxLayout.X_AXIS));
String[] numpage=new String[1000];
//Make a list from 1 to 1000 to choose page
int h=1;
for(int y=0;y<1000;y++)
{
    numpage[y]=""+h;h++;
}
list = new JComboBox(numpage);
list.setSelectedIndex(0);

namebar.add(labelname);
namebar.add(titlename);
titlename.setMaximumSize(new Dimension (300,20));
namebar.setBorderPainted(false);
titlebar.add(pagelabel);
titlebar.add(list);
titlebar.setFloatable(false);
JPanel barpanel=new JPanel();
barpanel.setLayout(new BorderLayout());
barpanel.add("North",namebar);
barpanel.add("West",titlebar);
helptext=new JTextField();
helptext.setBorder(BorderFactory.createEmptyBorder());
helptext.setBackground(mainpanel.getBackground());
helptext.setEditable(false);
helptext.setText("\n\n");
titlebar.setBorderPainted(false);
titlepanel.add("North",northbar);
titlepanel.add("West",barpanel);
titlepanel.add("South",helptext);
// Button title Panel
titlebuttonback=new JButton(back);
//titlebuttonback listener

```

```

titlebuttonback.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Back to main Panel
  buttonpaneltitle.setVisible(false);
  titlepanel.setVisible(false);
  contentPane.remove(titlepanel);
  contentPane.remove(buttonpaneltitle);
  contentPane.add("North",mainpanel);
  contentPane.add("South",buttonpanel);
  mainpanel.setVisible(true);
  buttonpanel.setVisible(true);
}
});
titlebuttonnext=new JButton(next);
titlebuttonnext.addActionListener(new ActionListener()
{//titlebuttonnext Listener
public void actionPerformed(ActionEvent e)
{
  DefaultMutableTreeNode nodehelp=node;
  //find the chapter of the node
  while(nodehelp.getLevel()!=1)
  {
    nodehelp=nodehelp.getPreviousNode();
  }
  Object nodeInfohelp = nodehelp.getUserObject();
  Info perhelp=(Info)nodeInfohelp;
  try
  {
    PdfReader reader = new PdfReader(perhelp.getPdfSource());
    // we retrieve the total pagenumber of the chapter
    page = reader.getNumberOfPages();
  }
  catch(Exception ex)
  {
    System.err.println(ex.getMessage());
  }
  //our selection
  int select=list.getSelectedIndex()+1;
  if(select>page )
  { //if the page we select is grader than the pagenumber of the chapter
    JOptionPane.showMessageDialog(frame,"The number of page is out of bounds\nPlease
enter a valid page number",
    "Invalid page number", JOptionPane.ERROR_MESSAGE);
    return;
  }
}
}

```

```

nodehelp=node;
//Correct the sequense of the subchapters from bottom to top
while(nodehelp.getLevel()!=1)
{
nodehelp=nodehelp.getPreviousNode();
Object nodeInfo = nodehelp.getUserObject();
Info per=(Info)nodeInfo;
int test=per.getPages();
if(nodehelp.getLevel()==1)test=0;
if(select<test)
{
JOptionPane.showMessageDialog(frame,"The page sequence is incorrect \nPlease enter
valid page number",
"Invalid page number", JOptionPane.ERROR_MESSAGE);
return;
}
}
nodehelp=node;
//Correct the sequense of the subchapters from top to bottom
while(nodehelp.getNextNode()!=null && nodehelp.getNextNode().getLevel()!=1 )
{
nodehelp=nodehelp.getNextNode();
Object nodeInfo = nodehelp.getUserObject();
Info per=(Info)nodeInfo;
int test=per.getPages();
if(nodehelp.getLevel()!=1 && !per.untitled)
if(select>test)
{
JOptionPane.showMessageDialog(frame,"The page sequence is incorrect \nPlease
check the previous page and correct them",
"Invalid page number", JOptionPane.ERROR_MESSAGE);
frame.dispose();return;
}
}
//Add informations to the node
Inf.reName(titlename.getText());
Inf.page= list.getSelectedIndex()+1;
Inf.untitled=false;
JOptionPane.showMessageDialog(frame,"Subchapter added sucesfully",
"Successfull subchapter", JOptionPane.INFORMATION_MESSAGE);
frame.dispose();
}
});
buttonpaneltitle=new JPanel();

```



```

buttonpaneltitle.add("Center",titlebuttonback);
buttonpaneltitle.add("West",titlebuttonnext);
buttonpaneltitle.setBorder(BorderFactory.createLineBorder(Color.blue));
/*
Set the find pdf panel (Edit a chapter)
*/
findpanel=new JPanel();
findpanel.setLayout(new BorderLayout(10,10));
comentpanel=new JPanel();
comentpanel.setLayout(new BorderLayout(10,10));
findbutton=new JButton("Find",open);
//findbutton Listener
findbutton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Show only files with pdf extension
fc.addChoosableFileFilter(new Acceptedfiles("pdf"));
fc.setAcceptAllFileFilterUsed(false);
int returnVal = fc.showOpenDialog(EditSourceFrame.this);
if (returnVal == JFileChooser.APPROVE_OPTION)
{
file = fc.getSelectedFile();
pdffile=file;
findtext.setForeground(Color.red);
findtext.setText( file.getPath() );
}
else
{
findtext.setText("Open command cancelled by user.");
}
}
});
findtext= new JLabel(" Enter a pdf Source");
nametext=new JTextField(Inf.getName());
findtitle=new JLabel(" Enter a chapter");
findtitle.setBackground(findpanel.getBackground());
findtitle.setFont(f1);
findtitle.setBorder(BorderFactory.createEmptyBorder());
findtitle.setForeground(Color.blue);
comenttext=new JTextField();
comenttext.setEditable(false);
comenttext.setFont(findpanel.getFont());
comenttext.setForeground(Color.blue);
comenttext.setBackground(findpanel.getBackground());
comenttext.setText("\nStep 1:Enter Pdf Source that already exists \n"+
"Step 2:Then add name to the chapter.The page number will \n"+

```

```

"automatically included\n\n"+
"*Note:You can`t enter Pdf Documents with encryption ");
comenttext.setBorder( BorderFactory.createEmptyBorder());
comentpanel.add("North",findtitle);
comentpanel.add("Center",comenttext);
findpanel.add(findtext);
JToolBar findpdfbar=new JToolBar();
findpdfbar.setFloatable(false);
findpdfbar.setLayout(new BorderLayout(findpdfbar,BoxLayout.X_AXIS) );
findpdfbar.add(findbutton);
findpdfbar.add(findtext);
findpdfbar.setBorderPainted(false);
findlabel=new JLabel("Enter the name of the chapter");
JToolBar namepdfbar=new JToolBar();
namepdfbar.setFloatable(false);
namepdfbar.setLayout(new BorderLayout(namepdfbar,BoxLayout.X_AXIS) );
namepdfbar.add(findlabel);
namepdfbar.add(nametext);
namepdfbar.setBorderPainted(false);
findpanel.add("North",comentpanel);
findpanel.add("Center",namepdfbar);
findpanel.add("South",findpdfbar);
//button panel2
backbutton2=new JButton(back);
//backbutton2 Listener
backbutton2.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Back to main Panel
buttonpanel2.setVisible(false);
findpanel.setVisible(false);
contentPane.remove(findpanel);
contentPane.remove(buttonpanel2);
contentPane.add("North",mainpanel);
contentPane.add("South",buttonpanel);
mainpanel.setVisible(true);
buttonpanel.setVisible(true);
}
});
nextbutton2=new JButton(next);
//nextbutton2 Listener
nextbutton2.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Check our choices and go to next panel (ResPanel)
String comments;

```

```

//Set the title
title=nametext.getText();
//Set the pdf source
if (pdffile==null)
{
    source="Warning : You must enter a valid source";
    valid=false;
}
else
{
    valid=true;
    source=pdffile.getAbsolutePath();
}
//if our choices are not correct we cant edit the chapter
if (!valid)
{
    finishbutton.setEnabled(false);
}
else
{
    finishbutton.setEnabled(true);
    Inf.setPdfSource(source);
}
contentPane.add("South",buttonpanel3);

if (Inf.getPages()==-2)
{ //Not accesible files(probably encrypted)
    JOptionPane.showMessageDialog(frame,
    "The file is not accesible\nPlease choose another file", "Invalid File",
    JOptionPane.ERROR_MESSAGE);return;
}
else
{
    comments="Pages\nThe doc has "+Inf.getPages()+"Pages";
}
//Display Result panel
Inf.setPdfSource(null);
buttonpanel2.setVisible(false);
findpanel.setVisible(false);
ResPanel.setVisible(true);
buttonpanel3.setVisible(true);
contentPane.remove(findpanel);
contentPane.remove(buttonpanel2);
contentPane.add("North",ResPanel);
Results.setText("You have entered the
following:\n\nTitle:\n"+title+"\n\n"+"Source\n"+source+"\n\n"+comments);

```

```

    Results.setEditable(false);
}
});

buttonpanel2=new JPanel();
buttonpanel2.add("Center",backbutton2);
buttonpanel2.add("West",nextbutton2);
buttonpanel2.setBorder(BorderFactory.createLineBorder(Color.blue));
/*
Set the Res Panel
*/
ResPanel=new JPanel();
Results=new JTextField();
ResPanel.setLayout(new GridLayout(1,0));
Results.setBackground(ResPanel.getBackground());
Results.setForeground(Color.blue);
Results.setBorder(BorderFactory.createEmptyBorder());
ResPanel.add(Results);
ResPanel.setBorder( BorderFactory.createEmptyBorder());
Results.setEditable(false);
//Button panel3
backbutton3=new JButton(back);
//backbutton3 Listener
backbutton3.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Go back to findpdf panel
    buttonpanel3.setVisible(false);
    ResPanel.setVisible(false);
    contentPane.remove(ResPanel);
    contentPane.remove(buttonpanel3);
    contentPane.add("North",findpanel);
    contentPane.add("South",buttonpanel2);
    findpanel.setVisible(true);
    buttonpanel2.setVisible(true);
}
});
finishbutton=new JButton("FINISH");
//finishbutton Listener
finishbutton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Set information for the node and dispose the frame
    Inf.reName(title);
    Inf.untitled=false;
    Inf.setPdfSource(source);
}
});

```

```

    frame.dispose();
}
});

buttonpanel3=new JPanel();
buttonpanel3.add(backbutton3);
buttonpanel3.add(finishbutton);
buttonpanel3.setBorder(BorderFactory.createLineBorder(Color.blue));
/*
Set the Create pdf panel
*/
createpanel=new JPanel();
createpanel.setLayout(new BorderLayout(createpanel,BoxLayout.Y_AXIS));
JToolBar xmlbar=new JToolBar();
xmlbar.setFloatable(false);
xmlbar.setLayout(new BorderLayout() );
xmlbar.setBorderPainted(false);
JToolBar xslbar=new JToolBar();
xslbar.setFloatable(false);
xslbar.setLayout(new BorderLayout() );
xslbar.setBorderPainted(false);
JToolBar pdfbar=new JToolBar();
pdfbar.setFloatable(false);
pdfbar.setLayout(new BorderLayout() );
pdfbar.setBorderPainted(false);
JToolBar namecreatebar=new JToolBar();
namecreatebar.setFloatable(false);
namecreatebar.setLayout(new BorderLayout() );
namecreatebar.setBorderPainted(false);
findxml=new JButton(" XML ",open);
//findxml Listener
findxml.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Show only files with xml extension
fc.addChoosableFileFilter(new Acceptedfiles("xml"));
fc.setAcceptAllFileFilterUsed(false);
int returnVal = fc.showOpenDialog(EditSourceFrame.this);
if (returnVal == JFileChooser.APPROVE_OPTION)
{
file = fc.getSelectedFile();
xmlfile=file;
labelxml.setForeground(Color.red);
labelxml.setText( file.getPath() );
}
else

```

```

    {
        labelxml.setText("Open command cancelled by user.");
    }
}
});
findxsl=new JButton(" XSL ",open);
//findxsl Listener
findxsl.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Show only files with xsl extension
fc.addChoosableFileFilter(new Acceptedfiles("xsl"));
fc.setAcceptAllFileFilterUsed(false);
int returnVal = fc.showOpenDialog(EditSourceFrame.this);
if (returnVal == JFileChooser.APPROVE_OPTION)
{
file = fc.getSelectedFile();
xslfile=file;
labelxsl.setForeground(Color.red);
labelxsl.setText( file.getPath() );
}
else
{
labelxml.setText("Open command cancelled by user." );
}
}
});

findsource=new JButton("Source",open);
//findsource Listener
findsource.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Show only files with pdf extension
fc.addChoosableFileFilter(new Acceptedfiles("pdf"));
fc.setAcceptAllFileFilterUsed(false);
int returnVal = fc.showOpenDialog(EditSourceFrame.this);
if (returnVal == JFileChooser.APPROVE_OPTION)
{
file = fc.getSelectedFile();
pdffile=file;
if(!file.getPath().endsWith(".pdf") && !file.getPath().startsWith("Enter a new pdf file"))
{
sourcepdffile=file.getPath()+".pdf";
}
}
else

```

```

    sourcepdffile=file.getPath();
//if file already exists display a message
if(file.exists())
{
    final JFrame message=new JFrame();
    final JOptionPane optionPane = new JOptionPane( "File already exists\nDo you want to
replase it;\n"
,JOptionPane.QUESTION_MESSAGE, JOptionPane.YES_NO_OPTION);
    final JDialog dialog = new JDialog(message, "File already exist",true);
    optionPane.addPropertyChangeListener( new PropertyChangeListener()
    {
        public void propertyChange(PropertyChangeEvent e)
        {
            String prop = e.getPropertyName();
            if (dialog.isVisible()&& (e.getSource() == optionPane)&&
(prop.equals(JOptionPane.VALUE_PROPERTY)
||prop.equals(JOptionPane.INPUT_VALUE_PROPERTY)))
            {
                dialog.setVisible(false);
            }
        }
    });
    dialog.setDefaultCloseOperation(JDialog.DO_NOTHING_ON_CLOSE);
    dialog.addWindowListener(new WindowAdapter()
    {
        public void windowClosing(WindowEvent we)
        {

        }

    });

    dialog.setContentPane(optionPane);
    dialog.pack();
    dialog.setLocation(200, 200);
    dialog.setVisible(true);

    int value = ((Integer)optionPane.getValue()).intValue();
    if (value == JOptionPane.NO_OPTION)
    { //if we choose enter a new file
        pdffile=null;
        sourcepdffile="Enter a new pdf file";
        message.dispose();
    }//else replace the file
    else if(value == JOptionPane.YES_OPTION)
    {
        message.dispose();
    }
}

```

```

    }

}
labelsource.setForeground(Color.red);
labelsource.setText(sourcepdffile );
}
else
{
    labelsource.setText("Open command cancelled by user.");
}
}
});
labelxml=new JLabel("Please enter a valid xml source");
labelxsl=new JLabel("Please enter a valid xsl source");
labelsource=new JLabel("Please enter a valid pdf source");
textname=new JTextField(Inf.getName());
name=new JLabel("Please Enter a name ");
comentcreate=new JTextField("\nStep 1:Enter Xml-Xsl files and the output pdf\n"+
"\nStep 2:Then enter a name for the chapter\n ");
comentcreate.setEditable(false);
comentcreate.setFont(findpanel.getFont());
comentcreate.setForeground(Color.blue);
comentcreate.setBackground(findpanel.getBackground());
comentcreate.setBorder( BorderFactory.createEmptyBorder());
xmlbar.add("West",findxml);
xmlbar.add("Center",labelxml);
xslbar.add("West",findxsl);
xslbar.add("Center",labelxsl);
pdfbar.add("West",findsource);
pdfbar.add("Center",labelsource);
namecreatebar.add("West",name);
namecreatebar.add("Center",textname);
JToolBar chartertitlebar=new JToolBar();
chartertitlebar.setFloatable(false);
chartertitlebar.setBorderPainted(false);
chartertitlebar.setBackground(findpanel.getBackground());
chartertitle=new JLabel("Enter a chapter");
chartertitle.setFont(f1);
chartertitle.setForeground(Color.blue);
chartertitle.setBackground(findpanel.getBackground());
chartertitlebar.add(chartertitle);
chartertitlebar.setAlignmentX(chartertitlebar.LEFT);
createpanel.add(chartertitlebar);
createpanel.add(comentcreate);
createpanel.add(xmlbar);
createpanel.add(xslbar);

```



```

createpanel.add(pdfbar);
createpanel.add(namecreatebar);
//Button panel4
buttonpanel4=new JPanel();
nextbutton4=new JButton(next);
//nextbutton4 Listener
nextbutton4.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Transform xml/xsl files to pdf
String comments;
JDialog dialog = new JDialog(frame, "Wait For The Transformation", false);
//Call PdfCreator class
PdfCreator creator = new PdfCreator();
//Test the file we enter and we pass them to PdfCreator class
if(xmlfile==null || !xmlfile.toString().endsWith(".xml"))
{
JOptionPane.showMessageDialog(frame,"Invalid XML File", "Invalid File",
JOptionPane.INFORMATION_MESSAGE);
}
else creator.setXmlFile(xmlfile.getPath());
if(xslfile==null || !xslfile.toString().endsWith(".xsl"))
{
JOptionPane.showMessageDialog(frame,"Invalid XSL File", "Invalid File",
JOptionPane.INFORMATION_MESSAGE);
}
else creator.setXsltFile(xslfile.getPath());
if(pdffile==null || !sourcepdffile.endsWith(".pdf"))
{
JOptionPane.showMessageDialog(frame,"Invalid PDF File", "Invalid File",
JOptionPane.INFORMATION_MESSAGE);
}
else creator.setOutputFile(pdffile.getPath());
//if our files are corect continue with transformation
if(xmlfile!=null&&xslfile!=null&&pdffile!=null&&
xmlfile.toString().endsWith(".xml")&&xslfile.toString().endsWith(".xsl") &&
pdffile.toString().endsWith(".pdf"))
try
{
source=pdffile.getPath();
buttonpanel4.setVisible(false);
createpanel.setVisible(false);
ResPanel2.setVisible(true);
buttonpanel5.setVisible(true);
contentPane.remove(createpanel);
contentPane.remove(buttonpanel4);
}
}
}

```

```

contentPane.add("Center",ResPanel2);
title="";
contentPane.add("South",buttonpanel5);
dialog.getContentPane().add(new JLabel("Reading options"));
dialog.setSize(300, 150);
dialog.setLocation(200, 200);
dialog.setVisible(true);
//Transform files dynamicly
creator.transform();
JOptionPane.showMessageDialog(frame,
"transformation Successful" + labelsource.getText(),
"transformationSuccessful",
JOptionPane.INFORMATION_MESSAGE);
}
catch(Exception ex)
{
JOptionPane.showMessageDialog(frame,"Exception" + ex,
"EXCEPTION", JOptionPane.ERROR_MESSAGE);
System.err.println(ex.getMessage());
}
catch(Error ex)
{
JOptionPane.showMessageDialog(frame,"Error" + ex,
"Error", JOptionPane.ERROR_MESSAGE);
System.err.println(ex.getMessage());
}
//show information in Res Panel2
dialog.setVisible(false);
title=textname.getText();
Inf.setPdfSource(source);
Results2.setText("You entered the
following:\n\nTitle:\n"+title+"\n\n"+"Pages:\n"+Inf.getPages()+"\n\n"+
"Source\n"+source);
Inf.pdfsource=null;
Inf.page=0;
Results2.setEditable(false);
}
});
backbutton4=new JButton(back);
//backbutton4 Listener
backbutton4.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //back to main panel
buttonpanel4.setVisible(false);
createpanel.setVisible(false);

```

```

contentPane.remove(createpanel);
contentPane.remove(buttonpanel4);
contentPane.add("North",mainpanel);
contentPane.add("South",buttonpanel);
mainpanel.setVisible(true);
buttonpanel.setVisible(true);
}
});
buttonpanel4.add(backbutton4);
buttonpanel4.add(nextbutton4);
buttonpanel4.setBorder(BorderFactory.createLineBorder(Color.blue));
/*
Set the Res panel 2
*/
ResPanel2=new JPanel();
Results2=new JTextField();
Results2.setAlignmentX(Results2.LEFT_ALIGNMENT);
Results2.setBorder(BorderFactory.createEmptyBorder());
ResPanel2.setLayout(new BorderLayout(ResPanel2,BoxLayout.X_AXIS));
Results2.setBackground(ResPanel.getBackground());
Results2.setForeground(Color.blue);
ResPanel2.add(Results2);
ResPanel2.setBorder( BorderFactory.createEmptyBorder());
Results2.setEditable(false);
//Button panel5
buttonpanel5=new JPanel();
backbutton5=new JButton(back);
//backbutton4 Listener
backbutton5.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //back to create pdf panel
buttonpanel5.setVisible(false);
ResPanel2.setVisible(false);
contentPane.remove(ResPanel2);
contentPane.remove(buttonpanel5);
contentPane.add("North",createpanel);
contentPane.add("South",buttonpanel4);
createpanel.setVisible(true);
buttonpanel4.setVisible(true);
}
});
finishbutton2=new JButton("FINISH");
//finishbutton2 Listener
finishbutton2.addActionListener(new ActionListener()
{

```

```

public void actionPerformed(ActionEvent e)
{ ///Set information for the node and dispose the frame
  Inf.reName(title);
  Inf.setPdfSource(source);
  Inf.page=Inf.getPages();
  Inf.untitled=false;
  frame.dispose();
}
});

buttonpanel5.add(backbutton5);
buttonpanel5.add(finishbutton2);
buttonpanel5.setBorder(BorderFactory.createLineBorder(Color.blue));
/*
Lay everything to Frame
*/
frame.setSize(400,300);
frame.setVisible(true);
frame.setLocation(200,200);
try
{
  UIManager.setLookAndFeel( UIManager.getSystemLookAndFeelClassName() );
}
catch( Exception e )
{
  e.printStackTrace();
}
}

//Initialize some styles.
protected void initStylesForTextPane(JTextPane text)
{
  Style def = StyleContext.getDefaultStyleContext().
  getStyle(StyleContext.DEFAULT_STYLE);

  Style regular = text.addStyle("regular", def);
  StyleConstants.setFontFamily(def, "SansSerif");

  Style s = text.addStyle("bold", regular);
  StyleConstants.setBold(s, true);

  s = text.addStyle("italic", regular);
  StyleConstants.setItalic(s, true);

  s = text.addStyle("underline", regular);

```

```

StyleConstants.setUnderline(s,true);

s = text.addStyle("bold-underline", regular);
StyleConstants.setBold(s, true);
StyleConstants.setUnderline(s,true);
}
}

```

5.1.5 DynamicTree.java

Η κλάση αυτή δημιουργεί και διαχειρίζεται το δέντρο των περιεχομένων package pdfmanager;

```

/**
 * Title: PdfManager
 * Description: A program tha manage Pdf documents
 * Copyright: Copyright (c) 2003
 * Company: Ece Ntua
 * @Vlotomas Neofytos el97605
 * @version 1.0
 */

import java.awt.*;
import javax.swing.*;
import javax.swing.tree.*;
import javax.swing.event.*;

public class DynamicTree extends JPanel
{
    /*
    Declarations for the tree
    */
    protected DefaultMutableTreeNode rootNode;
    protected DefaultTreeModel treeModel;
    protected JTree tree;
    private Toolkit toolkit = Toolkit.getDefaultToolkit();
    public boolean changed=false;
    public String information;

    public      DynamicTree(DefaultMutableTreeNode      rootNode,DefaultTreeModel
treeModel,final JTree tree)
    {
        this.rootNode=rootNode;
        this.tree=tree;
        this.treeModel=treeModel;
    }

```

```

JScrollPane scrollPanetree = new JScrollPane(tree);
setLayout(new GridLayout(1,0));
add(scrollPanetree);
}

// Remove all nodes except the root node.
public void clear()
{
    rootNode.removeAllChildren();
    treeModel.reload();
}

// Remove the currently selected node.
public void removeCurrentNode()
{
    TreePath currentSelection = tree.getSelectionPath();
    if (currentSelection != null)
    {
        DefaultMutableTreeNode currentNode = (DefaultMutableTreeNode)
(currentSelection.getLastPathComponent());
        MutableTreeNode parent = (MutableTreeNode)(currentNode.getParent());
        if (parent != null)
        {
            treeModel.removeNodeFromParent(currentNode);
            changed=true;
            return;
        }
    }
    // Either there was no selection, or the root was selected, beep
    toolkit.beep();
}

// Add child to the currently selected node.
public DefaultMutableTreeNode addObject(Object child)
{
    DefaultMutableTreeNode parentNode = null;
    TreePath parentPath = tree.getSelectionPath();
    if (parentPath == null)
    {
        parentNode = rootNode;
    }
    else
    {
        //Set the parent
        parentNode = (DefaultMutableTreeNode)(parentPath.getLastPathComponent());
    }
    return addObject(parentNode, child, true);
}

```

```

}

public DefaultMutableTreeNode addObject(DefaultMutableTreeNode parent, Object child)
{
    return addObject(parent, child, false);
}

public DefaultMutableTreeNode addObject(DefaultMutableTreeNode parent, Object
child, boolean shouldBeVisible)
{
    DefaultMutableTreeNode childNode = new DefaultMutableTreeNode(child);
    if (parent == null)
    {
        parent = rootNode;
    }
    treeModel.insertNodeInto(childNode, parent, parent.getChildCount());

    if (shouldBeVisible)
    {
        tree.scrollPathToVisible(new TreePath(childNode.getPath()));
    }
    return childNode;
}
}
}

```

5.1.6 DisplayUrl.java

Είναι ένα frame που βοηθά στην προβολή κάποιου URL

```

package pdfmanager;

/** * Title: PdfManager
 * Description: A program tha manage Pdf documents
 * Copyright: Copyright (c) 2003
 * Company: Ece Ntua
 * @Vlotomas Neofytos el97605
 * @version 1.0
 */

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.*;
import java.net.URL;
import java.applet.*;

```

```

public class DisplayUrl extends JFrame
{
/*
Swing declarations
*/
private JPanel centerPanel;
private JEditorPane text;
private JFrame frame;
/*
global Declarations
*/
private String window;
private String userHome=System.getProperty("user.home") +
System.getProperty("file.separator");

public DisplayUrl(String window )
{
this.window=window;
frame = new JFrame();
final Container contentPane = frame.getContentPane();
contentPane.setLayout(new BorderLayout(20,20));
//Center Panel
centerPanel = new JPanel(new GridLayout(1,0));
//Add Html initial page
try
{
text=new JEditorPane();
text.setEditable(false);
String s="";
//display credits
if (window.equals("credits"))
{
s = "file:"
+ System.getProperty("user.dir")
+ System.getProperty("file.separator")
+ "credits.html";
}
URL helpURL = new URL(s);

try
{ //set url
text.setPage(helpURL);
}
catch (IOException e)
{

```



```

    JOptionPane.showMessageDialog(frame,"Couldn't connect to the URL:",
    "Bad URL ", JOptionPane.ERROR_MESSAGE);
    return;
}
}
catch (Exception e)
{
    JOptionPane.showMessageDialog(frame,"Couldn't create URL: ",
    "Bad URL ", JOptionPane.ERROR_MESSAGE);
    return;
}
//Lay everything to Frame
JScrollPane editorScrollPane = new JScrollPane(text);
editorScrollPane.setVerticalScrollBarPolicy(
JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);
editorScrollPane.setPreferredSize(new Dimension(500, 400));
editorScrollPane.setMinimumSize(new Dimension(500,400));
centerPanel.add(editorScrollPane);
contentPane.add("Center",centerPanel);
JButton ok=new JButton("OK");
JPanel okpanel=new JPanel();
okpanel.setLayout(new BorderLayout(okpanel,BoxLayout.X_AXIS));
okpanel.add(ok);
contentPane.add("South",okpanel);
ok.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        frame.dispose();
    }
});
frame.setSize(400,300);
frame.setVisible(true);
frame.setLocation(200,200);
}
}

```

5.1.7 CreatePdfFrame.java

Ένα frame που έχει σαν σκοπό να εισάγει δεδομένα στο τελικό αρχείο που θέλουμε να δημιουργήσουμε :

- Που θα δημιουργηθεί (Source)
- Επιλογή εξώφυλλου (cover)
- Δημιουργία περιεχομένων
- Εισαγωγή Header- Footer
- Πως θα παρουσιάζεται (View Preferences)
- Απόκρυψη του αρχείου (Encryption)
- Επιπρόσθετα στοιχεία του αρχείου (Summary)

```
package pdfmanager;
```

```
/**
```

```
* Title: PdfManager
```

```
* Description: A program tha manage Pdf documents
```

```
* Copyright: Copyright (c) 2003
```

```
* Company: Ece Ntua
```

```
* @Vlotomas Neofytos e197605
```

```
* @version 1.0
```

```
*/
```

```
import java.awt.*;
```

```
import javax.swing.*;
```

```
import javax.swing.tree.*;
```

```
import java.awt.event.*;
```

```
import javax.swing.filechooser.*;
```

```
import java.io.*;
```

```
import java.beans.*;
```

```
public class CreatePdfFrame extends JFrame
```

```
{
```

```
/**
```

```
Swing Declarations
```

```
*/
```

```
private JCheckBox copyBox, printBox, hideToolBarBox, hideMenuebarBox,
```

```
encryptBox,pagecountbox,pagecountbox2,modifycontbox,modifyannotbox,fillinbox,
```

```
screenreadersbox,assemblybox,contentsbox;
```

```
private JTabbedPane tabbed;
```

```
private JButton findbutton,cancelbutton,okbutton,coverbutton;
```

```
private JTextField findtext;
```

```
private JComboBox list;
```

```
private JToolBar output ;
```

```

private JPanel viewpanel,viewcenter,vieweast,viewwest;
private JPanel editpanel ,mainsouth, editcenter,editsouth,editnorth;
private JPanel encryptpanel,enccenter,encsouth,summary;
private JFileChooser fc;
private JTextField text,header,footer,enccomment;
private JTextField titlepdf,key,subject,author,creator;
private JFrame frame;
private JLabel comentheader,comentfooter,pagelayout,pagemode,coverlabel;
private ButtonGroup group,group2;
private JRadioButton none,outlines,thumbs,fullscreen;
private JRadioButton single,one,twoleft,tworight;
/*
global declarations
*/
private File outputfile;
private String outputfilestring;
private String userpass=null;
private String ownerpass=null;
private String source=null;
private String arg[]=new String[200];
private boolean untitled[]=new boolean[200];
private JPasswordField jp,jo;
private int page[]=new int[200];
private int level[]=new int[200];
private int allpage;
private String title[]=new String[200];
private DefaultMutableTreeNode rootNode;
private String userdir=System.getProperty("user.dir")
+ System.getProperty("file.separator");
private Font f1,f2;
private ImageIcon open;
private String cover="";

public CreatePdfFrame( DefaultMutableTreeNode root)
{
//set the root of the tree
rootNode=root;
f1=new Font("TimesRoman",Font.PLAIN,18);
f2=new Font("TimesRoman",Font.BOLD,12);
open = new ImageIcon("images/open.gif");
frame = new JFrame("Create a pdf document");
final Container contentPane = frame.getContentPane();
contentPane.setLayout(new BorderLayout(20,20));
fc = new JFileChooser(userdir);
//Construct a TabbedPane to put several panels
tabbed=new JTabbedPane() ;

```

```

/*
Edit Panel
*/
editpanel =new JPanel();
editpanel.setLayout(new BorderLayout(editpanel,BorderLayout.Y_AXIS));
tabbed.addTab("Edit",editpanel);
editcenter = new JPanel(new GridLayout(4,0));
editnorth=new JPanel(new GridLayout(3,0));
editsouth=new JPanel(new GridLayout(2,0));
output = new JToolBar();
output.setFloatable(false);
findbutton=new JButton("Output",open);
//findbutton Listener
findbutton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{ //Show only files with pdf extension
fc.addChoosableFileFilter(new Acceptedfiles("pdf"));
fc.setAcceptAllFileFilterUsed(false);
int returnVal = fc.showOpenDialog(CreatePdfFrame.this);
if (returnVal == JFileChooser.APPROVE_OPTION)
{
outputfile = fc.getSelectedFile();
outputfilestring=outputfile.getPath();
//if the file already exists display a message
if(outputfile.exists())
{
final JFrame message=new JFrame();
final JOptionPane optionPane = new JOptionPane( "File already exists\nDo you want to
replase it;\n"
,JOptionPane.QUESTION_MESSAGE, JOptionPane.YES_NO_OPTION);
final JDialog dialog = new JDialog(message, "File already exist",true);
optionPane.addPropertyChangeListener(new PropertyChangeListener()
{
public void propertyChange(PropertyChangeEvent e)
{
String prop = e.getPropertyName();
if (dialog.isVisible())&& (e.getSource() == optionPane)&&
(prop.equals(JOptionPane.VALUE_PROPERTY)
||prop.equals(JOptionPane.INPUT_VALUE_PROPERTY)))
{
dialog.setVisible(false);
}
}
});
dialog.setDefaultCloseOperation(JDialog.DO_NOTHING_ON_CLOSE);

```

```

dialog.addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent we)
{

}
});

dialog.setContentPane(optionPane);
dialog.pack();
dialog.setLocation(300, 300);
dialog.setVisible(true);

int value = ((Integer)optionPane.getValue()).intValue();
//if we choose no we must enter a new file
if (value == JOptionPane.NO_OPTION)
{
    outputfile=null;
    outputfilestring="Enter a new pdf file";
    message.dispose();
}
//if we choose use continue
else if(value == JOptionPane.YES_OPTION)
{
    message.dispose();
}

}
if(!outputfilestring.endsWith(".pdf")&& !outputfilestring.startsWith("Enter a new pdf
file"))
{
    outputfilestring =outputfilestring+".pdf";
}
if(!outputfilestring.startsWith("Enter a new pdf file"))    arg[0]=outputfilestring;
    findtext.setText(outputfilestring);
}
else
{
    findtext.setText("Open command cancelled by user." );
}
}
});
findtext=new JTextField();
findtext.setEditable(false);
findtext.setBorder(BorderFactory.createEmptyBorder());
output.add(findbutton);

```

```

output.add(findtext);
output.setBorder(BorderFactory.createRaisedBevelBorder());
findtext.setBackground(editcenter.getBackground());
JToolBar contentbar=new JToolBar();
contentbar.setFloatable(false);
contentbar.setBorderPainted(false);
contentsbox=new JCheckBox("Create contents ");
contentsbox.setSelected(true);
contentbar.add(contentsbox);
coverbutton=new JButton("Add cover");
coverlabel=new JLabel();
coverlabel.setForeground(Color.blue);
coverlabel.setBorder(BorderFactory.createEmptyBorder());
///  

coverbutton.addActionListener(new ActionListener()
{
//Select a file for the cover
public void actionPerformed(ActionEvent e)
{
//Show only files with pdf extension
fc.addChoosableFileFilter(new Acceptedfiles("pdf"));
fc.setAcceptAllFileFilterUsed(false);
int returnVal = fc.showOpenDialog(CreatePdfFrame.this);
if (returnVal == JFileChooser.APPROVE_OPTION)
{
cover=fc.getSelectedFile().getPath();
coverlabel.setText(cover);
}
else
{
coverlabel.setText("Open command cancelled by user." );
}
}
});
contentbar.add(coverbutton);
contentbar.add(coverlabel);
editsouth.add(contentbar);
editsouth.add(output);
//Set Footer choice
pagecountbox=new JCheckBox("Footer Page Count ");
footer=new JTextField();
JToolBar footerbar=new JToolBar();
footerbar.setOrientation(footerbar.HORIZONTAL);
footerbar.setFloatable(false);
footerbar.add(new JLabel("Footer Coments "));
footerbar.setBorderPainted(false);

```

```

footerbar.add(footer);
footerbar.add(new JLabel(""));
comentfooter=new JLabel(" You can add footer comments and page count");
comentfooter.setForeground(Color.blue);
//Set the language for the contents
String language[]=new String[2];
language[0]="Selected Language (Greek)";
language[1]="Selected Language (English)";
list = new JComboBox(language);
list.setSelectedIndex(0);
JToolBar langbar=new JToolBar();
langbar.setFloatable(false);
langbar.setBorderPainted(false);
JLabel langcomments=new JLabel("You can change the selected Language for the contents
");
langcomments.setForeground(Color.blue);
langbar.add(langcomments);
langbar.add(list);
langbar.add(new JLabel(""));
editcenter.add(comentfooter);
editcenter.add(footerbar);
editcenter.add(pagecountbox);
editcenter.add(langbar);
//Set Header choice
header =new JTextField();
pagecountbox2=new JCheckBox("Header Page Count ");
JToolBar headerbar=new JToolBar();
headerbar.setFloatable(false);
headerbar.add(new JLabel("Header Coments "));
headerbar.add(header);
headerbar.setBorderPainted(false);
headerbar.add(new JLabel(""));
JToolBar commentsbar=new JToolBar();
commentsbar.setFloatable(false);
comentheader=new JLabel(" You can add header comments and page count ");
comentheader.setForeground(Color.blue);
commentsbar.add(comentheader);
commentsbar.setBorderPainted(false);
editnorth.add(commentsbar);
editnorth.add(headerbar);
editnorth.add(pagecountbox2);
String coments="You can enter properties to the document";
text=new JTextField(coments);
text.setEditable(false);
text.setBackground(editcenter.getBackground());
//Lay everything to edit panel

```

```

editpanel.setBorder(BorderFactory.createLineBorder(Color.blue));
editpanel.add(editnorth);
editpanel.add(editcenter);
editpanel.add(editsouth);
/*
View Panel
*/
viewpanel=new JPanel();
viewpanel.setLayout(new BorderLayout(20,20));
tabbed.addTab("View Preferences",viewpanel);
viewcenter=new JPanel(new GridLayout(2,1));
hideToolBarBox = new JCheckBox("hideToolBarBox");
hideMenuebarBox = new JCheckBox("hideMenuebarBox");
viewcenter.add(hideToolBarBox);
viewcenter.add(hideMenuebarBox);
viewpanel.setBorder(BorderFactory.createLineBorder(Color.blue));
//Page mode choices
vieweast=new JPanel();
vieweast.setLayout(new GridLayout(5,1 ));
none = new JRadioButton("No Page mode");
outlines = new JRadioButton("Document outline visible");
thumbs = new JRadioButton("Thumbnail images visible");
fullscreen = new JRadioButton("Full-screen mode");
outlines.setSelected(true);
//Group the radio buttons.
group = new ButtonGroup();
group.add(none);
none.setActionCommand("none");
group.add(outlines);
outlines.setActionCommand("outline");
group.add(thumbs);
thumbs.setActionCommand("thumb");
group.add(fullscreen);
fullscreen.setActionCommand("fullscreen");
pagemode=new JLabel("Enter the Page Mode");
pagemode.setForeground(Color.blue);
pagemode.setFont(f2);
vieweast.add(pagemode);
vieweast.add(none);
vieweast.add(outlines);
vieweast.add(thumbs);
vieweast.add(fullscreen);
vieweast.setBorder(BorderFactory.createEtchedBorder());
viewpanel.add("Center",viewcenter);
viewpanel.add("East",vieweast);
//Page Layout choices

```



```

viewwest=new JPanel();
viewwest.setLayout(new GridLayout(5,1 ));
single = new JRadioButton(" Display one page at a time ");
single.setActionCommand("single");
one = new JRadioButton("Display the pages in one column");
one.setActionCommand("one");
twoleft = new JRadioButton("Two columns, with pages on the left.");
twoleft.setActionCommand("twoleft");
tworight = new JRadioButton("Two columns, with pages on the right.");
tworight.setActionCommand("tworight");
group2=new ButtonGroup();
single.setSelected(true);
group2.add(one);
group2.add(single);
group2.add(twoleft);
group2.add(tworight);
pagelayout=new JLabel("Page Layout");
pagelayout.setFont(f2);
pagelayout.setForeground(Color.blue);
viewwest.add(pagelayout);
viewwest.add(single);
viewwest.add(one);
viewwest.add(twoleft);
viewwest.add(tworight);
viewwest.setBorder(BorderFactory.createEtchedBorder());
//Lay everything to viewpanel
viewpanel.add("West",viewwest);
viewpanel.add("Center",viewcenter);
viewpanel.add("East",vieweast);
/*
Encryption Panel
*/
encryptpanel=new JPanel();
encryptpanel.setLayout(new BorderLayout(20,20));
tabbed.addTab("Encryption", encryptpanel);
enccenter =new JPanel(new GridLayout(4,0));
encsouth=new JPanel(new GridLayout(1,0));
//Set choices
copyBox = new JCheckBox("Allow copy");
printBox = new JCheckBox("Allow print");
modifycontbox =new JCheckBox("Allow Modify Contents");
modifyannotbox =new JCheckBox("Allow Motify Annotations");
fillinbox= new JCheckBox("Allow Fill in");
screenreadersbox= new JCheckBox("Allow ScreenReader");
assemblybox= new JCheckBox("Allow Assembly");
encryptBox = new JCheckBox("Set Passwords");

```

```

//encryptBox Listener
//if we choose to set passwords the programm will create a new frame
encryptBox.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
if (encryptBox.isSelected())
{
final JFrame frame2=new JFrame("Set the Passwords");
Container contentPane2 = frame2.getContentPane();
contentPane2.setLayout(new GridLayout(3,2));
jp = new JPasswordField(10);
JLabel jplabel=new JLabel("Please enter user Password");
jplabel.setLabelFor(jp);
jo = new JPasswordField(10);
JLabel jolabel=new JLabel("Please enter owner Password");
jolabel.setLabelFor(jo);
JButton ok=new JButton("OK");
ok.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
if (jp.getPassword().equals(""))userpass=null;
else userpass = new String(jp.getPassword());
if (jo.getPassword().equals(""))ownerpass=null;
else ownerpass =new String( jo.getPassword());
frame2.dispose();
}
});
JButton cancel=new JButton("Cancel");
cancel.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
encryptBox.setSelected(false);
frame2.dispose();
}
});
contentPane2.add(jplabel);
contentPane2.add(jp);
contentPane2.add(jolabel);
contentPane2.add(jo);
contentPane2.add(ok);
contentPane2.add(cancel);
//Show Frame for the passwords
frame2.setSize(350,130);

```

```

    frame2.setVisible(true);
    frame2.setLocation(200,200);
}
else return;
}
});
enccenter.add(copyBox);
enccenter.add(modifycontbox);
enccenter.add(printBox);
enccenter.add(modifyannotbox);
enccenter.add(fillinbox);
enccenter.add(screenreadersbox);
enccenter.add(assemblybox);
enccenter.add(encryptBox);
enccoment=new JTextField();
enccoment.setText("You can set the encryption of the document. "+
"Setting an encryption protects your documents from other users \n\n");
enccoment.setEditable(false);
enccoment.setBorder(BorderFactory.createEmptyBorder());
enccoment.setBackground(enccenter.getBackground());
enccoment.setForeground(Color.blue);
encsouth.add(enccoment);
//Lay everything to encryption panel
encryptpanel.add("Center",enccenter);
encryptpanel.add("South",encsouth);
encryptpanel.setBorder(BorderFactory.createLineBorder(Color.blue));
/*
Summary Panel
*/
summary=new JPanel();
summary.setLayout(new GridLayout(5,1,10,10));
//Set Title
JToolBar titlebar=new JToolBar();
titlebar.setFloatable(false);
titlebar.setBorderPainted(false);
JLabel titlelabel=new JLabel( "   Enter Title:");
titlebar.add(titlelabel);
titlepdf=new JTextField();
titlebar.add(titlepdf);
titlebar.add(new JLabel("           "));
//Set Subject
JToolBar subjectbar=new JToolBar();
subjectbar.setFloatable(false);
subjectbar.setBorderPainted(false);
subject=new JTextField();
JLabel subjectlabel=new JLabel("Enter Subject:");

```

```

subjectbar.add(subjectlabel);
subjectbar.add(subject);
subjectbar.add(new JLabel("          "));
//Set Author
JToolBar authorbar=new JToolBar();
authorbar.setFloatable(false);
authorbar.setBorderPainted(false);
author=new JTextField();
JLabel authorlabel=new JLabel(" Enter Author:");
authorbar.add(authorlabel);
authorbar.add(author);
authorbar.add(new JLabel("          "));
//Set Creator
JToolBar creatorbar=new JToolBar();
creatorbar.setFloatable(false);
creatorbar.setBorderPainted(false);
creator=new JTextField();
JLabel creatorlabel=new JLabel("Enter Creator:");
creatorbar.add(creatorlabel);
creatorbar.add(creator);
creatorbar.add(new JLabel("          "));
//Set Keywords
JToolBar keybar=new JToolBar();
keybar.setFloatable(false);
keybar.setBorderPainted(false);
key=new JTextField();
JLabel keylabel=new JLabel("Enter Keyword:");
keybar.add(keylabel);
keybar.add(key);
keybar.add(new JLabel("          "));
//Lay everything to summary
summary.add(titlebar);
summary.add(subjectbar);
summary.add(authorbar);
summary.add(creatorbar);
summary.add(keybar);
summary.setBorder(BorderFactory.createLineBorder(Color.blue));
tabbed.addTab("Summary",summary);
/*
South Panel
*/
mainsouth=new JPanel();
okbutton=new JButton("OK");
//okbutton listener
okbutton.addActionListener(new ActionListener()
{

```

```

public void actionPerformed(ActionEvent e)
{ //Check the file we choose for output
  if (outputfile!=null && outputfilestring.endsWith(".pdf"))
  { //Gather information from the tree
    expandTree();
    int size=0;
    while(title[size]!=null)size++;
    String titles[]=new String[size];
    for(int y=0;y<size;y++)
    {
      titles[y]=title[y];
    }
    //Construct a Concat class
    Concat con=new Concat(arg,page,titles,untitled,level);
    //Send parametres and information to Concat class
    con.setHeader(header.getText(),pagecountbox2.isSelected());
    con.setFooter(footer.getText(),pagecountbox.isSelected());

con.setSummary(titlepdf.getText(),subject.getText(),author.getText(),creator.getText(),key.g
etText());
    con.setAllowPrinting(printBox.isSelected());
    con.setAllowCopying(copyBox.isSelected());
    con.setAllowAssembly(assemblybox.isSelected());
    con.setAllowFillIn(fillinbox.isSelected());
    con.setAllowModifyAnnotations(modifyannotbox.isSelected());
    con.setAllowModifyContents(modifycontbox.isSelected());
    con.setAllowScreenReaders(screenreadersbox.isSelected());
    con.setHideToolbar(hideToolbarBox.isSelected());
    con.setHideMenuebar(hideMenuebarBox.isSelected());
    con.setPageMode(group.getSelection().getActionCommand());
    con.setPageLayout(group2.getSelection().getActionCommand());
    con.setCreateContents(contentsbox.isSelected());
    if (list.getSelectedIndex()==0)con.setLanguage(true);
    else if (list.getSelectedIndex()==1)con.setLanguage(false);
    if (encryptBox.isSelected())
    {
      con.setOwnerPassword(ownerpass,true);
      con.setUserPassword(userpass,true);
    }
    else
    {
      con.setOwnerPassword(null,false);
      con.setUserPassword(null,false);
    }
    con.setCover(cover);
    con.Create();

```

```

    frame.dispose();
}
else
{
JOptionPane.showMessageDialog(frame,
"You have to enter a valid pdf output first", "Invalid File",
JOptionPane.ERROR_MESSAGE);return;
}
}
});
cancelbutton=new JButton("Cancel");
cancelbutton.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
frame.dispose();
}
});
mainsouth.add(okbutton);
mainsouth.add(cancelbutton);
contentPane.add("South",mainsouth);
/*
Lay everything to Frame
*/
contentPane.add("Center",tabbed);
frame.setSize(600,380);
frame.setVisible(true);
frame.setLocation(200,200);
}

//Collect all the information from the tree
public void expandTree()
{
boolean next=true;
int x=1;
int y=0;
allpage=1;
int fatherpage=2;
DefaultMutableTreeNode node;
Object nodeInfo;
node=rootNode;
boolean untitledmessage=false;
while(next)
{ //Check all nodes
node= node.getNextNode();
if (node==null)next=false;

```

```

else
{
    nodeInfo = node.getUserObject();
    Info per=(Info)nodeInfo;
    //Set untitled
    untitled[y]=per.untitled;
    //Set level
    level[y]=node.getLevel();
    //if we find untitled nodes
    if(untitled[y])untitlemessage=true;

    if (per.getPdfSource()!=null )
    {
        //Take information from a chapter
        //Set page for the chapter in the new pdf
        page[y]=allpage;
        fatherpage=allpage;
        allpage=allpage+per.getPages();
        title[y]=per.getName();
        y++;
        arg[x]=per.getPdfSource();
        x++;
    }
    else
    {
        //Take information from a subchapter
        //Set page for the subchapter in the new pdf
        page[y]=fatherpage+per.getPages()-1;
        title[y]=per.getName();
        y++;
        arg[x]=per.getPdfSource();
        x++;
    }
}
}
//if we have untitled nodes in the tree display a message
if (untitlemessage)JOptionPane.showMessageDialog(frame,
"You have untitled chapters or subchapters.\nThey program wont include them in the
contents ",
"Untitled Chapters or subchapters",
JOptionPane.WARNING_MESSAGE);
}
}

```

5.1.8 Concat.java

Είναι η κλάση που δημιουργεί το νέο αρχείο συνενώνοντας τα PDF που βρίσκονται στο δέντρο παίρνοντας δεδομένα απο τις υπόλοιπες κλάσεις .Στην κλάση αυτή δημιουργούνται και τα περιεχόμενα του κειμένου

```
package pdfmanager;

/**
 * Title: PdfManager
 * Description: A program that manage Pdf documents
 * Copyright: Copyright (c) 2003
 * Company: Ece Ntua
 * @Vlotomas Neofytos e197605
 * @version 1.0
 */
import java.io.*;
import java.awt.Color;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import java.awt.Point;
import java.awt.FontMetrics;
import java.awt.Graphics;
import com.lowagie.text.*;
import com.lowagie.text.pdf.*;

public class Concat extends Object {

private JFrame frame;
/*
Set metadata for the output file
*/
//set encryption
private boolean allowCopying = false;
private boolean allowPrinting = false;
private String userPassword = null;
private String ownerPassword = null;
private boolean allowmodifyContents=false;
private boolean allowmodifyAnnotations=false;
private boolean allowFillin=false;
private boolean allowscreenReaders=false;
private boolean allowAssembly=false;
private boolean encrypted=false;
//Set view preferences
```



```

private boolean hideToolbar = false;
private boolean hideMenuebar = false;
//set summary
private String titlepdf,subject, author, creator,key;
//set edit prefernces
private          String          curentdir=System.getProperty("user.dir")+
System.getProperty("file.separator");
private HeaderFooter header=null, footer=null;
//Set the language for the contents true=greek,false=english
private boolean greek=true;
private boolean createcontens=true;
private String cover;
/*
global Declarations
*/
private PdfPTable datatable,datatabletest;
private PdfReader reader;
private PdfContentByte cb;
private PdfImportedPage pagepdf;
private int contentsize=1;
private String args[],title[],actionmode,actionlayout;
private int page[];
private boolean untitled[];
private int level[],rotation;

public Concat (String args[],int page[],String title[],boolean untitled[],int level[])
{
    this.args=args;
    this.title=title;
    this.page=page;
    this.untitled=untitled;
    this.level=level;
}

//Create the Pdf Document
public void Create()
{
    try
    {
        //create new True Type fonts
        BaseFont          Times          =
BaseFont.createFont(curentdir+"conf"+System.getProperty("file.separator")+"times.ttf",
BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
        BaseFont          TimesBold          =
BaseFont.createFont(curentdir+"conf"+System.getProperty("file.separator")+"timesbd.ttf",
BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
    }
}

```

```

Font font = new Font(Times, 12);
Font fontbold=new Font(TimesBold,14);
int f = 1;
String arghelp[]=new String[200];
for(int g=1;g<args.length;g++)arghelp[g]=args[g];
//Find the not null source
while(args[f]==null)f++;
// creation of a document-object
Document document = new Document(PageSize.A4,30,40,30,50);

//We estimate the size off the contents in a new document
//We dont write the contents
if (createcontents)
{
    Document documentcont = new Document(PageSize.A4,30,40,30,50);
    // we create a writer that listens to the document
    OutputStream p1=new FileOutputStream(args[0]);
    PdfWriter writercont = PdfWriter.getInstance(documentcont, p1);
    header.setBorder(header.NO_BORDER);
    footer.setBorder(footer.NO_BORDER);
    documentcont.setHeader(header);
    documentcont.setFooter(footer);
    documentcont.open();
    documentcont.setMargins(30,40,30,50);
    int x=0;
    datatable = new PdfPTable(2);
    Cell cell=new Cell();
    datatable.getDefaultCell().setPadding(3);
    int headerwidths[] = {11, 1}; // percentage 11 to 1
    datatable.setWidths(headerwidths);
    datatable.setWidthPercentage(100); // percentage
    datatable.getDefaultCell().setHorizontalAlignment(Element.ALIGN_LEFT);
    datatable.getDefaultCell().setVerticalAlignment(Element.ALIGN_BOTTOM);
    String titlecont="Test\n\n";
    datatable.addCell(new Phrase (new Chunk(titlecont,fontbold)));
    datatable.addCell("");
    while(x<title.length)
    {
        if(!untitled[x])
        {
            //Link to pagenumber
            int link=1;
            String titleprint="";
            for(int u=1;u<=level[x];u++)
            {
                titleprint=titleprint+" ";
            }
        }
    }
}

```

```

    }
    titleprint=titleprint+title[x];
    Chunk chunk= new Chunk(titleprint,font);
    datatable.addCell(new Phrase(chunk));
    int printlink=link;
    chunk=new Chunk(""+printlink,font);
    datatable.addCell(new Phrase(chunk));
    }
    x++;
}

documentcont.add(datatable);
documentcont.close();
reader = new PdfReader(args[0]);
contentsize=reader.getNumberOfPages();
}
// we create a writer that listens to the document
PdfWriter writer = PdfWriter.getInstance(document, new FileOutputStream(args[0]));
//add some metadata about the document
int allowances = 0;
if (allowCopying) { allowances += PdfWriter.AllowCopy; }
if (allowPrinting) { allowances += PdfWriter.AllowPrinting; }
if( allowmodifyContents) { allowances += PdfWriter.AllowModifyContents; }
if ( allowmodifyAnnotations) { allowances += PdfWriter.AllowModifyAnnotations; }
if ( allowFillin) { allowances += PdfWriter.AllowFillIn; }
if (allowscreenReaders) { allowances += PdfWriter.AllowScreenReaders; }
if (allowAssembly) { allowances += PdfWriter.AllowAssembly; }
int viewerPreferences = 0;
if (hideMenuebar) { viewerPreferences += PdfWriter.HideMenubar; }
if (hideToolbar) { viewerPreferences += PdfWriter.HideToolbar; }
writer.setViewerPreferences(viewerPreferences);
//Set the Encryption
if(encrypted)
writer.setEncryption(false, this.userPassword, this.ownerPassword, allowances);
document.addKeywords( key );
document.addSubject(subject );
document.addTitle( titlepdf);
document.addAuthor( author );
document.addCreator(creator);
//Set the Page mode
if (actionmode.equals("none") )writer.setViewerPreferences(writer.PageModeUseNone);
else if(actionmode.equals("outline"))
writer.setViewerPreferences(writer.PageModeUseOutlines);
else if (actionmode.equals("thumb"))
writer.setViewerPreferences(writer.PageModeUseThumbs);

```

```

else if (actionmode.equals("fullscreen"))
writer.setViewerPreferences(writer.PageModeFullScreen);
//Set the Page Layout
if
(actionlayout.equals("single"))writer.setViewerPreferences(writer.PageLayoutSinglePage);
else
if(actionlayout.equals("one"))writer.setViewerPreferences(writer.PageLayoutOneColumn);
else
if(actionlayout.equals("twoleft"))writer.setViewerPreferences(writer.PageLayoutTwoColumn
Left);
else
if(actionlayout.equals("tworight"))writer.setViewerPreferences(writer.PageLayoutTwoColu
mnRight);
document.setMargins(30,40,30,50);
if(cover.equals(""))
{
//Set the header-footer
header.setAlignment(header.ALIGN_CENTER);
document.setHeader(header);
footer.setAlignment(header.ALIGN_CENTER);
document.setFooter(footer);
}
//Open the document
document.open();
//Set the cover of the document
cb = writer.getDirectContent();
int i=0;
if(!cover.equals(""))
{
PdfReader readercover = new PdfReader(cover);
// we retrieve the total number of pages of the cover
int coverpage = readercover.getNumberOfPages();
//current page
while (i < coverpage)
{
i++;
//Start writing to the page
document.setPageSize(readercover.getPageSizeWithRotation(i));
//Change Page
document.newPage();
pagepdf = writer.getImportedPage(readercover, i);
rotation = readercover.getPageRotation(i);
if (rotation == 90 || rotation == 270)
{
cb.addTemplate(pagepdf, 0, -1f, 1f, 0, 0,
readercover.getPageSizeWithRotation(i).height());

```

```

    }
    else
    {
        cb.addTemplate(pagepdf, 1f, 0, 0, 1f, 0, 0);
    }

}
document.resetPageCount();
}
if(!cover.equals(""))
{
    //Set the header-footer
    header.setAlignment(header.ALIGN_CENTER);
    document.setHeader(header);
    footer.setAlignment(header.ALIGN_CENTER);
    document.setFooter(footer);
    document.newPage();
}
//We write the contents and we know the size of them
reader = new PdfReader(args[f]);
if (createcontents)
{
    int x=0;
    datatable = new PdfPTable(2);
    Cell cell=new Cell();
    datatable.getDefaultCell().setPadding(3);
    int headerwidths[] = {11, 1}; // percentage 11 to 1
    datatable.setWidths(headerwidths);
    datatable.setWidthPercentage(100); // percentage
    datatable.getDefaultCell().setHorizontalAlignment(Element.ALIGN_LEFT);
    datatable.getDefaultCell().setVerticalAlignment(Element.ALIGN_BOTTOM);
    datatable.getDefaultCell().setBorderColor(Color.white);
    String titlecont="";
    //Set the title of the contents
    if(greek)
    {
titlecont="\u03A0\u03B5\u03C1\u03B9\u03B5\u03C7\u03CC\u03BC\u03B5\u03BD\u03B1";
    },
        PdfAction action=new PdfAction(args[0],i+1);
        cb = writer.getDirectContent();
        PdfOutline content = new PdfOutline(cb.getRootOutline(),action,titlecont);
        titlecont=titlecont+"\n\n";
    }
    else
    {

```

```

titlecont="CONTENTS";
PdfAction action=new PdfAction(args[0],i+1);
cb = writer.getDirectContent();
PdfOutline content = new PdfOutline(cb.getRootOutline(),action,titlecont);
titlecont=titlecont+"\n\n";
}
datatable.addCell(new Phrase (new Chunk(titlecont,fontbold)));
datatable.addCell("");
contentsize=contentsize+i;
//Can make up to 8 levels Bookmark
PdfOutline out[]=new PdfOutline[8];
while(x<title.length)
{
if(!untitled[x])
{
//Link to pagenumber
int link=page[x]+contentsize;
String titleprint="";
for(int u=1;u<=level[x];u++)
{
titleprint=titleprint+" ";
}
titleprint=titleprint+title[x];
Chunk chunk= new Chunk(titleprint,font).setRemoteGoto(args[0],link);
datatable.addCell(new Phrase(chunk));
int printlink=link-i;
chunk=new Chunk(""+printlink,font).setRemoteGoto(args[0],link);
datatable.addCell(new Phrase(chunk));
//Set the Bookmark of acrobat
PdfAction action=new PdfAction(args[0],link);
cb = writer.getDirectContent();
switch(level[x])
{
case 1: out[1] = new PdfOutline(cb.getRootOutline(),action,new
Paragraph(title[x],font));
break;
case 2: out[2] = new PdfOutline(out[1],action,new Paragraph(title[x],font));
break;
case 3: out[3] = new PdfOutline(out[2],action,new Paragraph(title[x],font));
break;
case 4: out[4] = new PdfOutline(out[3],action,new Paragraph(title[x],font));
break;
case 5: out[5] = new PdfOutline(out[4],action,new Paragraph(title[x],font));
break;
case 6: out[6] = new PdfOutline(out[5],action,new Paragraph(title[x],font));
break;
}
}
}

```

```

    case 7: out[7] = new PdfOutline(out[6],action,new Paragraph(title[x],font));
    break;
    case 8: out[8] = new PdfOutline(out[7],action,new Paragraph(title[x],font));
    break;
    }
}
x++;
}
document.setMargins(30,40,50,50);
//Add the content to the document
document.add(datatable);
}
// we retrieve the total number of pages of the first document
int n=reader.getNumberOfPages();
// we add content to the document
while(f < args.length)
{
    //current page
    i=0;
    if (args[f]!=null)
    while (i < n)
    {
        //n= Page number of current document
        i++;
        //Start writing to the page
        document.setPageSize(reader.getPageSizeWithRotation(i));
        //Change Page
        document.newPage();
        pagepdf = writer.getImportedPage(reader, i);
        rotation = reader.getPageRotation(i);
        if (rotation == 90 || rotation == 270)
        {
            cb.addTemplate(pagepdf, 0, -1f, 1f, 0, 0, reader.getPageSizeWithRotation(i).height());
        }
        else {
            cb.addTemplate(pagepdf, 1f, 0, 0, 1f, 0, 0);
        }
    }
    //Go to next document
    f++;
    if (f < args.length && args[f]!=null)
    {
        reader = new PdfReader(args[f]);
        // we retrieve the total number of pages
        n = reader.getNumberOfPages();
    }
}

```

```

    }
    // close the document
    document.close();
    JOptionPane.showMessageDialog(frame,
    "The document created succesfully", "Create ",
    JOptionPane.INFORMATION_MESSAGE);
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(frame,
    e.getClass().getName() + ": " + e.getMessage(), "Exception ex",
    JOptionPane.ERROR_MESSAGE);
}
}
//Set the Parametres for Metadata from CreatePdfFrame Class
public void setAllowPrinting(boolean allow)
{
    this.allowPrinting = allow;
    if (allow) encrypted=true;
}

public void setAllowCopying(boolean allow)
{
    this.allowCopying = allow;
    if (allow) encrypted=true;
}

public void setAllowModifyContents(boolean allow)
{
    this.allowmodifyContents = allow;
    if (allow) encrypted=true;
}

public void setAllowModifyAnnotations(boolean allow)
{
    this.allowmodifyAnnotations = allow;
    if (allow) encrypted=true;
}

public void setAllowFillIn(boolean allow)
{
    this.allowFillin = allow;
    if (allow) encrypted=true;
}

public void setAllowScreenReaders (boolean allow)

```



```

{
    this.allowScreenReaders = allow;
    if (allow) encrypted=true;
}

public void setAllowAssembly (boolean allow)
{
    this.allowAssembly = allow;
    if (allow) encrypted=true;
}

public void setHideToolbar(boolean hide)
{
    this.hideToolbar = hide;
}

public void setHideMenuebar(boolean hide)
{
    this.hideMenuebar = hide;
}

public void setUserPassword(String password ,boolean allow)
{
    this.userPassword = password;
    if (allow) encrypted=true;
}

/*
Sets the ownerpassword for the document.
The password can be null, in this case a random String
is generated if the document is encrypted
*/

public void setOwnerPassword(String password,boolean allow)
{
    this.ownerPassword = password;
    if (allow) encrypted=true;
}

public void setHeader( String header,boolean count)
{
    try
    {

```

```

        BaseFont                Times
BaseFont.createFont(curentdir+"conf"+System.getProperty("file.separator")+"times.ttf",
BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
    Font font = new Font(Times, 8);
    if(count)
        header=header+"\n";
    else header=header+"\n\n";
    this.header=new HeaderFooter(new Phrase(header,font ), count);

    }
catch(Exception ex)
    {
        System.err.println(ex.getMessage());
    }
}
public void setFooter( String footer,boolean count)
    {
        try
        {
            BaseFont                Times
BaseFont.createFont(curentdir+"conf"+System.getProperty("file.separator")+"times.ttf",
BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
            Font font = new Font(Times, 8);
            if(count)
                footer=footer+"\n";
            else footer=footer+"\n\n";
            this.footer=new HeaderFooter(new Phrase(footer,font ), count);
        }
        catch(Exception ex)
        {
            System.err.println(ex.getMessage());
        }
    }

public void setSummary(String titlepdf,String subject,String author,String creator,String
key)
    {
        this.titlepdf=titlepdf;
        this.subject=subject;
        this.author=author;
        this.creator=creator;
        this.key=key;
    }

public void setPageMode(String actionmode)
    {

```

```

    this.actionmode=actionmode;
}
public void setPageLayout(String actionlayout)
{
    this.actionlayout=actionlayout;
}
public void setLanguage(boolean greek)
{

    this.greek=greek;
}

public void setCreateContents(boolean create)
{
    this.createcontents=create;

}
public void setCover(String cover)
{
    this.cover=cover;

}
}
//End of class
}

```

5.1.9 Acceptedfiles.java

Χρησιμοποιείται για την επιλογή files με μια συγκεκριμένη επέκταση .

```

package pdfmanager;
/**
 * Title: PdfManager
 * Description: A program tha manage Pdf documents
 * Copyright: Copyright (c) 2003
 * Company: Ece Ntua
 * @Vlotomas Neofytos e197605
 * @version 1.0
 */

import java.io.File;
import javax.swing.*.*;
import javax.swing.filechooser.*;

public class Acceptedfiles extends FileFilter {

    public String extension;

```

```

public Acceptedfiles(String extension)
{
    this.extension=extension;
}
//Accepts only file with the extension
public boolean accept(File f)
{
    if (f.isDirectory())
    {
        return true;
    }
    if (extension != null)
    {
        if (f.getName().endsWith(extension)) return true;
        else
        {
            return false;
        }
    }
}

return false;
}

//The description of the filter
public String getDescription()
{
    return "Select "+extension+" file";
}
}

```

5.10 SplitPdf.java

Είναι ένα frame το οποίο παίρνει σαν είσοδο ένα αρχείο PDF και το διασπά σε όλες τις σελίδες του .Αυτό έγινε για να μπορούμε να διαχειριζόμαστε μεγάλα αρχεία PDF πιο εύκολα .Δεν συμμετέχει άμεσα στο τελικό αποτέλεσμα αλλά βοηθά στην καλύτερη διαχείριση των αρχείων PDF που έχουμε .

```

package pdfmanager;

/**
 * Title: PdfManager
 * Description: A program tha manage Pdf documents
 * Copyright: Copyright (c) 2003
 * Company: Ece Ntua

```

```

* @Vlotos Neofytos el97605
* @version 1.0
*/

import java.io.*;
import com.lowagie.text.Document;
import com.lowagie.text.pdf.*;
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class SplitPdf extends JFrame
{
    /*
    Swing declarations
    */
    private JButton find,split;
    private JPanel north,center,south;
    private JTextField comment,title;
    private JLabel filelabel,findsplit;
    private JFileChooser fc;
    private JFrame frame;
    /*
    global Declarations
    */
    private Document document;
    private String args[]=new String[4];
    private String userdir=System.getProperty("user.dir")+
System.getProperty("file.separator");
    private File file,sourcedir;
    private ImageIcon open = new ImageIcon("images/open.gif");

    public SplitPdf ()
    {
        frame = new JFrame("Splid a pdf document");
        final Container contentPane = frame.getContentPane();
        contentPane.setLayout(new BorderLayout(10,10));
        north=new JPanel(new GridLayout(2,0));
        center=new JPanel(new GridLayout(1,0));
        south=new JPanel(new BorderLayout());
        find=new JButton("",open);
        split=new JButton("Split");
        /*
        North panel
        */

```

```

title=new JTextField("      Split a Pdf Document page by page\n");
title.setBorder(BorderFactory.createEmptyBorder());
title.setFont(new Font("Arial",Font.BOLD,12));
title.setEditable(false);
title.setBackground(contentPane.getBackground());
title.setForeground(Color.blue);
comment=new JTextField(" In order to manage large Pdf documents you can split a
srcfile page \n"+
" by page and put the produced pdf in a selected directory .\n"+
" Then you can build smaller documents with the create command in \n"+
" the menu Bar with or without contents");
comment.setBorder(BorderFactory.createEmptyBorder());
comment.setForeground(Color.blue);
comment.setEditable(false);
comment.setBackground(contentPane.getBackground());
north.add(title);
north.add(comment);
contentPane.add("North",north);
/*
Center panel
*/
JToolBar findbar=new JToolBar();
findbar.setFloatable(false);
findbar.setBorder(BorderFactory.createEmptyBorder());
findbar.setBackground(contentPane.getBackground());
findbar.setForeground(Color.blue);
filelabel=new JLabel();
filelabel.setBackground(contentPane.getBackground());
filelabel.setForeground(Color.blue);
filelabel.setBorder(BorderFactory.createEmptyBorder());
findsplit=new JLabel("Find a Pdf to Split ");
findsplit.setForeground(Color.blue);
findsplit.setBorder(BorderFactory.createEmptyBorder());
findbar.add(findsplit);
findbar.add(find);
findbar.add(filelabel);
center.add(findbar);
contentPane.add("Center",center);
fc=new JFileChooser("c:/");
/*
South Panel
*/
JToolBar splitbutton=new JToolBar();
splitbutton.setBorder(BorderFactory.createLineBorder(Color.blue));
splitbutton.setFloatable(false);
splitbutton.add(new JLabel("

```

```

splitbutton.add(split);
south.add("Center",splitbutton);
contentPane.add("South",south);
//Split button Listener
split.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
try
{
// we create a reader for the selected document
PdfReader reader = new PdfReader(file.getPath());
// we retrieve the total number of pages
int n = reader.getNumberOfPages();
if(n==1)
{
JOptionPane.showMessageDialog(frame,"There is only one page in the file",
"Split Error", JOptionPane.ERROR_MESSAGE);
frame.dispose();
return;
}
//We chose a directory to put the produced files
fc.addChoosableFileFilter(new Acceptedfiles("Directory"));
fc.setAcceptAllFileFilterUsed(false);
int returnVal = fc.showOpenDialog(SplitPdf.this);
if (returnVal == JFileChooser.APPROVE_OPTION)
{
sourcedir = fc.getSelectedFile();
}
else
{
return;
}

document = new Document(reader.getPageSizeWithRotation(1));
PdfWriter writer = PdfWriter.getInstance(document, new
FileOutputStream(sourcedir.getParentFile().getPath()+System.getProperty("file.separator")+
"split1.pdf"));
System.out.println(sourcedir.getParentFile().getPath());
document.open();
PdfContentByte cb1 = writer.getDirectContent();
PdfImportedPage page;
int rotation;
//we add content for the first page
document.setPageSize(reader.getPageSizeWithRotation(1));
document.newPage();

```

```

page = writer.getImportedPage(reader, 1);
rotation = reader.getPageRotation(1);
if (rotation == 90 || rotation == 270)
{
    cb1.addTemplate(page, 0, -1f, 1f, 0, 0, reader.getPageSizeWithRotation(1).height());
}
else
{
    cb1.addTemplate(page, 1f, 0, 0, 1f, 0, 0);
}
document.close();
int i = 1;
//we add content to the other pages
while(i!=n)
{
    int pagepdf=i+1;
    // creation of a document-object
    document = new Document(reader.getPageSizeWithRotation(i));

    // we create a writer that listens to the document
    writer = PdfWriter.getInstance(document, new
FileOutputStream(sourcedir.getParentFile().getPath()+System.getProperty("file.separator")
+"split"+pagepdf+".pdf"));

    // we open the document
    document.open();
    cb1 = writer.getDirectContent();
    // we add content
    i++;
    document.setPageSize(reader.getPageSizeWithRotation(i));
    document.newPage();
    page = writer.getImportedPage(reader, i);
    rotation = reader.getPageRotation(i);
    if (rotation == 90 || rotation == 270)
    {
        cb1.addTemplate(page, 0, -1f, 1f, 0, 0, reader.getPageSizeWithRotation(i).height());
    }
    else
    {
        cb1.addTemplate(page, 1f, 0, 0, 1f, 0, 0);
    }
    document.close();

    frame.dispose();
}

```



```

        JOptionPane.showMessageDialog(frame,"The Split documents are in
"+sourcedir.getParentFile().getPath()+" directory",
        "Split Successful", JOptionPane.INFORMATION_MESSAGE);
    }
    catch(Exception ex)
    {
        JOptionPane.showMessageDialog(frame,ex.getClass().getName() + ": " +
ex.getMessage(),
        "Exception", JOptionPane.WARNING_MESSAGE);
    }
    });
//find the PDF to split
find.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        fc.addChoosableFileFilter(new Acceptedfiles("pdf"));
        fc.setAcceptAllFileFilterUsed(false);
        int returnVal = fc.showOpenDialog(SplitPdf.this);
        if (returnVal == JFileChooser.APPROVE_OPTION)
        {
            file = fc.getSelectedFile();
            filelabel.setText( file.getPath() );
        }
        else
        {
            filelabel.setText("Open command cancelled by user.");
        }
    }
});
frame.setSize(400,260);
frame.setVisible(true);
frame.setLocation(200,200);
try
{
    UIManager.setLookAndFeel( UIManager.getSystemLookAndFeelClassName() );
}
catch( Exception e )
{
    e.printStackTrace();
}
}
}

```

5.2 Πακέτο UpdatePdfM

5.2.1 ConvertDBtoXML.java

Είναι ένα Servlet που χρησιμοποιείται για την ανταλλαγή πληροφοριών μεταξύ του χρήστη (client) και της βάσης δεδομένων του Ε.Μ.Π. μέσω XML αρχείων .

```
package updatepdfm;

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
import java.sql.*;

public class ConvertDBtoXML extends HttpServlet {
    /*
    Define global variables
    */
    private String filename;
    private Connection con=null;
    private String query;

    //Process the HTTP Post request
    public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
    {
        doPost(request,response);
    }
    //Process the HTTP Get request
    public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
    {
        boolean paramcheck=true;
        /*
        Take parameters from client
        */
        //Get Database Driver
        String dbDriver = request.getParameter("driver");
        if (dbDriver.equals(null) || dbDriver.equals(""))
```

```

{
    paramcheck = false;
}
//Get Database Url
String dbUrl = request.getParameter("dburl");
if (dbUrl.equals(null) || dbUrl.equals(""))
{
    paramcheck = false;
}
//Get username
String user = request.getParameter("user");
if (user.equals(null) || user.equals(""))
{
    user = "";
}
//Get password
String pass = request.getParameter("pass");
if (pass.equals(null) || pass.equals(""))
{
    pass = "";
}
//Get the file you want to download
String file = request.getParameter("file");
if (file == null)
{
    paramcheck = false;
}
//if the parameters are correct
if(paramcheck)
{
    try
    {
        Class.forName(dbDriver);
        //Connect to database
        con = DriverManager.getConnection(dbUrl,user,pass);
        //Set Metadata for the database
        DatabaseMetaData dmd = con.getMetaData();
        //get results from database
        ResultSet rs=dmd.getTables(null,null,"%",null);
        Vector Tables = new Vector();
        while (rs.next())
        {
            Tables.addElement(rs.getString("TABLE_NAME"));
        }
        if (Tables.isEmpty()) throw new SQLException("No tables found in database");
    }
    /*

```



```

        out.flush();
        out.println(" </"+returnRank(g)+">");
    }
    out.println(" </Division"+y+">");
}
out.println("</database_data>");
out.println("</database>");
out.flush();
}
/*
Download Chap7-9.xml
*/
else if(file.equals("chap7-9"))
{
    filename="Chap7-9.xml";
    response.setContentType("text/plain");
    response.setStatus(206);
    response.setHeader("Content-Disposition","attachment; filename=\""+
filename+"\"");
    PrintWriter out = response.getWriter();
    //Set the output xml file
    out.println("<?xml version=\"1.0\" encoding=\"ISO-8859-7\" ?>");
    out.println("<!-- " + filename + " -->");
    out.println("<database>");
    Iterator ITables = Tables.iterator();
    Statement stmt = con.createStatement ();
    String data;
    ResultSetMetaData rsmd;
    out.println("<database_data>");
    ITables = Tables.iterator();
    //Divide by semester
    for(int y=1;y<=5;y++)
    {
        //Set the query for the database
        query="Select
Courses.CourseID,Code,Courses.TitleGR,LectureHours,LabHours,CourseCategories.TitleG
R,Courses.DescriptionGR "+
        "from Courses,CourseCategories where "+
        "CourseCategories.CourseCategoryID=Courses.Category AND Semester="+y;
        //execute query
        rs = stmt.executeQuery(query);
        rsmd = rs.getMetaData();
        out.println(" <semester_"+y+">");
        while (rs.next())
        {
            out.println(" <row>");

```

```

for (int i = 1; i<=rsmd.getColumnCount(); i++)
{
    data = "\t<" + rsmd.getColumnName(i) + " _"+i+">";
    data = data + "<![CDATA[" + rs.getString(i)+"]>";
    data = data + "</"+ rsmd.getColumnName(i)+ " _"+i+">";
    out.println(data);
}
out.println(" </row>");
}

out.flush();
out.println(" </semester_" +y+">");
}

//Set the query for the database
query="select CourseID,NameGR,SurnameGR from"+
" Staff,CourseLecturers where CourseLecturers.StaffID=Staff.Staffid";
//Execute query
rs = stmt.executeQuery(query);
rsmd = rs.getMetaData();
//Divide by professor
out.println(" <Professors>");
String code="";
boolean first=true;
while (rs.next())
{
for (int i = 1; i<=rsmd.getColumnCount(); i++)
{
if(i==1)
{
if (!code.equals(rs.getString(1)))
{
if(!first)
{
out.println(" </row>");
}
code=rs.getString(1);
out.println(" <row ");
out.println(rsmd.getColumnName(i) + "=");
out.println(""" +rs.getString(i)+"">");
first=false;
}
}
}
else
{
if(i==2) out.println("<professor>");
}
}
}
}

```

```

        out.println("\t<" + rsmd.getColumnName(i) + ">");
        out.println("<![CDATA[" + rs.getString(i)+"]]>");
        out.println("</" + rsmd.getColumnName(i) + ">");
        if(i==3) out.println("</professor>");
    }
}

}
out.println(" </row>");
out.println(" </Professors>");
out.println("</database_data>");
out.println("</database>");
out.flush();
}
/*
Download chap8-10.xml
*/
else if(file.equals("chap8-10"))
{
    filename="Chap8-10.xml";
    response.setContentType("text/plain");
    response.setStatus(206);
    response.setHeader("Content-Disposition","attachment; filename=\"" +
    filename+"\"");
    //Set the output xml file
    PrintWriter out = response.getWriter();
    out.println("<?xml version=\"1.0\" encoding=\"ISO-8859-7\" ?>");
    out.println("<!-- " + filename + " -->");
    out.println("<database>");
    Iterator ITables = Tables.iterator();
    Statement stmt = con.createStatement ();
    String data;
    ResultSetMetaData rsmd;
    out.println("<database_data>");
    //Divide by flow
    ITables = Tables.iterator();
    for(int y=1;y<=14;y++)
    {
        query="Select
Courses.CourseID,Code,Courses.TitleGR,Semester,LectureHours,LabHours,CourseCategori
es.TitleGR,Courses.DescriptionGR "+
        "from Courses,CourseCategories where "+
        "CourseCategories.CourseCategoryID=Courses.Category AND Flow="+y;
        rs = stmt.executeQuery(query);
        rsmd = rs.getMetaData();
        out.println(" <flow_" +y+">");

```

```

while (rs.next())
{
out.println(" <row>");
for (int i = 1; i<=rsmd.getColumnCount(); i++)
{
data = "\t<" + rsmd.getColumnname(i) + " _"+i+">";
data = data + "<![CDATA[" + rs.getString(i)+"]]>";
data = data + "</"+ rsmd.getColumnname(i)+ " _"+i+">";
out.println(data);
}
out.println(" </row>");
}
out.flush();
out.println(" </flow_ "+y+">");
}
query="select CourseID,NameGR,SurnameGR from"+
" Staff,CourseLecturers where CourseLecturers.StaffID=Staff.Staffid";
rs = stmt.executeQuery(query);
rsmd = rs.getMetaData();
out.println(" <Professors>");
String code="";
//Divide by professor
boolean first=true;
while (rs.next())
{
for (int i = 1; i<=rsmd.getColumnCount(); i++)
{
if(i==1)
{
if (!code.equals(rs.getString(1)))
{
if(!first)
{
out.println(" </row>");
}
code=rs.getString(1);
out.println(" <row ");
out.println(rsmd.getColumnname(i) + "=");
out.println(""" +rs.getString(i)+"">");
first=false;
}
}
}
else
{
if(i==2) out.println("<professor>");
out.println("\t<" + rsmd.getColumnname(i) + ">");
}
}
}
}

```



```

        out.println("<![CDATA[" + rs.getString(i) + "]]>");
        out.println("</" + rsmd.getColumnName(i) + ">");
        if(i==3) out.println("</professor>");
    }
}
}
out.println(" </row>");
out.println(" </Professors>");
out.println("</database_data>");
out.println("</database>");
out.flush();
}
//close connection
con.close();
} catch (SQLException e)
{
e.printStackTrace(); ShowError(response,e.toString());
}
catch (java.lang.Exception e)
{
e.printStackTrace(); ShowError(response,e.toString());
}
}
else
ShowError(response,"Parameter Error...");
} //end doGet
//Show Error
private void ShowError(HttpServletResponse response,String msg)
{
String value1="<html><head><meta http-equiv=\"Content-Type\" content=\"text/html; charset=windows-1253\">";
value1=value1+"<title>ERROR</title></head><body>";
value1=value1+"<h1><font color=\"red\">Error</font></h1><br><hr>";
value1=value1+"<h2>"+msg+"</h2>";
value1=value1+"Please click the link below to resume the procedure";
//return back
value1=value1+"<hr><a href=\"/index.htm\">Click Here to Resume</a>";
value1=value1+"<hr></body></html>";
response.setContentType("text/html; charset=windows-1253");
try
{
PrintWriter out = response.getWriter();
out.println(value1);
out.flush();
out.close();
}
}

```

```
catch (java.io.IOException e)
{
e.printStackTrace();}
} //end ShowError

//get Rank of professor
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;
}

//Clean up resources
public void destroy() {

}
}
```

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Το ηλεκτρονικό εμπόριο είναι μια επανάσταση στο χώρο της αγοράς, που θα επιφέρει τεράστιες αλλαγές τόσο στον τρόπο συναλλαγής όσο και στον τρόπο διακίνησης των προϊόντων, καθώς ουσιαστικά καταργεί τα γεωγραφικά σύνορα. Σίγουρο είναι πως για αρκετά χρόνια ακόμα το βιβλίο, ένα από τα πρώτα είδη που διακινήθηκαν ηλεκτρονικά μέσω Διαδικτύου, θα διατηρήσει για πολύ ακόμη τη μορφή με την οποία το γνωρίζουμε εδώ και πέντε αιώνες, αλλά ταυτόχρονα εξίσου βέβαιο είναι πως τα ψηφιακά έγγραφα θα πρωταγωνιστήσουν στο μέλλον.

Το Adobe Portable Document Format παρέχει ένα αξιόπιστο και αποτελεσματικό τρόπο για την ανταλλαγή ψηφιακών εγγράφων. Με την χρήση PDF αρχείων υπάρχει η δυνατότητα εμπλουτισμού τους με επιπρόσθετες λειτουργίες αυξάνοντας το εύρος της χρησιμότητας τους. Το PDF αποτελεί σήμερα μια εντυπωσιακή τεχνολογική εξέλιξη και καθώς η χρήση του στις γραφικές τέχνες διευρύνεται, είναι πιθανό να δούμε πολύ περισσότερα χρήσιμα plug-ins να αναπτύσσονται για το Acrobat Suite. Αυτό επίσης σημαίνει ότι θα έχουμε την ευκαιρία να προσαρμοστούμε στον τρόπο που δουλεύει το πρόγραμμα στην εκάστοτε ροή εργασίας. Όπως συμβαίνει με κάθε τεχνολογική εξέλιξη, το PDF θα καταστεί ισχυρότερο μέσα από την ίδια τη χρήση του. Αυτή τη στιγμή είναι ήδη μια ώριμη τεχνολογία για παρουσίαση και εκτύπωση εγγράφων και συχνά επικρατέστερη λύση για χρήση στο Internet.

Στο μέλλον η δημιουργία PDF αρχείων από δεδομένα που παίρνουμε από βάσεις δεδομένων θα είναι αναγκαία λόγω του τεράστιου όγκου αυτών των δεδομένων και η ταχύτητα με την οποία αυτά μεταβάλλονται. Με αυτό τον τρόπο μια εταιρία ή ένας οργανισμός θα μπορεί να εκμεταλλευτεί όσο το δυνατόν καλύτερα τα δυναμικά δεδομένα της χρησιμοποιώντας τις επιπλέον λειτουργίες που προσφέρει ένα PDF αρχείο πέραν από τα στενά πλαίσια που μας προσφέρει μέχρι τώρα μια δυναμική σελίδα HTML.

Η παρούσα εφαρμογή είναι μια πολύ καλή λύση που συνδυάζει την διαχείριση αρχείων που ήδη υπάρχουν και την δυναμική δημιουργία νέων. Στόχος της είναι η δημιουργία αρχείων υπό μορφή ψηφιακού βιβλίου που να μπορούν εύκολα να τροποποιηθούν. Ένας χρήστης θα μπορεί να χρησιμοποιήσει την εφαρμογή για να θέσει επιπλέον λειτουργίες του Acrobat σε ψηφιακά αρχεία που ήδη υπάρχουν χωρίς να προσθέσει επιπρόσθετη πληροφορία ή να τροποποιήσει την δομή τους.

Για να ικανοποιηθούν οι ανάγκες του τμήματος η εφαρμογή αυτή έχει σαν αντικειμενικό σκοπό την δημιουργία του οδηγού σπουδών του τμήματος Ηλεκτρολόγων Μηχανικών του Ε.Μ.Π. .Ωστόσο κατά την διάρκεια της

υλοποίησης παρουσιάστηκαν μερικά προβλήματα σχετικά με την βάση δεδομένων που παίρνουμε την δυναμική πληροφορία .Το πρόβλημα έγκειται στο γεγονός ότι αφενός μεν η βάση δεν είναι πλήρως ενημερωμένη και αφετέρου δεν υπάρχει όλη η πληροφορία που χρειαζόμαστε .Για τον λόγο αυτό χρειάστηκε να παρέμβουμε με όχι πολύ κομψό τρόπο για να έχουμε το τελικό αποτέλεσμα.

Η εφαρμογή μπορεί να επεκταθεί και να καλύψει τα κενά που θα εμφανιστούν στο μέλλον με την προσθήκη νέων τεχνολογιών στο Acrobat . Επιπλέον λόγω του ότι η δημιουργία των XSL αρχείων απαιτεί εξειδικευμένους χρήστες θα μπορούσαν να προστεθούν και νέες λειτουργίες που να μπορούν να συνδυάσουν και την δημιουργία κατάλληλων XSL αρχείων με ποιο απλό και εύχρηστο τρόπο σε ένα ειδικά διαμορφωμένο interface .Μια ακόμα πιο ενδιαφέρουσα επέκταση θα μπορούσε να είναι και η δημιουργία αρχείων με άλλα formats όπως PCL, PS, SVG, XML , Print, AWT, MIF , TXT αφού όπως έχει αναφερθεί το FOP API έχει αυτή την δυνατότητα. Με τις προσθήκες αυτές θα έχουμε ένα ολοκληρωμένο πρόγραμμα που θα μπορεί να επεξεργάζεται πληροφορία απο βάσεις δεδομένων χωρίς την μεσολάβηση εξειδικευμένων χρηστών και σε διάφορες μορφές .

Βιβλιογραφία :

1. iText Project <http://www.lowagie.com/iText>
2. Apache Avalon Project <http://avalon.apache.org>
3. Fop <http://xml.apache.org/fop>
4. MySql <http://www.mysql.com>
5. The Apache XML project <http://xml.apache.org>
6. Java sun <http://java.sun.com>
7. Laura Lemay, Charles L. Perkins, Michael Morrison , Εγχειρίδιο της Java
8. Stephen J. Mellor, Marc J. Balcer Executable UML.
9. David Hunter , Beginning XML